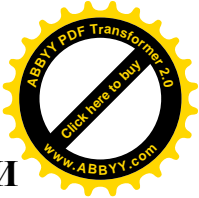
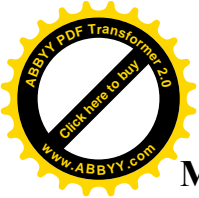


**МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

# **КУРС ЛЕКЦІЙ З ДИСЦИПЛІНИ “ІНФОРМАТИКА”**

**(для студентів економічних спеціальностей)**

**2011**



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

# КУРС ЛЕКЦІЙ З ДИСЦИПЛІНИ “ІНФОРМАТИКА”

(для студентів економічних спеціальностей)

*Розглянуто на засіданні кафедри  
обчислювальної математики і  
програмування.  
Протокол № 1 від 30.08.2011 р.*

*Затверджено на навчально-видавничій  
раді ДонНТУ  
Протокол № 6 від 6.10.2011 р.*

**Донецьк - ДонНТУ - 2011**



**УДК 002(075.8)**

Курс лекцій з дисципліни “Інформатика” для студентів економічних спеціальностей /укладачі: Д.В. Бельков, Є.М. Єдемська - Донецьк: ДонНТУ, 2011. – 296 с.

Посібник відповідає програмі навчального курсу дисципліни „Інформатика”. Він містить в собі конспект теоретичних відомостей необхідних для засвоєння студентами економічних спеціальностей всіх тем курсу і виконання лабораторних робіт.

Укладачі:

Д.В. Бельков, доцент,  
Є.М. Єдемська, ст. викладач,

Рецензент

Відповідальний за випуск:

В.М. Павлиш, професор



## ВСТУП

Навчальний посібник до курсу „Інформатика” відповідає освітньо-професійній програмі підготовки бакалавра за спеціальностями напряму 0501 – „Економіка і підприємство”.

Зміст курсу розкривається в темах:

1. Предмет, методи і завдання дисципліни.
2. Теоретичні основи економічної інформатики.
3. Системне забезпечення інформаційних процесів.
4. Мережні технології.
5. Застосування Інтернету в економіці.
6. Організація комп'ютерної безпеки та захисту інформації.
7. Основи ВЕБ-дизайну.
8. Програмні засоби роботи зі структурованими документами.
9. Програмні засоби роботи з базами та сховищами даних.
10. Основи офісного програмування.
11. Експертні та навчальні системи.
12. Перспективи розвитку інформаційних технологій.

**Метою** курсу є формування знань про принципи побудови і функціонування обчислювальних машин, організацію обчислювальних процесів на персональних комп'ютерах та їх алгоритмізацію, програмне забезпечення персональних комп'ютерів і комп'ютерних мереж, а також ефективне використання сучасних інформаційно-комунікаційних технологій у професійній діяльності.

**Завдання** курсу – вивчення теоретичних основ інформатики і набуття навичок використання прикладних систем оброблення економічних даних та систем програмування для персональних комп'ютерів і локальних комп'ютерних мереж під час дослідження соціально-економічних систем та розв'язання завдань фахового спрямування.

У результаті вивчення дисципліни „Економічна інформатика” студенти зобов'язані:

**знати:**

- теоретичні основи економічної інформатики;
- основи теорії інформації та інформаційних ресурсів;
- кодування інформації в економічних задачах;
- основні етапи вирішення економічних задач;
- апаратні та програмні складові комп'ютерних систем;
- системне забезпечення інформаційних процесів;
- основи Web-дизайну;



- сутність офісного програмування;
- основні поняття сучасних технологій обробки економічної інформації;
- мережні технології;
- основи інформаційної безпеки та захисту інформації;
- програмні засоби роботи зі структурованими документами;
- програмні засоби роботи з базами та сховищами даних;
- поняття про експертні та навчальні системи.

**уміти:**

- виконувати формалізацію економічної задачі;
- здійснювати діалог з операційною системою ПЕОМ;
- створювати різноманітні файли і директорії (папки);
- застосовувати стандартні програмні продукти;
- опрацьовувати текст, графіку, аудіо і відео інформацію;
- здійснювати перевірку і при необхідності форматування носіїв інформації;
- володіти навичками роботи з основними компонентами пакету MS Office (текстовим редактором MS Word, калькулятор електронних таблиць MS Excel, СУБД MS Access);
- розробляти ділову графіку;
- розробляти макроси у MS Excel;
- застосовувати Internet при розв'язанні економічних завдань.



# 1 ПРЕДМЕТ, МЕТОДИ І ЗАВДАННЯ ДИСЦИПЛІНИ

**Метою** курсу є формування знань про принципи побудови і функціонування обчислювальних машин, організацію обчислювальних процесів на персональних комп'ютерах та їх алгоритмізацію, програмне забезпечення персональних комп'ютерів і комп'ютерних мереж, а також ефективне використання сучасних інформаційно-комунікаційних технологій у професійній діяльності.

**Завдання курсу** – вивчення теоретичних основ інформатики і набуття навичок використання прикладних систем оброблення економічних даних та систем програмування для персональних комп'ютерів і локальних комп'ютерних мереж під час дослідження соціально-економічних систем та розв'язання завдань фахового спрямування.

**Предмет** – засоби автоматизації інформаційних процесів з використанням економічних даних.

Слово «інформатика», що позначає автоматизовану обробку, збереження і передачу даних з'явилося в 1960 році у Франції, як гібрид слів «інформація» і «автоматика».

Інформатика займає важливе місце серед комп'ютерних наук (рис. 1.1). Вона складається з 2 великих частин: програмного забезпечення (ПО) і алгоритмічного забезпечення. Програмне забезпечення включає системне ПО, прикладні програми і мови програмування.

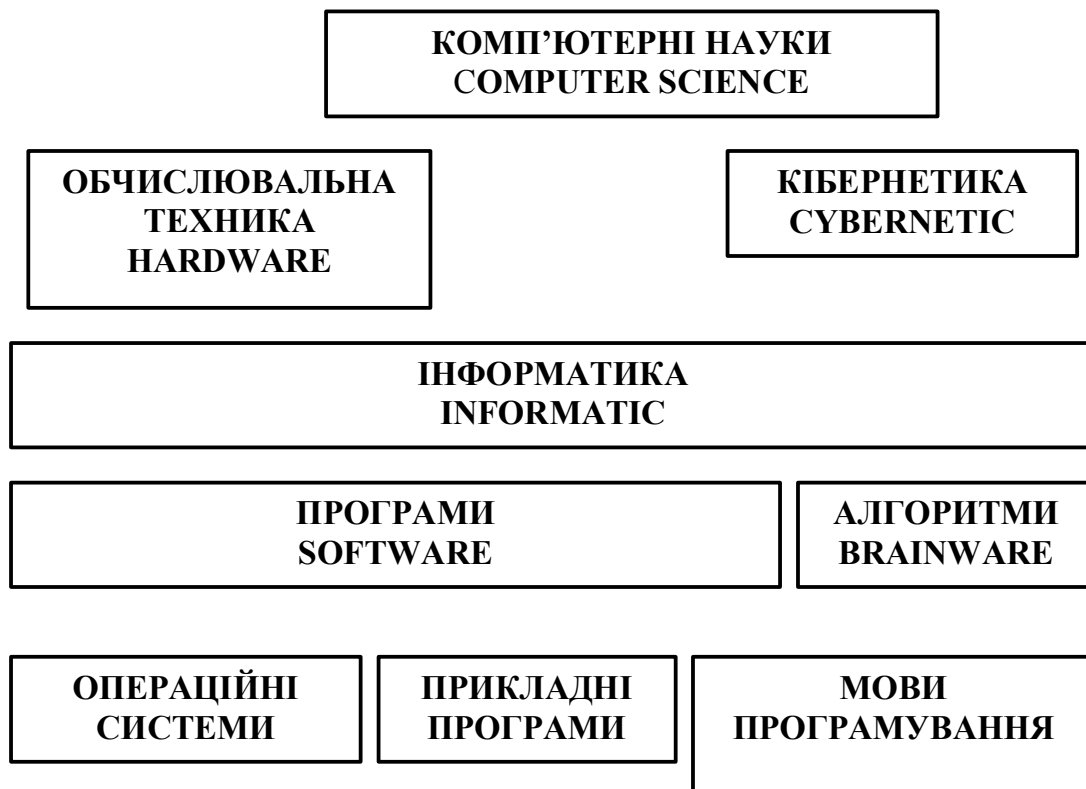


Рис. 1.1 - Структура інформатики



Інформатика як наука виникла в результаті цілеспрямованого вивчення інформаційних процесів.

**Економічна інформатика** виявляє і вивчає властивості економічної інформації, закономірності її переробки; процесів керування переробкою інформації в штучних, соціальних і біологічних системах і спрямована на рішення економічних задач. Її найважливішими призначеннями є: збір, перетворення і реєстрація інформації, обробка і збереження, перетворення, тиражування, поширення і використання для рішення економічних задач по керуванню економічними об'єктами.

Під **економічною інформацією** розуміють інформацію про процеси суспільного виробництва, обміну і споживання матеріальних благ. По призначенню в процесі керування економічна інформація підрозділяється на керуючу (командну) і що інформує (наприклад обліково-статистичну).

Економічна інформація:

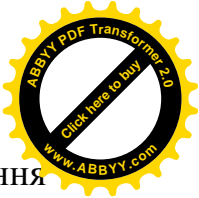
- *специфічна за формою представлення*, відбиває у виді первинних і зв'язкових, юридично оформлених (тобто підписи, що має,) на традиційних електронних документах;
- *об'ємна*, тобто містить детальну інформацію про процеси, для керування якими вона використовується;
- *циклічна*, тому що для більшості виробничих і господарських процесів характерна повторюваність складових їхніх стадій і відповідної інформації, що описує ці процеси;
- *специфічна по способах обробки* (переважають арифметичні і логічні операції, а результати представляються у виді текстів, таблиць, діаграм, графіків).

Найбільш важливими вимогами, пропонованими до економічної інформації, є:

- *коректність*, тобто однозначність для всіх споживачів;
- *цінність* - відносна (для різних споживачів - різна) властивість, що
- виявляється в тім випадку, якщо інформація використовується для досягнення мети;
- *оперативність*, відбиває актуальність інформації для необхідних розрахунків і прийняття рішень в умовах, що змінюються;
- *точність* - визначає припустимий рівень перекручувань вхідний, вихідний і ін. типів інформації;
- *вірогідність* - властивість інформації відбивати реально існуючі об'єкти з необхідною точністю;
- *стійкість* - здатність реагувати на зміну вихідних даних без порушення необхідної точності;
- *достатність* - означає, що вона містить мінімально необхідний обсяг зведень, необхідних для ухвалення правильного рішення.

Дослідження економічної інформації дозволили:

- класифікувати інформацію з місця виникнення (вхідна, вихідна); по участі в процесі обробки/збереження (вихідна, похідна, збережена без



обробки, проміжна, результуюча); стосовно функцій керування (планова, прогнозна, нормативна, конструкторсько-технологічна, облікова, фінансова і т.д.);

- виявити ряд особливостей, що впливають на організацію автоматизованої обробки.

## 1.1 Інформація і керування

Визначальною областю в процесі діяльності людини є організаційне керування, що розуміється як процес цілеспрямованого впливу на об'єкт, що організує його функціонування по деякій заданій програмі.

З погляду інформаційних процесів **керування** складається з наступних складових:

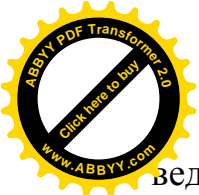
- вироблення керуючим органом керуючої інформації, що відповідає мети (програмі) керування;
- передача керуючої інформації об'єктові керування;
- одержання й аналіз реакції об'єкта;
- коректування або вироблення нової керуючої інформації з метою оптимізації функціонування об'єкта керування.

Система керування економічним об'єктом являє собою людино-машинний комплекс, в основі якого визначаються наступні підсистеми:

- інформаційне забезпечення - це система класифікації і кодування інформації, технологічна схема обробки даних, нормативно-довідкова інформація документообігу;
- організаційне забезпечення - сукупність мір і заходів, що регламентують функціонування системи керування, її опис, інструкції і регламенти обслуговуючому персоналові;
- технічне забезпечення - комплекс використовуваних у системі керування технічних засобів, у тому числі ЕОМ і засобів зв'язку;
- математичне забезпечення - сукупність методів, правил, математичних моделей і алгоритмів рішення задач;
- лінгвістичне забезпечення - сукупність термінів і штучних мов, правил формації природної мови;
- програмне забезпечення - сукупність програм, систем обробки даних і документів, необхідних для експлуатації цих програм;
- правове забезпечення - сукупність правових норм, що визначають створення, юридичний статус і функціонування системи.

Керування економічним об'єктом - основна частина інформаційної технології рішення економічної задачі. Її найважливішими процедурами є: збір, реєстрація, збереження, обробка, перетворення, тиражування, поширення і використання інформації. У процесі керування об'єктом економічна інформація піддається, як правило, усім процедурам, у ряді випадків частини з них може не бути. Послідовність процедур також може бути різної. Деякі з них можуть повторюватися. Їхній склад і особливості залежать від економічного об'єкта, що





веде автоматизовану обробку інформації, і процесів, що протікають у його середовищі.

## 1.2 Інформаційні технології

Під **інформаційною технологією (ІТ)** розуміють сукупність методів, способів, прийомів і засобів обробки документованої інформації, включаючи прикладні програмні і засоби, порядок проходження їхніх застосувань, а також сукупність усіх видів інформаційної техніки. ІТ орієнтуються на одержання, обробку і поширення інформації.

Становлення і розвиток технологій стрімко змінює навколишній світ. Із самого початку ці процеси розглядалися в тісному зв'язку з економічними об'єктами, тому що процес усякої діяльності здійснюється за технологією, обумовленою метою, предметом, засобами, характером операції і результатами.

Поняття "технологічний процес" в економічній літературі фактично було витиснуто терміном "**бізнес-процес**". Ці поняття багато в чому збігаються.

В основі опису бізнес-процесів лежать поняття:

- об'єкт - інформаційний, матеріальний або фінансовий, використовуваний у бізнесі-процесі (устаткування, рахунок);
- подія – зовнішнє (не контрольоване в рамках процесу) дія, що відбулася з об'єктом (одержання листа, поломка устаткування, нарахування штрафу);
- операція – елементарна дія, виконувана в рамках розглянутого бізнесу-процесу (посилка листа, оплата рахунка);
- виконавець – посадова особа, відповідальне за виконання однієї або декількох операцій бізнесу-процесу (менеджер, співробітник архіву, директор).

ІТ відіграють важливу роль у підтримці бізнесу-процесу. Вони проникли в усі види діяльності людини, тому що дозволяють інтегрувати різні види технологій, синтезувати і накопичувати інформацію для впровадження в практику відповідно до суспільних потреб. Метою широкого поширення ІТ є рішення проблеми інформатизації суспільства, що розуміється як поширення і впровадження комплексу мір, спрямованих на своєчасне використання достовірної інформації у всіх сферах людської діяльності.

### Інформаційний менеджмент

У сучасних організаціях часто виникає потреба керувати процесом одержання інформації. Метою такого керування є підвищення ефективності розвитку організації. Процес досягнення цієї мети має назва інформаційного менеджменту. Інформація в такому випадку стає стратегічним ресурсом і повинна розглядатися як економічна категорія. Інформаційний менеджмент охоплює планування, координацію, контроль інформаційної діяльності організації й інформаційних процесів у ній. Він приводить до цілеспрямованого використання інформації як ресурсу.

Економічна інформація – сукупність зведень про соціально-економічні процеси, що служать для керування цими процесами і колективами людей у виробничій і невиробничій сферах.

Метою керування може бути виживання в конкурентній боротьбі, одержання максимального прибутку або вихід на визначені ринки. Як показано на рис. 1.2, будь-яка система керування складається з керованої частини (об'єкта керування) і керуючої частини (орган керування).

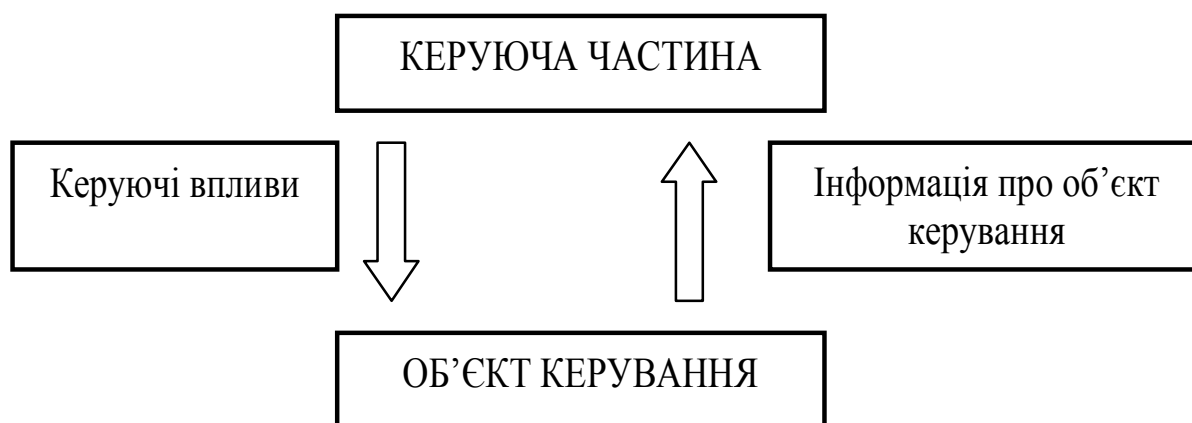


Рис. 1.2 – Система керування

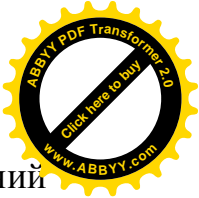
Керуюча частина одержує від керованої частини інформацію про керований процес і передає в керовану частину керуючі впливи. Таким чином, у системі керування завжди присутнє замкнутий інформаційний контур.

Інформаційний контур організації разом із засобами збору, передачі, обробки і збереження інформації, а також разом з персоналом, що здійснює ці дії, утворюють *інформаційну систему* організації.

Інформаційна система керування – сукупність інформації, економіко-математичних методів і моделей, технічних, програмних, інших технологічних засобів і фахівців, призначена для обробки інформації і прийняття управлінських рішень. Інформаційна система повинна вирішувати поточні задачі стратегічного і тактичного планування, бухгалтерського обліку й оперативного керування фірмою.

Інформаційні системи керування дозволяють:

- Підвищувати ступінь обґрунтованості прийнятих рішень за рахунок оперативного збору, передачі й обробки інформації
- Забезпечувати своєчасність прийняття рішень по керуванню організацією в умовах ринкової економіки
- Домагатися росту ефективності керування за рахунок своєчасного представлення необхідної інформації керівникам усіх рівнів керування з єдиного інформаційного фонду
- Погоджувати рішення, прийняті на різних рівнях керування й у різних структурних підрозділах



- За рахунок інформованості управлінського персоналу про поточний стан економічного об'єкта забезпечувати ріст продуктивності праці і скорочення невиробничих утрат.

Керуюча частина системи керування, як правило, має трьохрівневу структуру: стратегічне керування, тактичне керування, оперативне керування. Тому інформаційні системи бувають 3-х рівнів (рис. 1,3):

- 1) стратегічні інформаційні системи;
- 2) тактичні інформаційні системи;
- 3) оперативні інформаційні системи.

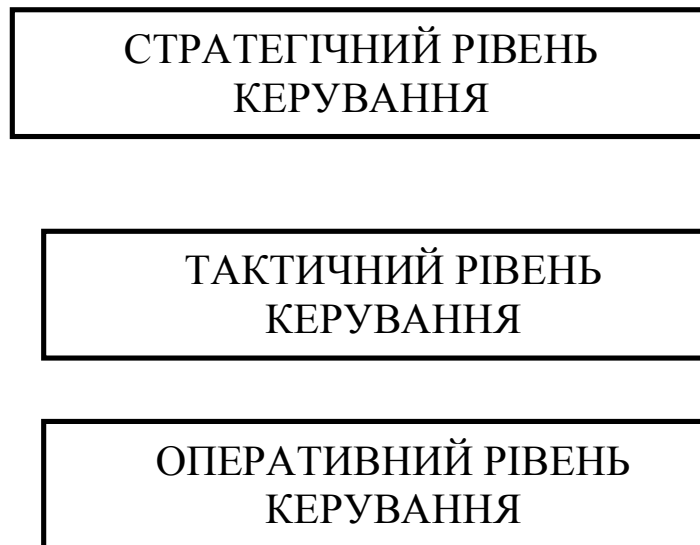
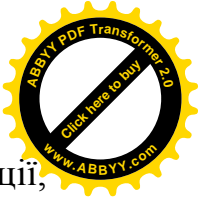


Рис. 1.3 – Рівні керування

У рамках інформаційної системи відбувається перетворення інформації. Спосіб перетворення інформації називається інформаційною технологією. Інформаційна система є середовищем для реалізації інформаційних технологій. Однак інформаційна технологія ширше, ніж інформаційна система. В інформаційній системі можуть використовуватися кілька інформаційних технологій. Ціль будь-якої інформаційної технології – одержати потрібну інформацію необхідної якості на заданому носії. При цьому існують обмеження на вартість обробки даних, трудомісткість процесів використання даних, надійність, оперативність і якість інформації.

Інформаційні технології організації служать стратегічного цілям бізнесу, використовуються для керування структур і об'єктів, фінансовими, інформаційними, матеріальними потоками, робітниками місцями і колективами людей. Стратегічні цілі інформаційних технологій – забезпечити розвиток бізнесу, його керованість і якість, конкурентноздатність, зниження вартості виконання бізнес-процесів.

До типових технологічних операцій відносять: збір і реєстрацію інформації, її передачу, уведення, обробку, висновок, збереження, нагромадження, пошук, аналіз, прогноз, прийняття рішень.



Зростання обсягів інформації в інформаційній системі організації, потреба в прискоренні й ускладненні обробки інформації приводить до необхідності автоматизації інформаційної системи організації.

Як правило, автоматизована інформаційна система являє собою систему підтримки прийняття рішень або систему виробництва інформаційних продуктів, що використовує комп'ютерну інформаційну технологію, і персонал, що працює з комп'ютерами і телекомунікаціями.

Автоматизована інформаційна система має забезпечуючу і функціональну частини.

Забезпечуюча частина складається з інформаційного, технічного, математичного, програмного, методичного й організаційного, лінгвістичного забезпечення.

Інформаційне забезпечення – сукупність проектних рішень по обсягах, розміщенню, формам організації інформації. У рамках інформаційного забезпечення розрізняють зовнішньомашинні і внутрішньомашинні дані. Зовнішньомашинна інформаційна база сприймається людиною без технічних засобів (убрання, накладні, акти і т.п.). Внутрішньомашинна інформаційна база складається з файлів і зберігається на носіях.

Технічне забезпечення – комплекс технічних засобів, призначених для роботи інформаційної системи, а також технічна документація на ці засоби.

Математичне забезпечення - сукупність математичних моделей, методів і алгоритмів, що використовуються в інформаційній системі.

Програмне забезпечення – сукупність програм для реалізації цілей і рішення задач інформаційної системи.

Методичне й організаційне забезпечення – сукупність методів, засобів і документів, що регламентують взаємодію персоналу інформаційної системи з технічними засобами і між собою в процесі розробки й експлуатації системи.

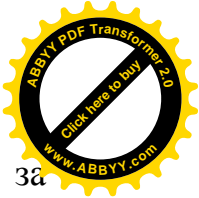
Лінгвістичне забезпечення – сукупність мов спілкування (язикових засобів) персоналу інформаційної системи з програмним, технічним і інформаційним забезпеченням.

Функціональна частина забезпечує виконання задач інформаційної системи. Фактично тут утримується модель системи керування організацією. Функції інформаційних систем реалізуються за допомогою інформаційних технологій.

Існує класифікація інформаційних технологій автоматизованих інформаційних.

### **1.3 Приклади інформаційних технологій і процедур обробки економічної інформації**

Електронний офіс – це технологія обробки інформації в установі електронними засобами, що базуються на обробці даних, документів, таблиць, текстів, зображень, графіків.



Найбільше ефективно технологія електронного офісу реалізується за допомогою інтегрованих пакетів прикладних програм, наприклад Microsoft Office.

Найбільшу складність представляє автоматизація функцій аналізу, адміністрування, прийняття рішень і прогнозу.

Однієї з важливих процедур обробки економічної інформації є обробка таблиць. Вона здійснюється комплексом прикладних програм у складі електронного офісу і доповнюється поруч аналітичних можливостей.

Технології обробки фінансово-економічної інформації на наступних прикладних програмних продуктах:

- універсальні табличні процесори
- спеціалізовані програми бухгалтерських розрахунків
- спеціалізовані банківські програми
- спеціалізовані програми фінансово-економічного аналізу і бізнес - планування.

Електронна пошта – здійснює технологію передачі повідомлень, текстів, документів зображень з використанням електронної техніки.

Різноманітні фірми – виробники пропонують різний набір послуг і структуру електронної пошти. Електронна пошта є стандартною послугою світової мережі Інтернет

Інтернет - технологія заснована на об'єднанні інформаційних мереж у глобальну інформаційну структуру. Інтернет – це глобальна міжнародна асоціація інформаційних мереж, що має інформаційні центри, що обслуговують користувачів: вони надають документацію, поширюють програми, тексти книг, ілюстрації, комунікаційний сервіс, електронну пошту, службу новин.

Інтернет - технологія використовується в освітніх і наукових цілях, у бізнесі.

Через Інтернет численні фірми надають різноманітні види сервісу: свої бази даних, рекламу продукції і послуг, консультації й інші види допомоги.

Розвиток Інтернет – технологій усе більш спрямовано в область інтелектуальних, наукомістких проблем.

Візуалізація даних, обробка зображень, створення віртуального простору дозволяє людині зануритися в образне середовище рішення складних задач, наблизитися до поставленим цілям на якісно новому рівні, полегшити підготовку і прийняття управлінських рішень.



## 2 ТЕОРЕТИЧНІ ОСНОВИ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ

### 2.1 Загальні відомості про персональний комп'ютер

Персональний комп'ютер (ПК) - це універсальна мікроЕОМ. ПК складається з декількох пристроїв (блоків). Всі блоки комп'ютера з'єднуються між собою спеціальною системною шиною. Типова конструкція ПК складається з базового комплекту: системного блоку, дисплея (монітора) і клавіатури. До базового комплекту можуть приєднуватися такі допоміжні пристрої: миша; сканер; модем; плоттер; сітьовий адаптер; принтер; музична приставка; графічний планшет й інші.

#### 2.1.1 Системний блок

Він містить процесор, постійну й оперативну пам'ять, пристрої для зберігання інформації на магнітних дисках, адаптери (пристрої для управління дисплеєм, принтером, мишею).

**Процесор** - це основний пристрій ПК або мозковий центр комп'ютера. Він містить:

- основний мікропроцесор, що керує роботою комп'ютера і виконує всі обчислення;
- оперативну пам'ять, в якій розташовуються програми, що виконуються комп'ютером, і дані, що використовуються програмами;
- електронні схеми (контролери), що керують роботою різних пристроїв (монітора, накопичувачів на магнітних дисках й інших.);
- порти вводу-виводу, через які процесор обмінюється даними із зовнішніми пристроями.

У сучасних ПК застосовуються такі типи мікропроцесорів: 80286, 80386, 80486, 80486SX, 80486DX/2, 80486DX/4, 80586 (Pentium), AMD K5, Cyrix 6x86, 80686 (Pentium Pro), Pentium MMX, Pentium II й інші.

При зазначенні типу мікропроцесора перші дві цифри (80) частіше за все не вказуються.

Тип мікропроцесора, що використовується, пов'язаний з такою технічною характеристикою, як швидкодія. Основним показником швидкодії є тактова частота (що вимірюється в мегагерцах - МГц) електричних імпульсів, що задають ритм роботи електронних схем. Чим більше тактова частота, тим вище продуктивність комп'ютера. Основні типи мікропроцесорів і їх тактові частоти наведені в таблиці 2.1.

Продуктивність комп'ютера залежить також і від розрядності шини, яка буває 8-ми, 16-ти, 32-х і 64-х розрядна. Тобто істотною характеристикою процесора є кількість біт, яку він може обробляти одночасно в одиницю часу.

Обсяг оперативної пам'яті комп'ютера також впливає на його швидкодію. Чим більше оперативної пам'яті на ПК, тим вище його швидкодія.





Таблиця 2.1 – Основні типи мікропроцесорів

Тип мікропроцесора (CPU)	Тактова частота, МГц
80286	6 - 20
80386SX	16 - 25
80386DX	33
80486DX	33 - 50
80486SX	20 - 33
80486DX/2	66
80486DX/4	75 - 100
80586(Pentium)	60 - 200
AMD K5	75 - 133
Cyrix 6x86	133 - 200
80686(Pentium Pro)	133 - 200
Pentium MMX	133 - 233
AMD K6	166 - 260
Pentium II	233 і вище

### 2.1.2 Пам'ять

Оперативна пам'ять ПК поділяється на постійний запам'ятовуючий пристрій (ПЗП, ROM), оперативний запам'ятовуючий пристрій (ОЗП, RAM) і розширену пам'ять. ОЗП - енергозалежна пам'ять, ПЗП - енергонезалежна пам'ять. Це означає, що при вимкненні ПК інформація, що зберігалася в ОЗП, безповоротно зникає.

На даний час використовуються комп'ютери з оперативною пам'яттю від 1 до 64 Мб і більше.

ОЗП - це тимчасова пам'ять, тобто в неї завантажується інформація з моменту включення комп'ютера, яка зберігається до вимкнення комп'ютера. У ОЗП зберігаються програми, що виконуються, і дані до них. Для довгострокового зберігання інформації призначена зовнішня пам'ять, що знаходиться на гнучких і жорстких магнітних дисках.

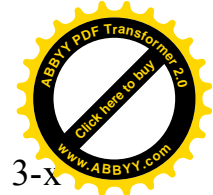
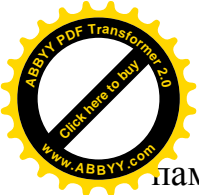
### 2.1.3 Гнучкі магнітні диски

У системному блоці можуть розташовуватися накопичувачі (дисківоди) для гнучких магнітних дисків (дискет) двох розмірів: 3,5 дюйма і 5,25 дюйма. Дискети 3,5 дюйма мають обсяг пам'яті 720 Кб, 1,44 Мб і надто рідко 2,88 Мб. Дискети 5,25 дюйма підвищеної щільності мають стандартний обсяг 1,2 Мб, а подвійної щільності - 360 Кб.

Дискети - це зовнішні зйомні накопичувачі інформації. Після запису інформації на гнучку дискету на одному комп'ютері її можна прочитати з цієї ж дискети на іншому комп'ютері.

### 2.1.4 Жорсткий диск (Вінчестер)

Жорсткий диск (HDD - Hard Disk Drive) знаходиться в спеціальному дисківоді жорсткого диска, розташованому в системному блоці. Обсяг його



пам'яті на декілька порядків вище за обсяг дискет і складає від 20 Мб до 3-х Гб і більше. Жорсткий диск - це зовнішній незйомний накопичувач інформації.

### **2.1.5 Дисплей (Монітор)**

Дисплей - це пристрій для відображення інформації на екрані. Сигнали, які отримує монітор, формуються відеокартою.

Монітор і відеокарта повинні працювати, як єдине ціле, і для оптимальної роботи повинні бути відповідно настроєні.

У даний час нараховується більше ніж 30 модифікацій різних типів відеокарт. Їх прийнято класифікувати за такими стандартами: DA, CGA, EGA, VGA, Super VGA.

### **2.1.6 Клавіатура**

Клавіатура - це пристрій для введення інформації в ЕОМ.

### **2.1.7 Приводи CD-ROM**

CD (Compact Disk - компакт-диск) - це лазерний диск, досить могутній, для зберігання інформації. Його обсяг 500 - 700 Мб. (CD-ROM - Compact Disk Read Only Memory - компакт-диск тільки для читання). Лазерний диск вставляється в привід CD-ROM (дискковод лазерного диска), розташований в системному блоці ПК.

### **2.1.8 Принтер**

Принтери призначені для виведення інформації на друк. Бувають такі типи принтерів: матричні, струминні, лазерні, термічні.

### **2.1.9 Плоттер**

Цей пристрій для виведення на друк креслень, схем, графіків, діаграм й інш. Плоттери звичайно використовуються спільно з програмами Систем автоматичного проектування (САПР).

### **2.1.10 Миша**

Миша - це пристрій, який прискорює введення інформації в ПК.

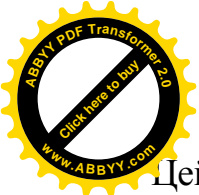
### **2.1.11 Сканер**

Сканер - це пристрій, що дозволяє вводити в комп'ютер в графічному вигляді текст, слайди, фотографії, малюнки, схеми, креслення й інш. Сканери бувають різних типів: ручні, планшетні, роликові (факс-сканери), барабанні й інш.

### **2.1.12 Модем**

Модем (Модулятор - Демодулятор) - це пристрій, що дозволяє обмінюватися інформацією між ПК через звичайні телефонні канали зв'язку.





Цей пристрій перетворює сигнали комп'ютера в сигнали телефонної лінії й навпаки.

## 2.2 Загальна характеристика програмного забезпечення

Вся сукупність програм називається програмним забезпеченням, а сукупність технічних засобів комп'ютера - його апаратним забезпеченням. Програмне забезпечення поділяється на системне й прикладне забезпечення.

Системним програмним забезпеченням називається таке програмне забезпечення, яке використовується для розробки, підтримки, виконання інших програм, тестування обладнання, а також для надання користувачеві ПК набору всіляких послуг.

Серед системних програмних продуктів основну роль грає операційна система (ОС). Технічна справність комп'ютера є необхідною умовою нормальної роботи користувача за комп'ютером. Достатньою умовою нормальної роботи є установка ОС на комп'ютер. Якщо ОС не встановлена, то робота користувача за комп'ютером практично неможлива.

ОС організує виконання всіх призначених для користувача програм, забезпечує взаємодію користувача з комп'ютером, а також надає як програмам, так і користувачеві ряд інших послуг. Вона бере на себе реалізацію багатьох рутинних операцій, зокрема, введення даних в програму і виведення результатів її роботи. Крім цього, ОС завантажує програми в оперативну пам'ять для їх виконання, розподіляє між ними свої ресурси (пам'ять, процесорний час), а також сприймає і виконує команди користувача. Дуже важливим є й те, що вона бере на себе функцію збереження програм і результатів їх роботи у зовнішній пам'яті (на дисках). Таким чином, ОС є посередником між комп'ютером з одного боку й програмами, що виконуються, і користувачем з іншого боку. Представниками сім'ї ОС є MS-DOS, WINDOWS-95, WINDOWS-98, WINDOWS-NT, OS-2 й інші.

До системних продуктів відносяться також сервісні системи, інструментальні системи й системи технічного обслуговування.

Сервісні системи розширюють можливості ОС, надаючи користувачеві, а також програмам, що виконуються, набір додаткових послуг. До сервісних систем відносяться: оболонки, утиліти, а також програмні продукти, здатні істотно змінити вигляд ОС (часто їх називають операційними засобами або інтерфейсними системами).

Оболонка операційної системи - це такий програмний продукт, який полегшує спілкування користувача з комп'ютером і надає йому ряд додаткових можливостей, яких не має ОС (наприклад, оболонка NORTON COMMANDER для ОС MS-DOS). Для систем з розвиненим інтерфейсом (тобто що мають засоби й методи взаємодії з користувачем, які дозволяють простіше й наочніше запускати програми, відкривати й зберігати документи, працювати з дисками) оболонки практично не потрібні.

Утиліта - це службова програма, що надає користувачеві деякі додаткові послуги. До утиліт відносяться такі програми:

- дискові компресори (системи динамічного стиснення даних);



- дискові дефрагментатори;
- програми, що забезпечують захист і відновлення комп'ютерних даних;
- архіватори;
- програми резервування (резервного копіювання) даних;
- антивірусні програми;
- утиліти, що оптимізують використання оперативної пам'яті й інш.

Відмінності між оболонками й утилітами досить умовні. Вони полягають лише в універсальному характері перших і відносній спеціалізації других.

Інтерфейсна система має всі ознаки оболонки, але додатково до цього формує нову середу виконання програм, що є винятковою прерогативою ОС. Інтерфейсну систему не можна назвати операційною, тому що вона не може функціонувати без ОС. Вона є надбудовою над операційною системою. Типовими представниками інтерфейсних систем є WINDOWS 3.1, WINDOWS 3.11 й інш.

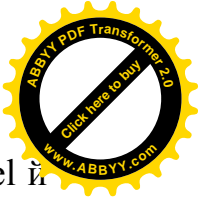
Інструментальні системи - це такі програмні продукти, які створені для розробки програмного забезпечення. До них відносяться:

- системи програмування (Turbo Pascal, MS Visual Basic 5.0, Delphi й інш.);
- системи управління базами даних (Access, FoxPro, Clipper й інш.);
- інструментарій штучного інтелекту й інш.

Системи технічного обслуговування призначені для полегшення тестування ПК і пошуку несправностей.

Прикладне програмне забезпечення - це програми, призначені для рішення певних цільових задач з різних областей людської діяльності. Відносно недавно набув поширення термін додаток - під ним розуміють прикладну програму. У даний час (після появи ОС WINDOWS 98) в поняття додатка все частіше вкладають дещо інший зміст - до додатка неявно відносять будь-які програми, що функціонують під управлінням операційної системи (значить, будь-які програми, за винятком ОС). Якщо потрібно вказати, для якої операційної системи додаток розроблено, то кажуть, наприклад, так: "додаток WINDOWS" або "WINDOWS додаток". До прикладних програм відносяться:

- текстові редактори, що забезпечують підготовку текстових документів (MS WORD, LEXICON й інш.);
- графічні редактори, що служать для підготовки ілюстрацій (Corel Draw, Paint й інш.);
- презентаційні програми, що полегшують створення слайдів для їх показу на різних офіційних заходах, узагальнено званих презентаціями (MS PowerPoint й інш.);
- планувальники, що забезпечують планування робочого часу і зустрічей (MS Schedule + й інш.);
- системи верстки або видавничі пакети (Ventura, PageMaker й інш.);
- інтелектуальні, в тому числі експертні системи;
- інформаційно-пошукові системи;
- табличні процесори (електронні таблиці) - програми, що



автоматизують форматування таблиць і побудову діаграм (MS Excel й інш.);

- інструментарій мультимедіа (засоби відтворення - "програвачі", засоби запису - "магнітофони", а також редактори звукових і відео файлів, музичні синтезатори) і програмні продукти, що мають властивості мультимедіа (довідники і енциклопедії);
- навчальні системи;
- математичні програми - інструмент математиків, що допомагає робити обчислення по складних формулах і розв'язувати системи рівнянь;
- програми моделювання;
- системи автоматичного проектування (AutoCAD);
- комунікаційні програми;
- комп'ютерні ігри й інш.

### 2.3 Представлення інформації в ЕОМ

ЕОМ - це електронний цифровий обчислювальний пристрій.

Електронний пристрій - тому, що будь-яка інформація в ньому обробляється, як правило, за допомогою електронних сигналів.

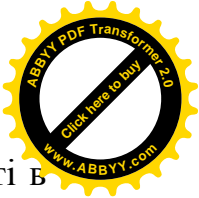
Цифровий пристрій - тому, що будь-яка інформація в ньому подається за допомогою чисел.

Для того, щоб записувати числа, треба використати якусь систему числення. У ЕОМ для запису чисел використовується двійкова система числення. У цій системі числення всього дві цифри 0 і 1, основа системи дорівнює 2. Двійкова система використовується в комп'ютерах тому, що електричними сигналами дуже просто визначити двійкові цифри: 0 - немає сигналу й 1 - є сигнал. Будь-яке число ЕОМ автоматично подає у вигляді послідовності цифр 0 і 1, тобто в двійковій системі числення.

Будь-яку інформацію можна розподілити на невеликі елементарні частини. Так, наприклад, елементарною часткою тексту є символ. Щоб ЕОМ сприймала текстову інформацію, кожному символу поставили у відповідність десяткове число, яке називають кодом даного символу. Елементарною ж часткою обсягу пам'яті в ЕОМ є 1 біт, в якому може зберігатися цифра 0 або 1 двійкової системи числення. Тому будь-який десятковий код символу, перетворений за допомогою двійкової системи числення в набір  $N$  штук нулів і одиниць, буде займати  $N$  біт пам'яті. Але оскільки 1 біт дуже дрібна одиниця обсягу пам'яті, то в комп'ютерах прийняли за основну одиницю обсягу пам'яті 1 байт, що складається з 8-ми бітів. Тобто  $1 \text{ байт} = 2^3 \text{ біт} = 8 \text{ біт}$ .

Максимальне ціле позитивне число, яке можна вмістити в обсяг пам'яті 1 байт дорівнює 255, тому що  $(255)_{10} = (11111111)_2$ .

Оскільки код будь-якого символу не більш ніж 255, то для його зберігання буде потрібний 1 байт пам'яті. Наприклад, для зберігання однієї сторінки тексту з 500 символів буде потрібно 500 байт пам'яті.



Використовуються і більш великі одиниці вимірювання обсягу пам'яті в ЕОМ (кілобайт - Кб, мегабайт - Мб, гігабайт - Гб):

$$1 \text{ Кб} = 2^{10} \text{ байт} = 1024 \text{ байт};$$

$$1 \text{ Мб} = 2^{10} \text{ Кб} = 1024 \text{ Кб} = 1024 \cdot 1024 \text{ байт};$$

$$1 \text{ Гб} = 2^{10} \text{ Мб} = 1024 \text{ Мб} = 1024 \cdot 1024 \text{ Кб} = 1024 \cdot 1024 \cdot 1024 \text{ байт}.$$

Вся пам'ять ЕОМ розбита на комірки пам'яті. Обсяг пам'яті однієї комірки становить 1 байт. Кожна комірка пам'яті має адресу або номер комірки (він, як правило, записується в шістнадцятковій системі числення) і вміст у вигляді двійкового коду, який зберігається в ній.

### 2.3.1 Переведення числа з однієї системи числення в іншу

Розглянемо двійкову, вісімкову, шістнадцяткову й десяткову системи числення.

У десятковій системі числення основою системи є число 10, а як цифри використовуються: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

У двійковій системі числення основою системи є число 2, а як цифри використовуються: 0 і 1.

У вісімковій системі числення - основою системи є число 8, а як цифри використовуються: 0, 1, 2, 3, 4, 5, 6, 7.

У шістнадцятковій системі числення основою системи є число 16, а як цифри використовуються: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15. Причому замість 10, 11, 12, 13, 14, 15 для запису чисел в шістнадцятковій системі використовують відповідно символи А, В, С, D, Е, F.

У будь-якій системі числення будь-яке число N у позиційному запису зображується у вигляді:

$$N = (P_K P_{K-1} \dots P_2 P_1 P_0, P_{-1} P_{-2} \dots P_{-m})_b, \quad (2.1)$$

де  $b$  - основа системи числення;

$K$  - дорівнює кількості цифр цілої частини числа мінус одиниця;

$m$  - кількість цифр дробової частини числа;

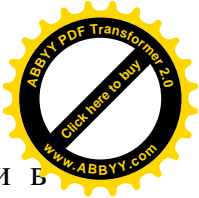
$P_i$  - позиційна цифра (причому  $0 \leq P_i \leq b-1$ , для  $i = -m, \dots, -2, -1, 0, 1, 2, \dots, K-1$ , що стосується  $P_k$ , то  $1 \leq P_k \leq b-1$ ).

Формула (2.1) - це символічний запис суми членів степенного многочлену:

$$N = P_K b^K + P_{K-1} b^{K-1} + \dots + P_2 b^2 + P_1 b^1 + P_0 b^0 + P_{-1} b^{-1} + P_{-2} b^{-2} + \dots + P_{-m} b^{-m} = \sum_{i=-m}^K P_i b^i. \quad (2.2)$$

Отже, якщо відомо число в системі числення  $b$  з позиційними цифрами  $P_i$ , то, підставивши їх в праву частину (2.2), отримаємо число N в десятковій системі числення.

**Правило 1.** Щоб перевести число з будь-якої системи числення з основою  $b$  в десяткову систему числення, необхідно використати формулу (2).



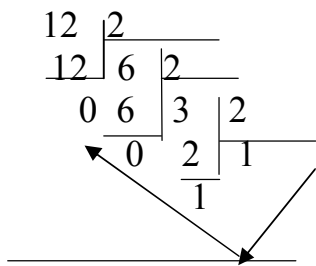
Наприклад, відоме число в двійковій системі  $(101)_2$  треба перевести в десяткову систему.

У цьому випадку:  $b = 2$ ;  $m = 0$ ;  $K = 2$ ;  $P_2 = 1$ ;  $P_1 = 1$ ;  $P_0 = 1$ ;  $N = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 1 = 5$ .

**Правило 2.** Щоб перевести цілу частину числа, представлену в десятковій системі числення, в число іншої системи числення, необхідно:

- 1) поділити цілу частину числа на основу нової системи числення;
- 2) залишок від ділення буде останньою позиційною цифрою числа в новій системі числення;
- 3) якщо частка від ділення більше основи нової системи числення, то його треба ділити на основу цієї системи. Залишок від ділення буде передостанньою позиційною цифрою числа в новій системі;
- 4) таке ділення повторювати, доки частка від ділення не стане меншою від основи нової системи числення, причому ця частка від ділення буде першою позиційною цифрою нового числа.

Наприклад, десяткове число 12 перевести в двійкову систему числення:



Результат:  $12_{10} = (1100)_2$

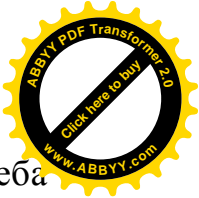
**Правило 3.** Для переведення дробової частини числа, представленого в десятковій системі числення, в число іншої системи числення необхідно:

- 1) помножити дробову частину числа на основу нової системи числення;
- 2) у отриманому добутку виділити цілу частину, яка стане першою позиційною цифрою дробової частини числа в новій системі числення;
- 3) дробову частину добутку знову помножити на основу нової системи числення, ціла частина нового добутку буде наступною позиційною цифрою дробової частини числа;
- 4) пункт 3) повторювати доти, доки не буде досягнута потрібна точність числа.

Наприклад, перевести число 0,25 в двійкову систему числення.

$$\begin{array}{r}
 0,25 \\
 \times \quad 2 \\
 \hline
 0,50 \\
 \times \quad 2 \\
 \hline
 1,00
 \end{array}$$

Результат:  $(0,25)_{10} = (0,01)_2$



Для переведення з двійкової системи числення у вісімкову систему треба використати тріади, що відраховуються зліва направо в двійковому числі, оскільки для зберігання вісімкової позиційної цифри потрібно три біти.

**Правило 4.** Для переведення з двійкової системи числення у вісімкову систему необхідно:

- 1) у двійковому числі виділити підряд по три цифри (тріади) вліво і вправо від коми, доповнивши при необхідності, нулями, які не мають значення;
- 2) для кожної тріади поставити у відповідність вісімкову цифру (використовуючи таблицю 2.2).
- 3) отримані вісімкові числа утворять число у вісімковій системі числення (якщо читати цифри зліва направо).

*Наприклад*, перевести число, подане в двійковій системі числення  $(11\ 001,11)_2$ , в число у вісімковій системі числення.

3    1    6

Результат:  $(11\ 001,11)_2 = 31,6_8$

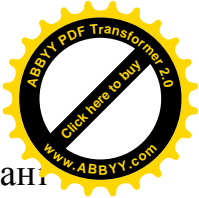
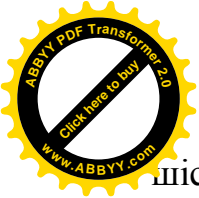
Таблиця 2.3 – Кодування десяткових чисел в різних системах числення

Десяткове число	Двійкове число	Вісімкове число	Шістнадцяткове число
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

**Правило 5.** Для переведення числа з вісімкової системи числення в двійкову систему виконуються дії, зворотні діям правила 4. А саме, для кожної позиційної вісімкової цифри числа виписати відповідні двійкові тріади з таблиці 2.2.

**Правило 6.** Для переведення числа з двійкової системи числення в шістнадцяткову систему необхідно в двійковому числі послідовно виділити чотирьохбітові групи вліво і вправо від коми, доповнивши, при необхідності, незначущими нулями. Для кожної групи поставити відповідну позиційну цифру





шістнадцяткової системи числення згідно з таблицею 1. Читаючи отримані цифри зліва направо, ми отримуємо подання числа в новій системі числення.

### 2.3.2 Запис нечислової інформації в двійковій системі числення

Будь-яка нечислова інформація також може бути закодована двійковим кодом за допомогою двох цифр: 0 і 1. Покажемо, як закодувати, наприклад, текст будь-якого повідомлення, складеного українською мовою. Український телеграфний алфавіт містить 31 літеру (не розрізняються “і” та “ї”). Враховуючи ще пропуск між словами, маємо 32 символи, тобто 25. Отже, кожен символ можна позначити п’ятизначним двійковим числом.

#### Приклад

А-00000	З-01000	О-10000	Ц-11000
Б-00001	И-01001	П-10001	Ч-11001
В-00010	Ї-01010	Р-10010	Ш-11010
Г-00011	Й-01011	С-10011	Щ-11011
Д-00100	К-01100	Т-10100	Ь-11100
Е-00101	Л-01101	У-10101	Ю-11101
Є-00110	М-01110	Ф-10110	Я-11110
Ж-00111	Н-01111	Х-10111	пропуск-11111

Тоді фраза “цікава наука” у такому кодї має вигляд:

11000 01010 01100 00000 00010 00000 111111 01111 00000  
10101 01100 00000

Над двійковими числами, так само як і над числами, записаними в будь-якій СЧ, можна виконувати арифметичні операції: додавання, віднімання, множення і ділення. Додавання двійкових чисел аналогічне додаванню десяткових. Починають цю операцію з додавання двійкових цифр у самих молодших розрядах доданків. Якщо результат додавання більший 1, то відбувається перенесення до наступного розряду — і так доти, доки не дістануть остаточну суму.

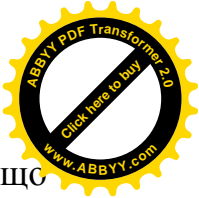
#### Двійкова арифметика

Таблиця додавання:

<b>1 + 1 = 10</b>	101	111
<b>0 + 1 = 1</b>	+	+
<b>1 + 0 = 1</b>	1	11
<b>0 + 0 = 0</b>	110	1010

#### Двійкова логіка

В обчислювальній техніці та програмуванні широко застосовуються основні поняття *алгебри логіки*.



Об'єктами алгебри логіки є оповідні речення, про які можна сказати, що вони або істинні, або хибні. Такі речення називаються *простими висловлюваннями* і позначаються великими латинськими літерами ( $A, B, D$ )

Приклад

Київ — столиця України. (Істинне висловлювання  $A=1$ .)

$13 > 27$  (Хибне висловлювання  $B=0$ ).

Прості висловлювання об'єднуються в складні висловлювання за допомогою логічних операцій:

- логічного додавання;
- логічного множення;
- логічного заперечення.

Логічним множенням двох висловлювань  $A$  і  $B$  називається їх об'єднання в одне за допомогою сполучника “И”

Логічний елемент “И” (and)

$A$	$B$	$A$ and $B$
1	1	1
1	0	0
0	1	0
0	0	0

Логічним додаванням двох висловлювань  $A$  і  $B$  називається їх об'єднання в одне за допомогою сполучника “ИЛИ”

Логічний елемент “ИЛИ” (or)

$A$	$B$	$A$ or $B$
1	1	1
1	0	1
0	1	1
0	0	0

Приєднання частки “НЕ” до присудка простого висловлювання а називається логічним запереченням.

Логічний елемент “НЕ” (not)

$B$	not $B$
1	0
0	1
1	0
0	1

Логічні елементи “И”, “ИЛИ” та “НЕ” об'єднуються в більш складні схеми

На базі елемента двійкової логіки “И-НЕ” можна створити найпростіший пристрій для збереження двійкової цифри — *тригер*. Декілька тригерів об'єднуються в єдину схему, яка призначається для збереження двійкового числа. Такий пристрій називається *регістром*.





### 3 СИСТЕМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ

Операційна система WINDOWS-XP, розроблена фірмою MICROSOFT.

Це операційна система, в якій реалізовані такі можливості:

- розвинений інтерфейс користувача. Призначений для користувача інтерфейс - це командна мова для управління функціонуванням комп'ютера і набір сервісних послуг, що звільняють користувача від виконання рутинних операцій. Інтерфейс користувача вважається розвиненим, якщо він:
  - 1) графічний (виводить інформацію на екран монітора не посимвольно, а поточечно);
  - 2) інтуїтивно-зрозумілий;
  - 3) об'єктно-орієнтований (тобто в його основі знаходяться об'єкти, що взаємодіють між собою);
  - 4) документо-орієнтований (тобто увага користувача концентрується не на програмах, які він може використати в своїй діяльності, а на документах, з якими працює).

Розвиненим інтерфейсом користувача забезпечується: суворі ієрархія доступних користувачеві об'єктів і ресурсів; автоматизація складних робіт; розвинена система допомоги;

- багатозадачність, тобто здатність забезпечувати виконання декількох програм одночасно (можна одночасно формувати дискету, друкувати який-небудь текст, посилати повідомлення по факсу);
- багатопоточність - можливість організувати виконання програм, що поступають декількома паралельними потоками, і конкурентних між собою за процесорний час (наприклад, при роботі з програмою обробки електронних таблиць, яка побудована на принципах багатопоточності, можна перераховувати одну електронну таблицю і в той же час розпечатувати іншу);
- підтримка довгих імен файлів;
- вбудована підтримка роботи в мережі;
- підтримка стандарту Plug and Play, який дозволяє максимально спростити установку і настройку периферійних пристроїв;
- підтримка портативних комп'ютерів;
- поліпшена підтримка додатків мультимедіа, тобто додатків, що працюють зі звуком і відео;
- вбудована підтримка електронної пошти й факсимільного зв'язку;
- максимальне використання 32-розрядних модулів та інш.



### 3.1 Файли

Опишемо більш детально поняття файла й даних в WINDOWS-XP.

Під даними розуміється все те, що підлягає зберіганню - програми (в початковому або машинному вигляді), початкові дані для їх виконання, результати виконання програм, тексти, ілюстрації, бази даних, електронні таблиці і т.п. Якщо не вдаватися в деталі, всі дані можна поділити на програми (маються на увазі програми на машинній мові) і документи (вони створюються й модифікуються програмами).

Щоб забезпечити можливість роботи з групою взаємозв'язаних даних як з єдиним цілим, її зберігають як окремий файл (від англійського слова File - підшивка), якому дається деяке ім'я. Таким чином, файл - іменована цілісна сукупність даних на диску. Останнім часом все частіше програмний файл (файл з програмою) називають програмою, а файл з документом - просто документом. Особливих обмежень на ім'я файла операційна система WINDOWS-XP, на відміну від своїх попередників, не накладає. У ньому можна використати майже всі символи, що друкуються: літери (рядкові і прописні, латинські і російські), цифри, розділові знаки і пробіли. Великі і малі літери в іменах файлів вважаються однаковими.

Обмеження на імена файлів:

- 1) ім'я файла не може починатися або закінчуватися крапкою;
- 2) в імені файла не можна застосовувати такі символи: \, /, :, \*, ?, ", <, >, |;
- 3) довжина імені файла не повинна перевищувати 255 символів.

Після імені файла ставиться розширення його імені. Між іменем файла і його розширенням ставиться крапка. Незважаючи на те, що ім'я файла можна забезпечити будь-яким розширенням або не вказувати його взагалі, набагато краще користуватися трьохсимвольними розширеннями, що вже устоялися. Розширення вказує на характер інформації, що міститься в файлі.

Наприклад:

exe - файл з готовою до виконання програмою;

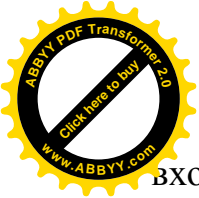
sys - системний файл;

doc - файл з текстовим документом в форматі редактора MS WORD та інш.

### 3.2 Папки

Структурування безлічі файлів забезпечується завдяки тому, що на дисках допускається створювання **папок**, в які і вкладаються файли. Звичайно папки об'єднують взаємозв'язані за якоюсь ознакою файли. При іменуванні папок використовуються ті ж правила, що і при іменуванні файлів, однак розширення імені не вказується. Виключно корисне те, що в одну папку можуть вкладатися інші папки. У результаті на дисках утворюються порою вельми розгалужені файлові структури.

Будь-який диск, що використовується в середовищі WINDOWS-XP, розглядається як папка. Ця папка називається головною папкою диска. Поняття папки аналогічно поняттю каталога в MS DOS. До складу імені папки диска



входить ім'я самого диска (*Наприклад* - A:, B:, C:).

Якщо поглянути на файловою структуру з точки зору підлеглості папок, стає зрозумілим, що вона є ієрархічною багаторівневою, якщо коротше - деревовидною файловою структурою, в корені якої знаходиться головна папка, а листям є файли і порожні папки, якщо такі існують. Дерево файлової структури має гілки, що "виростають" з непорожніх папок. Якщо одна папка безпосередньо міститься в іншій, то перша називається дочірньою по відношенню до другої папки, а друга - батьківською по відношенню до першої папки.

Концепція папок отримала в WINDOWS-XP подальший розвиток - файлові структури всіх доступних дисків, а також деякі пристрої вводу-виводу (зокрема, принтери) об'єднуються в єдине дерево ресурсів. Вважається, що самою головною папкою є папка **"Робочий стіл"**, що показується на екрані після завантаження системи.

Для однозначної ідентифікації файла (папки) необхідно вказати ланцюжок імен вкладених одна в одну папок, починаючи від головної папки диска, до папки, що містить цей файл (папку). Імена папок в цьому ланцюжку розділяються між собою символом "\". Потім в кінці цього ланцюжка ставиться ще один роздільник "\", після якого вказується ім'я файла (папки), що ідентифікується. Отриманий таким чином ланцюжок імен папок і імені файла (папки), що ідентифікується, називають повним ім'ям (або специфікацією) файла (папки).

*Наприклад*, C:\LETO\adc.doc - повне ім'я текстового файла adc.doc, що знаходиться на диску C:, в папці LETO.

У одній папці не можуть знаходитися файли (папки) з однаковими іменами, але в різних папках можуть міститися файли (папки), що мають одне і те ж ім'я. Тільки повне ім'я файла (папки) однозначно визначає його (її) серед інших.

Папка **"Робочий стіл"** - материнська для всіх інших, вона має ряд особливих властивостей. Так, наприклад, папка **"Робочий стіл"** знаходиться під всіма іншими вікнами і видна повністю, коли всі вікна згорнені. Ця папка завжди відкрита і зберігається в папці WINDOWS системного диска (як правило, це диск з ім'ям C:).

Для зображення вмісту папок використовується два типи вікон, а саме, вікно папки і вікно **"Проводника Windows"** (робота з "Проводником WINDOWS" буде розглянута нижче).

### 3.3 Запуск WINDOWS - XP

ОС WINDOWS - XP завантажується автоматично при вмиканні комп'ютера. Після її завантаження на екрані з'являється зображення, показане на рис. 3.1, що іменується **"Робочим столом"** WINDOWS-XP.

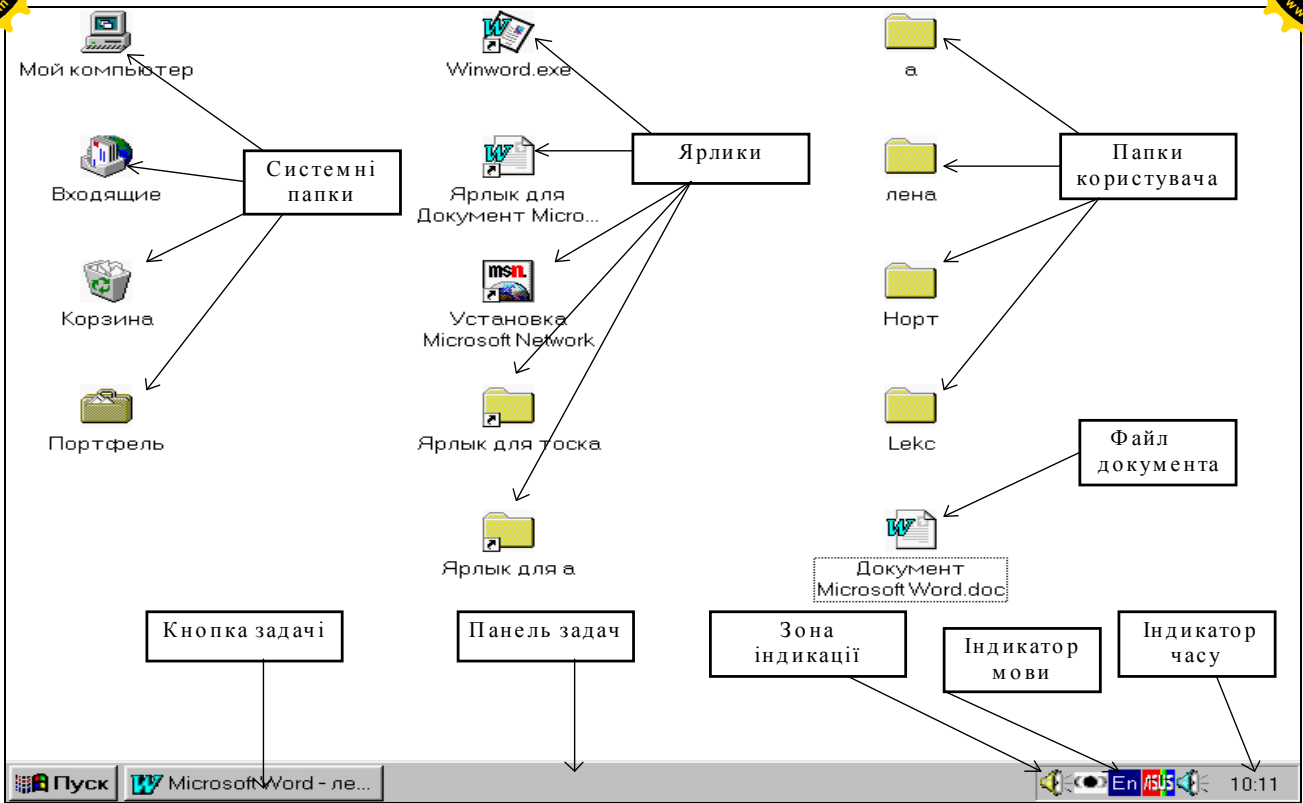


Рис. 3.1- "Робочий стол" WINDOWS-XP

У нижній частині "Робочого стола" розташовується **"Панель задач"** - основний засіб взаємодії користувача з системою, який дозволяє легко запуснути будь-який додаток на виконання або відкрити потрібний документ.

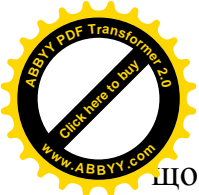
У лівому верхньому кутку "Робочого стола" знаходяться значки об'єктів (значок - це маленька ілюстрація, що допомагає швидко ідентифікувати об'єкт, з яким він пов'язаний): системних папок ("Мой компьютер", "Корзина", "Входящие" й інш.); папок користувача; ярликів; файлів документів; програмних файлів. Значок ярлика відрізняється від інших значків наявністю в його нижньому лівому кутку маленької стрілки. *Визначення перерахованих об'єктів буде дане нижче.*

### 3.4 Робота з мишею

Основна кнопка миші - ліва, додаткова - права. Якщо в тексті не конкретизована кнопка миші, то мається на увазі основна.

Для того, щоб вибрати (виділити) якийсь об'єкт, необхідно розташувати покажчик миші над цим об'єктом, потім один раз натиснути і відпустити основну кнопку миші. У результаті значок об'єкта буде пофарбований іншим кольором (частіше за все темно-синім).

Подвійне натиснення кнопки миші складається в швидкому послідовному подвійному її натисненні. Для того, щоб "перетягнути" той або інший об'єкт всередині вікна папки з одного місця на інше, необхідно вмістити на нього покажчик миші, натиснути основну кнопку і, не відпускаючи її, пересунути мишу в потрібне місце, потім відпустити кнопку миші. Для того,



щоб вибрати мишею фрагмент тексту, необхідно підвести покажчик миші до його початку, натиснути основну кнопку і, не відпускаючи її, перемістити покажчик миші до кінця фрагмента. У кінці фрагмента відпустити кнопку миші.

### 3.5 Основні елементи "Панели задач"

Зліва на **"Панели задач"** знаходиться кнопка **"Пуск"**, що дозволяє викликати на екран **Головне меню** (його опис буде даний нижче). У центрі **"Панели задач"** можуть розташовуватися *кнопки програм, що працюють* або *відкритих папок*. Натиснення однієї з цих кнопок мишею робить вікно папки або задачі активним (тобто вікно, пов'язане з цією кнопкою, розташується на "Робочому столі" поверх інших вікон).

У правій частині **"Панели задач"** знаходиться *зона індикації*, в якій відображаються показання системного годинника (поточний час) і різні додаткові індикатори, що сповіщають користувачу про режими роботи й стан пристроїв вводу-виводу і програмних засобів. Якщо затримати на мить покажчик миші над індикатором годинника, то на екрані з'явиться поточна дата. За допомогою таких індикаторів вдається виконати різного роду оперативні настройки, тому зона індикації щось більше, ніж індикатори. Для настройки годинника необхідно клацнути двічі кнопкою миші по його індикатору і встановити потрібні параметри в діалоговому вікні, що з'явиться. Серед додаткових індикаторів можуть бути такі індикатори:

- індикатор клавіатури (мови), що дозволяє дізнатися, на якому з національних реєстрів (російському або латинському) клавіатура в даний момент знаходиться, а також забезпечує перехід з одного реєстра на інший. Якщо клацнути мишею по індикатору мови, то з'явиться перелік підключених мов. Можна активізувати будь-яку з них, клацнувши по ній мишею;
- індикатор регулювальника гучності звукової плати, за допомогою якого можна змінити рівень гучності і відключити звук, а також звернутися до всіх функцій програми Volume Control й інш.

### 3.6 Головне меню

Натиснення мишею кнопки **"Пуск"**, що знаходиться на "Панелі задач", приводить до появи на екрані **"Головного меню"** (рис. 3.2).

Щоб вибрати потрібний пункт меню, досить одного натиснення миші на цьому пункті. "Головне меню" здійснює швидкий доступ до всіх програм і документів. Воно також дозволяє здійснити пошук програм і документів на дисках, настроїти зовнішній вигляд **"Робочого стола"**, отримати довідкову інформацію, здійснити вихід з системи.

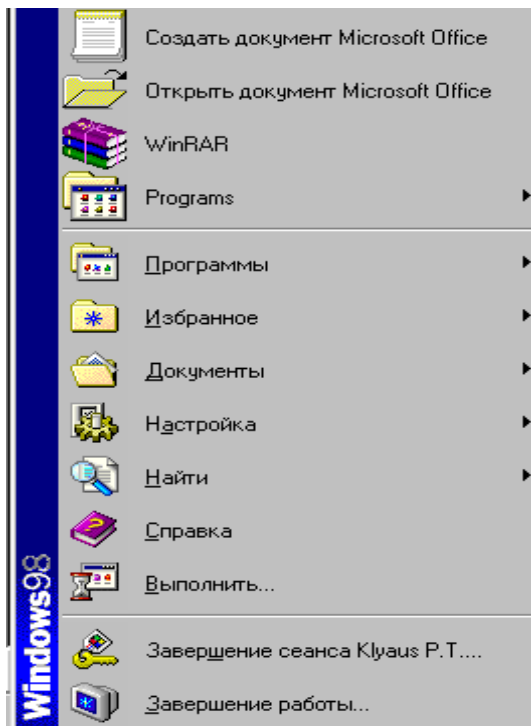




Рис. 3.2 - Пункты Головного меню"

### 3.7 Вихід з системи WINDOWS - XP

Для виходу з ОС Windows-XP необхідно виконати такі дії:

- клацнути мишею по кнопці **"Пуск"** для виклику Головного меню;
- в Головному меню виконати команду **"Завершение работы"**;
- у вікні, що з'явилося (рис. 3.3), виконати команду **"Выключить компьютер"** (пункт активний, якщо перемикач перед ним має вигляд  і неактивний, якщо він має вигляд . Змінити режим на протилежний можна одним натисненням миші по перемикачу, що стоїть перед ним);
- підтвердити команду, клацнувши мишею по кнопці **Да**.

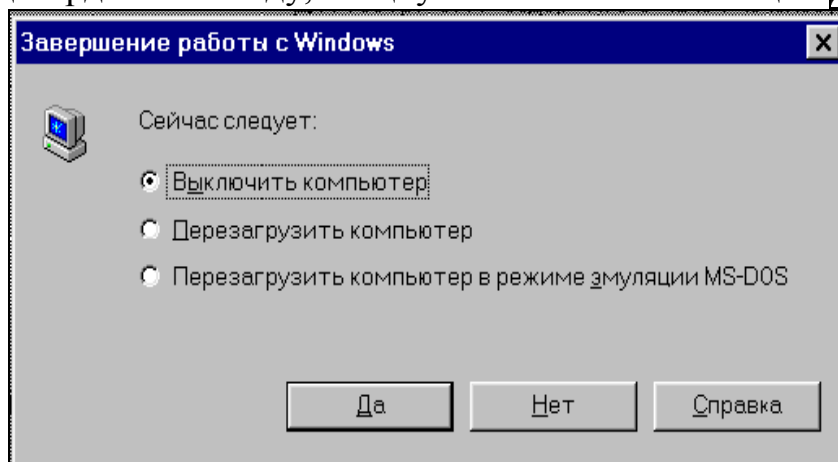


Рис. 3.3 - Завершення роботи з WINDOWS - XP



### 3.8 Відкриття папки

Подвійне натиснення по значку папки відкриває її, після чого на екрані з'являється вікно цієї папки. У робочій частині цього вікна виводяться значки всіх об'єктів (папок, файлів, ярликів), що знаходяться всередині неї.

### 3.9 Папка "Мой компьютер"

Папка *"Мой компьютер"* - системна папка, що дозволяє ознайомитися зі станом ресурсів ПК. Після відкриття цієї папки на екрані з'явиться вікно, показане на рис. 3.4. У цьому вікні будуть присутні по одному значку для кожного дисководу, жорсткого або гнучкого дисків. Крім того, в ньому можуть бути присутні додаткові системні папки "Панель управління", "Принтеры" і "Программа удалённого доступа к сети", призначена для зв'язку з мережевими ресурсами по телефонних каналах за допомогою модему.

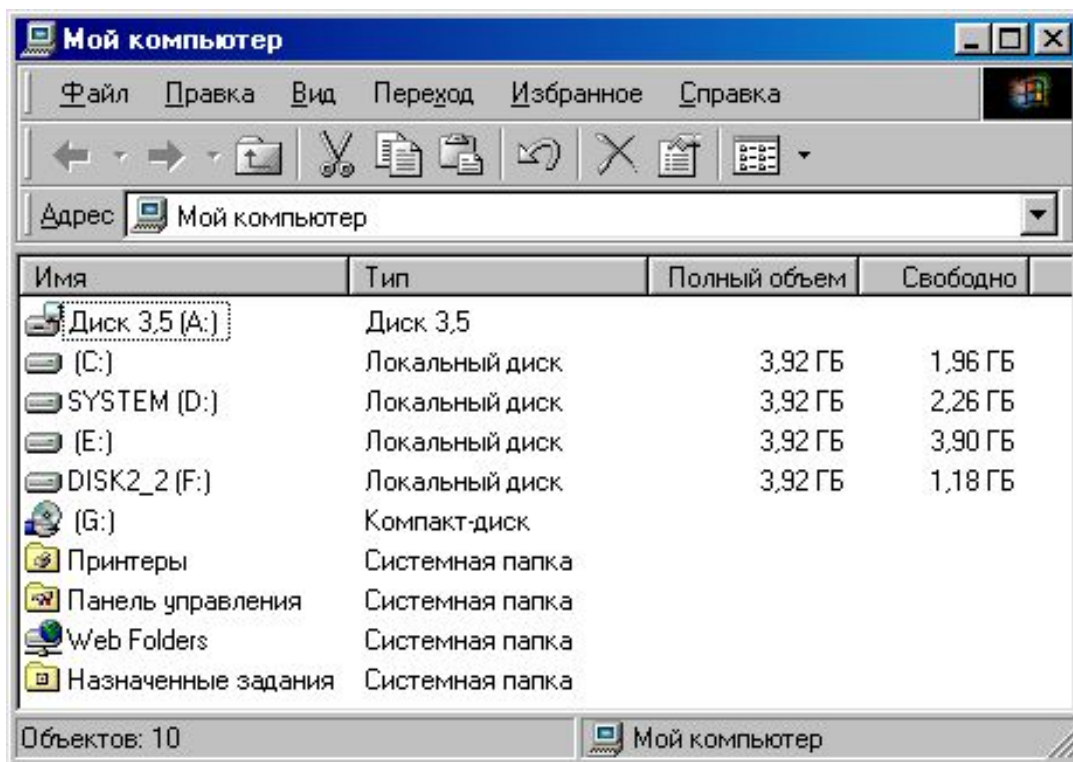


Рис. 3.4 - Вікно папки "Мой компьютер"

### 3.10 Папка "Корзина"

Системна папка *"Корзина"* - це місце тимчасового зберігання видалених з жорстких дисків файлів. Якщо якийсь з файлів видалений випадково, то його можна відновити шляхом витягання його з папки "Корзина" простою командою, яка буде розглянута далі. У папці "Корзина" запам'ятовується ім'я, початкове місцезнаходження, дата видалення, тип і розмір видаленого файла. Обсяг пам'яті для папки "Корзина" виділяється на кожному жорсткому диску в відсотках від його обсягу.



### 3.11 Ярлики

Ярлик - це файл, що містить покажчик (посилання) на об'єкт (інший файл, папку або периферійний пристрій, наприклад, принтер). Він забезпечує швидкий доступ до відповідного об'єкта. Можна створити ярлики для об'єктів практично всяких типів, що часто використовуються. Частіше за все ярлики розміщують на "Рабочем столе". Створення ярлика не означає створення копії об'єкта, на який ярлик посилається.

Для одного й того ж об'єкта можна створити необмежену кількість ярликів, в тому числі і в одній папці, але тоді вони повинні називатися по-різному.

Іменуються ярлик за тими ж правилами, що і файл, але йому приписується стандартне розширення LNK (від LiNK - зв'язок). Проте, саме розширення побачити за допомогою високорівневих засобів WINDOWS-XP неможливо. Ім'я ярлика може бути будь-яким - воно може навіть співпадати з ім'ям того об'єкта, на який ярлик посилається.

**!!! При переміщенні об'єкта в іншу папку посилання в ярликах для нього автоматично не поновлюються, тому доступ до переміщеного об'єкта за допомогою ярликів безпосередньо забезпечуватися не буде.**

*При пересилці, копіюванні і видаленні ярликів дії виконуються тільки над ярликами - об'єкти, на які вони посилаються, залишаються незмінними.*

*Таким чином, не треба вважати, що ярлик повністю замінює об'єкт, на який посилається, - він розглядається як заміник об'єкта тільки при виконанні з ним операцій доступу, зокрема, операції відкриття, так і те не завжди. У разі інших операцій, частина з яких вже названа, він вважається самостійним, нічим не відмінним від інших файлів.*

### 3.12 Меню об'єкта

Якщо клацнути правою кнопкою миші на якомусь об'єкті або групі виділених об'єктів, то з'явиться меню команд, відповідних даному об'єкту або групі об'єктів. Наприклад, праве клацання на "Панели задач" виводить меню команд, що відносяться тільки до "Панели задач".

Одним з пунктів меню об'єкта є пункт "**Свойства**". Ця команда дозволяє переглядати й змінювати властивості об'єктів.

Одна з команд меню об'єкта зображається півжирним кольором. Таке виділення відмічає ту команду, яка виконується автоматично при подвійному натисненні по об'єкту.

### 3.13 Робота з вікнами

#### 3.13.1 Елементи вікна

Всі додатки Windows працюють в прямокутних областях, що називаються вікнами. Типове вікно Windows представлено на малюнку 3.5.

Вікна містять такі елементи: границі; рядок заголовка; значок системного меню; кнопку, що закриває; кнопки скрутити, відновити, розвернути; рядок меню; панелі інструментів; робочу область; смуги прокрутки.



Границі - це лінії, що обмежують вікно з чотирьох сторін.

Заголовок знаходиться у верхній частині вікна і містить його назву.

Значок системного меню знаходиться зліва в рядку заголовка. Натиснення по ньому відкриває команди управління вікном, за допомогою яких можна переміщувати це вікно і змінювати його розміри.

Кнопка, що закриває розташована біля правої границі заголовка вікна і позначена хрестиком . Натиснення по цій кнопці закриває документ (папку) або завершує роботу додатку.

Кнопки скрутити, відновити і розвернути. Лівіше кнопки, що закриває знаходяться кнопки:

- Скрутити;
- Відновити;
- Розвернути.

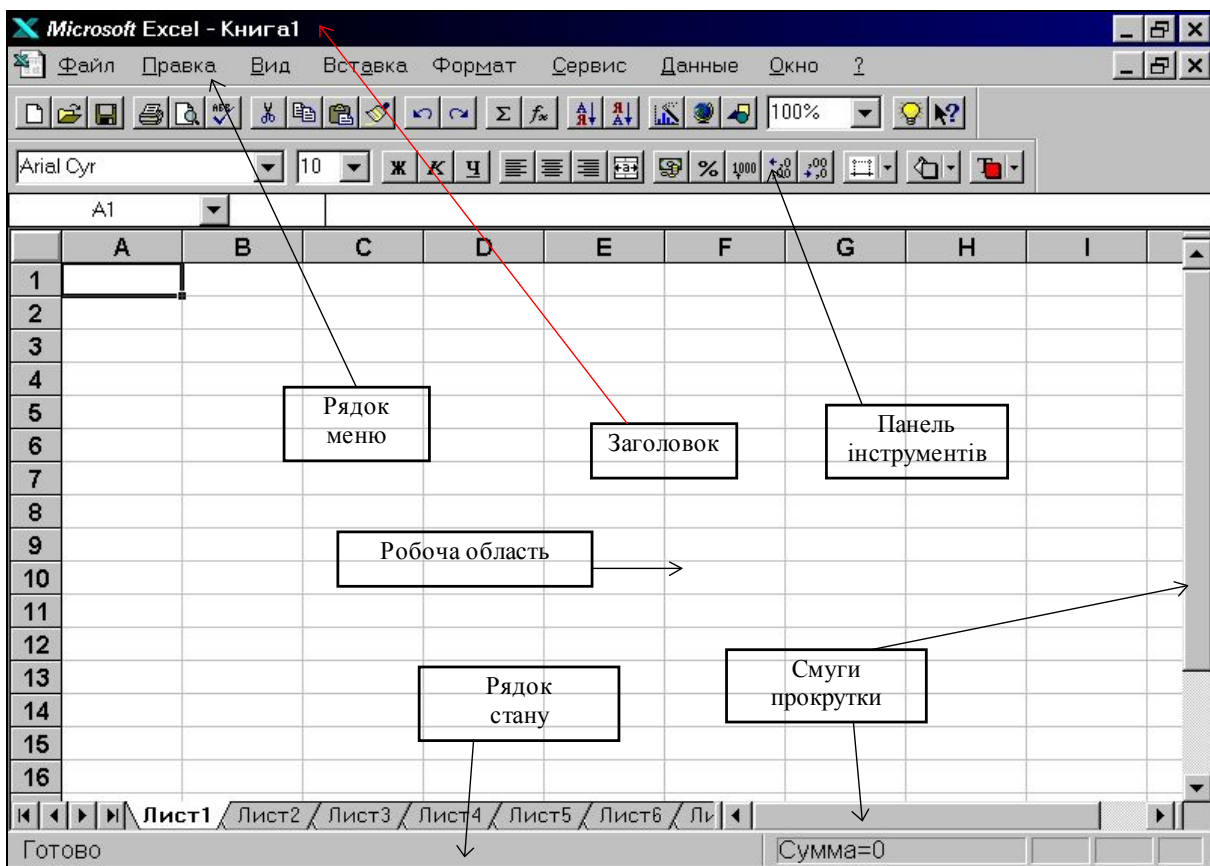
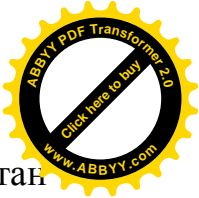


Рис. 3.5 - Типове вікно додатку WINDOWS – XP

Натиснення по кнопці **Скрутити** прибирає вікно з "Робочого стола", залишаючи лише кнопку на "Панелі задач", при цьому додаток залишається відкритим і продовжує виконуватися.

Кнопка **Розвернути** збільшує вікно до розмірів екрана. Це корисно при роботі з додатками і при перегляді папок, щоб використати всю корисну площу екрана.



Кнопка **Відновити** переводить вікно в початковий стан. Цей стан застосовують, коли необхідно побачити на екрані інші відкриті вікна.

Розвернути вікно на весь екран можна іншим способом - подвійним натисненням на його заголовок.

Якщо вікно вже займає весь екран, то подвійне натиснення по його заголовку поверне його в колишній стан.

Рядок меню. Під заголовком розташовується рядок меню. Він забезпечує доступ до більшості команд даного додатку або відкритої папки.

Панель інструментів. Це набір значків і кнопок, призначених для швидкого доступу до команд додатку, що найчастіше використовуються.

Робоча область. Це внутрішня область вікна.

Смуги прокрутки. Вертикальна й горизонтальна смуги прокрутки розташовуються в правій (вертикальна) або в нижній (горизонтальна) частинах вікна. Вони з'являються в тому випадку, якщо інформація, що відображається у вікні, повністю в ньому не вміщується.

### 3.13.2 Типи вікон

Існують такі типи вікон - вікна додатків, вікна документів і діалогові вікна. **Вікна додатків** є вікнами програм або папок, їх можна переміщувати в будь-яке місце "Рабочего стола", розвертати на весь екран або згортати в кнопки на "Панелі задач". **Вікна документів** знаходяться всередині вікон додатків. Їх можна розвертати, згортати, відновлювати і змінювати розміри, але вони завжди залишаються в межах вікна додатку.

Якщо для виконання якої-небудь команди необхідні додаткові відомості від користувача, то при її виконанні з'являється **Діалогове вікно**, в якому можна заповнити поля введення або вибрати один з варіантів, що пропонуються.

### 3.13.3 Зміна розмірів вікон і переміщення вікон

Для того, щоб змінити розміри вікна, необхідно підвести покажчик миші до його границь (покажчик миші повинен прийняти вигляд ↔ для вертикальних кордонів і ↓ для горизонтальних), натиснути кнопку миші і, утримуючи її, переміщувати границі в потрібну сторону. Аналогічно, потягнувши за один з кутів вікна, можна змінити його розміри в двох напрямках одночасно.

Щоб перемістити вікно, необхідно натиснути кнопку миші на його заголовок і, не відпускаючи кнопку, перетягнути вікно в потрібне місце екрана.

Дуже часто діалогове вікно закриває вікна інших додатків, що знаходяться під ним. Щоб побачити ці вікна, необхідно перемістити діалогове вікно, схопивши його за заголовок мишею, в інше місце екрана.

### 3.13.4 Перемикання між вікнами

Одночасно може бути відкрито декілька вікон. Вікно, розташоване над іншими, називається активним (переднім або верхнім). Тільки одне вікно може

бути активним. Для активізації неактивного вікна треба виконати одну з таких дій:

- клацнути мишею по кнопці цього вікна, розташованій на панелі задач;
- клацнути мишею по будь-якому видимому місцю цього вікна.

### 3.13.5 Упорядкування вікон на екрані

Для упорядкування вікон на "Рабочем столе" необхідно клацнути правою кнопкою миші на вільній частині "Панелі задач" (якщо панель задач не видна, то треба заздалегідь натиснути одночасно клавіші **Ctrl** + **Esc** і вибрати одну з команд: *"Каскадом"*, *"Слева направо"*, *"Сверху вниз"*. Внаслідок дії цих команд вікна на екрані розташуються, як це показано на рис. 3.6, 3.7, 3.8.

Для того, щоб скрутити всі активні вікна, треба зробити праве натиснення по панелі задач і вибрати команду *"Свернуть всё"*.

### 3.13.6 Системне меню й рядок меню

Вони являють собою два основних елементи системи меню додатків WINDOWS. Системне меню відкривається натисненням миші по її значку, який знаходиться в лівій частині заголовка вікна. Рядок меню розташовується під заголовком вікна. Він забезпечує доступ до більшості команд даного додатку (відкритої папки).

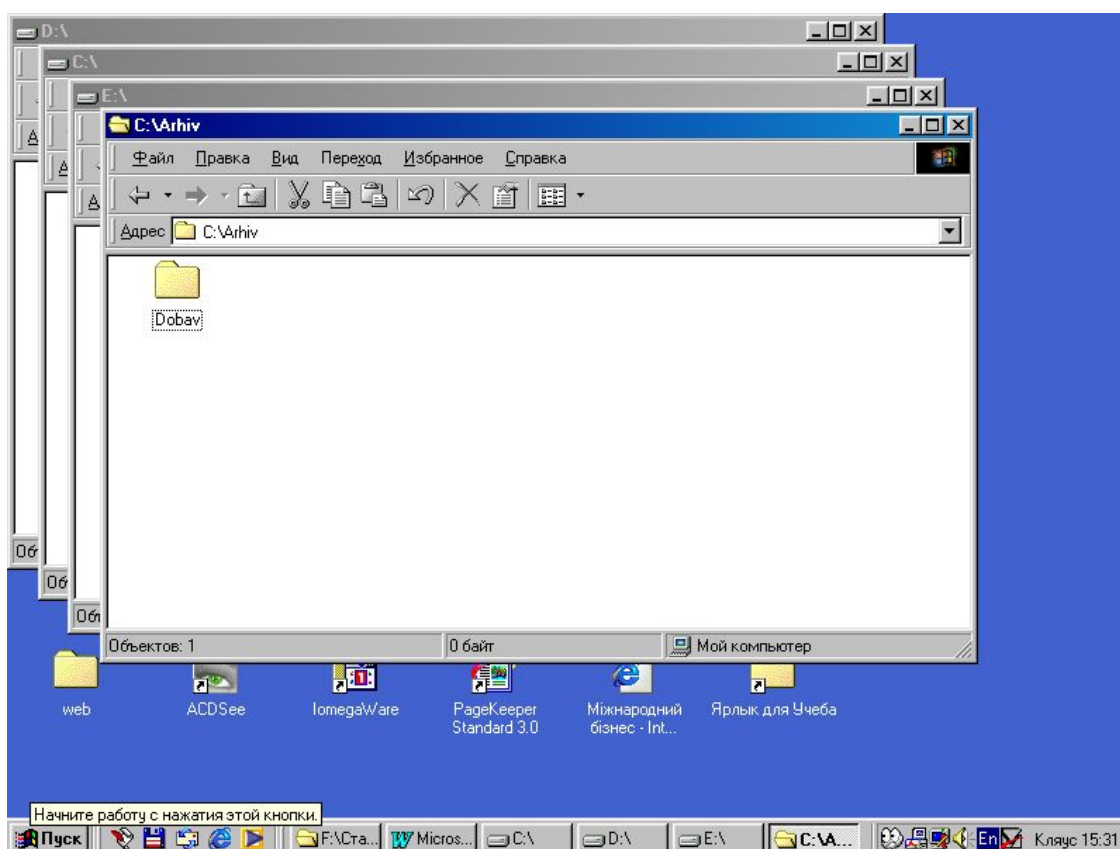


Рис. 3.6 - Розташування вікон "Каскадом"

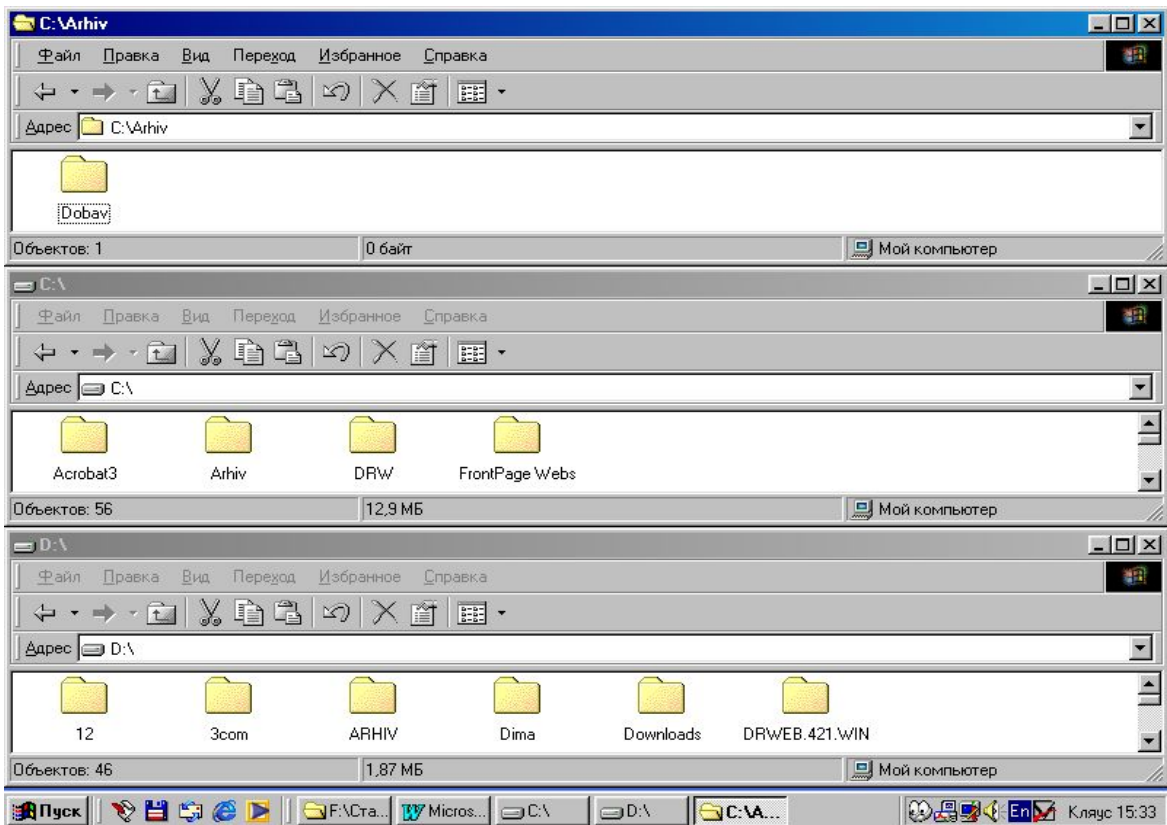


Рис. 3.7 - Розташування вікон "Сверху вниз"

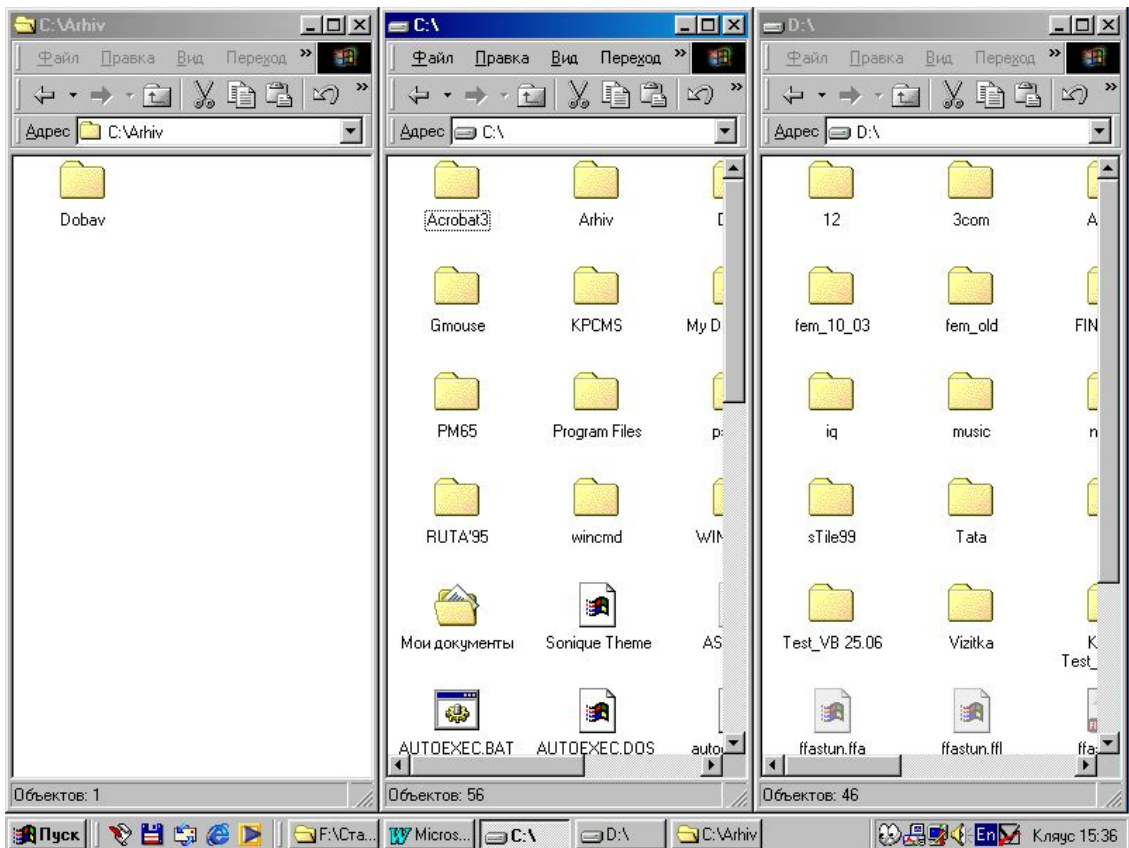


Рис. 3.8 - Розташування вікон "Слева направо"

Системне меню забезпечує базовий набір команд, загальний для всіх вікон (крім діалогових вікон), і дозволяє змінювати розміри вікна, переміщувати або закривати його. Команди системного меню, що найчастіше використовуються, винесені у вигляді кнопок в праву частину заголовка вікна.

Рядок меню включає команди, властиві даному конкретному додатку. Кожне слово рядка меню відкриває список команд, що відносяться до нього. Увійти в систему меню можна за допомогою миші, клацнувши нею по потрібному слову в рядку меню. Щоб виконати команду з відкритого меню, треба клацнути по ній мишею.

Крапки (...) після команди меню означає незавершеність фрази. Це означає, що при виконанні даної команди з'явиться діалогове вікно для введення користувачем додаткової інформації.

Якщо немає необхідності виконання команд, що містяться у відкритому меню, то це меню можна закрити, клацнувши мишею в будь-якому місці за його межами або натиснувши клавішу **ESC**.

Команди, які зображені в меню сірим кольором, в даний момент недоступні.

Значок "галочка" біля команди меню означає, що режим підключений. Повторний вибір цієї команди відмінить або змінить режим.

Стрілка праворуч від команди означає, що при її виборі з'явиться підменю.

### 3.13.7 Пункт "Вид" рядка меню вікна папки

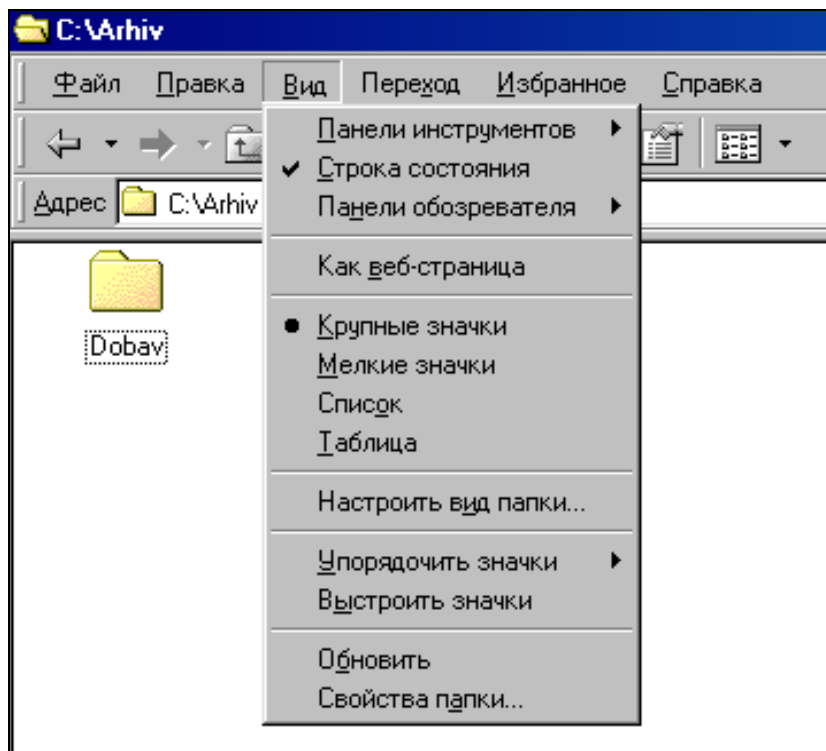


Рис. 3.9 - Меню "Вид" вікна папки





Команди меню "Вид" рядка меню вікна папки (рис. 3.9) дозволяють:

- підключити (відключити) панель інструментів (команда "*Панель інструментов*");
- підключити (відключити) рядок стану (команда "*Строка состояния*");
- вибрати форму представлення значків об'єктів в папці шляхом виконання однієї з взаємовиключаючих команд: "*Крупные значки*", "*Мелкие значки*", "*Список*", "*Таблица*";
- упорядкувати значки об'єктів в папці (команда "*Упорядочить значки*");
- відновити відображення вмісту папки (команда "*Обновить*");
- указати режими перегляду і відображення об'єктів в папці (команда "*Свойства папки*") і інш.

### 3.13.8 Робота з діалоговими вікнами

При виконанні деяких команд з'являється *Діалогове вікно*, в якому можна заповнити поля вводу або вибрати один з пропонованих варіантів. Діалогові вікна, на відміну від інших типів вікон, не мають системного меню, рядка меню, кнопок "розвернути", "скрутити", "відновити". Крім того, вони завжди вміщуються тільки зверху інших вікон.

Діалогові вікна містять наступні елементи:

- текстові поля;
- списки;
- списки, що розкриваються;
- перемикачі;
- прапорці;
- лічильники;
- повзунки;
- командні кнопки;
- вкладки.

1) **Текстове поле** служить для введення і редагування інформації. Щоб ввести в це поле текст, необхідно клацнути у вікні мишею, в результаті з'явиться текстовий курсор. У місці знаходження курсора можна набирати текст. Після набору тексту необхідно натиснути клавішу **ENTER**.

2) **Список** - це набір значень, що пропонуються на вибір. Вибір одного з них здійснюється натисненням миші по ньому.

3) **Список, що розкривається**.  Після натиснення миші по цій кнопці з'явиться список значень, що пропонуються на вибір. З нього можна вибрати потрібний натисненням миші.

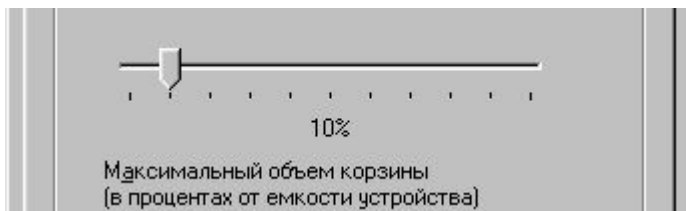
4) **Перемикачі**  (або кнопки вибору) служать для вибору тільки одного з варіантів.

5) **Прапорці**  або  . Прапорець використовується для підключення (активізації) або відключення якого-небудь режиму. Якщо всередині прапорця стоїть "галочка", то даний режим підключений, в іншому випадку ні. Натиснення на прапорець або встановлює галочку, або її відключає.

6) **Лічильник**   - це пара стрілок, які дозволяють збільшувати або зменшувати значення в пов'язаному з ним полі. Натиснення верхньої стрілки збільшує числове значення поля, нижньої - зменшує його. Значення поля можна змінити, не користуючись стрілками. Для цього треба клацнути мишею в цьому полі, і після появи в ньому текстового курсора набрати з клавіатури потрібне число.

7) **Повзунок** - це такий компонент діалогового вікна, який дозволяє (можливо, з деякою дискретністю) змінювати числове значення пов'язаної з ним величини. Як на вигляд, так і за функціями він подібний регулювальнику гучності або тембру. Змінити положення повзунка можна безпосереднім перетягненням його мишею на інше місце.

Наприклад:



8) **Командна кнопка**. Більшість діалогових вікон має командну кнопку, натиснення якої підтверджує правильність заповнення полів вікна, і кнопку, що дозволяє відмовитися від змін і покинути діалог. Частіше за все на цих кнопках пишеться "ОК" і "Отмена" відповідно. Багато які діалоги до того ж забезпечуються кнопкою "Справка" або кнопкою із зображенням знаку питання - ними варто клацнути у тому випадку, якщо ви сумніваєтеся в правильності своїх дій. Існує також і ряд інших командних кнопок.

9) **Вкладки**. У верхній частині діалогового вікна можуть знаходитися вкладки. Натиснення мишею вкладки відкриває її. Після цього користувач може встановити потрібні йому опції в цій вкладці та свої дії підтвердити натисненням мишею кнопки "ОК".

### 3.14 Робота з папками й файлами

#### 3.14.1 Вибір папок, файлів і ярликів у вікні папки (вибір об'єктів)

Вибір *одного об'єкта* здійснюється натисненням миші по його значку.

Вибір *групи об'єктів* можна здійснити одним з таких способів:

1. Натиснути клавішу  і, не відпускаючи її, послідовно клацати мишею по потрібних об'єктах.
2. Якщо об'єкти, що виділяються, слідуєть один за одним, то для прискорення виділення необхідно натисненням миші виділити перший



об'єкт, а потім, натиснувши клавішу **Shift** і, не відпускаючи її, клацнути мишею по останньому об'єкту.

3. Якщо значки групи об'єктів займають прямокутну область, то для їх виділення зручніше скористатися перетягненням лівою кнопкою миші. Для цього встановити покажчик миші біля одного з уявних кутів групи об'єктів, натиснути ліву кнопку миші і, не відпускаючи її, охопити, переміщуючи мишу, прямокутною областю групу об'єктів, що виділяються.
4. Використовуючи поєднання способів 1 - 3.

Вибір *всіх* об'єктів у вікні папки можна здійснити одночасним натисненням клавіш **Ctrl** + **A**.

### 3.14.2 Властивості папок і файлів

Щоб подивитися або змінити властивості папки (файла), треба клацнути по значку папки (файла) правою кнопкою миші і вибрати команду "**Свойства**".

Якщо папка відкрита, то для перегляду або виправлення її властивостей треба клацнути правою кнопкою миші в будь-якому вільному місці вікна папки або по значку її системного меню і вибрати команду "**Свойства**".

Для вибраної групи об'єктів можна подивитися їх загальні властивості. Для цього треба клацнути правою кнопкою миші на будь-якому з вибраних об'єктів і в меню об'єкта вибрати команду "**Свойства**".

### 3.14.3 Створення нових папок

Створити нову папку можна такими способами:

- Відкрити папку, в якій створюється нова папка. Потім клацнути правою кнопкою миші в будь-якому вільному місці вікна відкритої папки і виконати команду "**Создать**". У меню, що з'явилося, вибрати команду "**Папка**". Значок нової папки з'явиться у вікні. Набрати з клавіатури ім'я нової папки. Підтвердити набране ім'я натисненням клавіші **ENTER**.
- Відкрити папку, в якій створюється нова папка. У рядку меню вікна відкритої папки клацнути мишею по пункту "**Файл**", потім виконати команду "**Создать**", а потім команду "**Папка**". Набрати ім'я нової папки і натиснути клавішу **ENTER**.

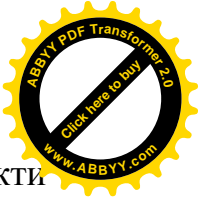
### 3.14.4 Переміщення й копіювання папок, файлів і ярликів (об'єктів)

Переміщення і копіювання об'єктів можна здійснювати або "лівим" або "правим" перетягненням кнопкою миші, або використовуючи "Буфер обміна" ("Буфер обміна" - це тимчасове сховище даних).

Щоб перемістити або скопіювати об'єкти методом перетягнення правою кнопкою миші, необхідно виконати таку послідовність дій:

- 1) відкрити папку, в яку треба копіювати або переміщувати об'єкти (папка-приймач);





- 2) відкрити папку, з якої треба копіювати або переміщувати об'єкти (папка-джерело);
- 3) розташувати вікна папок так, щоб вони були добре видні;
- 4) в папці-джерелі виділити переміщувані об'єкти або об'єкти, що копіюються;
- 5) зробити праве натиснення на будь-кому з виділених об'єктів і, не відпускаючи кнопки миші, перетягнути об'єкти в папку-приймач, після чого відпустити кнопку миші. У меню, що з'явиться, виконати команду **"Копировать"** (якщо об'єкти копіюються) або команду **"Переместить"** (якщо об'єкти переміщуються).

Щоб перемістити або скопіювати об'єкти, використовуючи метод перетягнення лівою кнопкою миші, необхідно здійснити таку послідовність дій:

Виконати пункти 1)-4) "правого перетягнення". Аналогічно виконати пункт 5), за винятком того, що на виділених об'єктах здійснюється натиснення і перетягнення не правою, а лівою кнопкою миші. У цьому випадку, після відпущення кнопки миші, меню об'єкта не з'являється. Внаслідок виконаних дій відбувається наступне:

- 1) При "перетягненні" файла-програми з однієї папки в іншу програма залишається на місці, а в новій папці створюється її ярлик.
- 2) При "перетягненні" файла будь-якого іншого типу з папки в папку в межах одного диска відбувається його переміщення.
- 3) При "перетягненні" файла з папки одного диска в папку іншого диска відбувається його копіювання.

Якщо натиснути клавішу **[Ctrl]**, то при "лівому перетягненні" об'єкт копіюється незалежно від його типу.

Якщо натиснути клавішу **[Shift]**, то при "лівому перетягненні" об'єкта він переміщується.

### **Зауваження!!!**

- 1) Якщо папка, в яку "перетягується" об'єкт закрита, то "перетягнення" можна здійснювати на її значок.
- 2) Якщо на робочому столі є ярлик деякої папки, то можна перетягнути об'єкт на ярлик папки.

Щоб перемістити або скопіювати об'єкти, використовуючи "Буфер обміна", необхідно виконати такі дії:

- 1) виділити об'єкти;
- 2) зробити праве натиснення на будь-якому виділеному об'єкті;
- 3) якщо об'єкти копіюються, то виконати команду **"Копировать"**, а якщо об'єкти переміщуються, то команду **"Вырезать"** (команди **"Копировать"** і **"Вырезать"** вміщують копії об'єктів в "Буфер обміна");
- 4) виконати праве натиснення в папці, в якій повинні з'явитися переміщувані об'єкти або об'єкти, що копіюються, а потім в меню об'єкта, що з'явилося - команду **"Вставить"**.



### 3.14.5 Перейменування папок, файлів і ярликів (об'єктів)

Для перейменування об'єкта необхідно клацнути правою кнопкою миші по його значку і виконати в меню об'єкта команду **"Переименовать"**. Потім відразу ж набрати нове ім'я з клавіатури і натиснути клавішу **ENTER**.

### 3.14.6 Видалення папок, файлів і ярликів (об'єктів)

Для видалення об'єктів необхідно вибрати об'єкт або групу об'єктів, а потім виконати одну з вказаних далі дій:

- Натиснути клавішу **Del**. Підтвердити видалення.
- Клацнути правою кнопкою миші на будь-кому з виділених об'єктів і в меню об'єкта, що з'явилося, виконати команду **"Удалить"**. Підтвердити видалення.
- Перетягнути виділені об'єкти лівою кнопкою миші на значок або у відкрите вікно папки **"Корзина"**. Підтвердити видалення.

### 3.14.7 Відновлення видалених файлів

Для відновлення видалених файлів необхідно відкрити папку **"Корзина"**, виділити файл або групу файлів, що відновлюються, клацнути правою кнопкою миші на будь-якому з виділених файлів, в меню об'єкта, що з'явиться, вибрати команду **"Восстановить"**. Ця команда перенесе вибраний файл або групу файлів в ту папку, з якої він був видалений. Якщо була видалена разом з файлами і папка, що їх містила, то при відновленні хоч би одного файла з цієї папки здійснюється створення цієї папки на колишньому місці.

Потрібно пам'ятати: об'єкти, видалені з дискет, не попадають в папку **"Корзина"!!!**

Якщо видаляється папка, то в папку **"Корзина"** вміщуються всі файли, що знаходяться в ній, але не сама папка.

### 3.14.8 Очищення папки "Корзина"

Щоб видалити декілька об'єктів з папки **"Корзина"**, треба виконати таку послідовність дій:

- 1) виділити об'єкти, що видаляються;
- 2) виконати праве натиснення на будь-кому з виділених об'єктів, а потім виконати з меню об'єкта команду **"Удалить"**. Підтвердити видалення.

Щоб очистити усю папку **"Корзина"**, треба клацнути по значку папки **"Корзина"** правою кнопкою миші і виконати в меню об'єкта команду **"Очистить корзину"**. Підтвердити видалення.

Видалені з папки **"Корзина"** об'єкти відновити вже неможливо.

### 3.14.9 Атрибути папок, файлів (об'єктів)

Об'єкти можуть мати такі атрибути:

- архівний;
- прихований;



- системний;
- тільки для читання.

Атрибут "архівний" вказує на те, що в файл були внесені зміни за час, який пройшов з моменту виконання останньої операції резервного копіювання. Кожний раз, коли ви створюєте новий файл або змінюєте вміст файла, що вже існує, WINDOWS встановлює для нього цей атрибут. Спеціальні програми резервного копіювання звичайно обнуляють цей атрибут в процесі архівування. Якщо будуть внесені зміни в файл після резервного копіювання, атрибут "архівний" встановлюється знову - для того, щоб програма резервного копіювання розпізнала цей файл як такий, що вимагає повторного збереження.

Атрибут "прихований" дозволяє зробити об'єкт прихованим. Приховані об'єкти, якщо не вжити спеціальні заходи, у вікнах папок можуть не показуватися.

Атрибут "системний" встановлюється для системних об'єктів, тобто об'єктів, що належать ОС.

Атрибут "тільки для читання" встановлюється для файлів. Такий файл не може бути видалений або змінений без вживання спеціальних заходів.

Щоб подивитися або змінити атрибути файла або папки, треба клацнути правою кнопкою миші по значку об'єкта, виконати в меню об'єкта команду "Свойства", в діалоговому вікні, що з'явиться, встановити потрібні або зняти непотрібні атрибути, клацнувши кнопкою миші по відповідних прапорцях. Підтвердити зроблені установки натисненням миші по кнопці **OK** діалогового вікна.

### 3.14.10 "Проводник WINDOWS"

Насправді всі вікна папок підтримуються єдиним компонентом WINDOWS-XP - оболонкою, що відповідає за інтерфейс користувача і називається Windows Explorer (дослівний російський переклад – "Проводник WINDOWS").

Якщо дану обставину не враховувати, то вікна папок в окрему групу можна і не виділяти, вважаючи їх звичайними вікнами додатку.

Поняття Windows Explorer в операційній системі WINDOWS-XP неоднозначне, оскільки таку назву має як вбудована в систему оболонка, що реалізовує інтерфейс користувача, так і диспетчер файлів (в російському варіанті системи він називається "Проводник WINDOWS"), що є її частиною. З поняттям Windows Explorer в першому значенні кінцевий користувач не стикається. Тому далі в цьому посібнику під Windows Explorer розуміється тільки диспетчер файлів – "Проводник Windows".

До допомоги "Проводника Windows" доцільно прибігати, якщо в даний момент необхідно виконати цілу серію операцій з папками і файлами - наприклад, реорганізувати дерево папок.

Для запуску "Проводника WINDOWS " є декілька способів:

- в "Главном меню" вибрати пункт "**Программы**", а потім пункт

**"Проводник";**

- клацнути правою кнопкою миші по папці або ярлику і в меню об'єкта вибрати команду **"Проводник"**.

Вікно "Проводника" має дві області: ліву і праву (рис. 3.10).

Ліва область "Проводника WINDOWS " являє собою дерево папок, що знаходяться на комп'ютері, коренем дерева є папка "Рабочий стол". Щоб вибрати яку-небудь папку, треба клацнути мишею по її назві або її значку, водночас в правій області буде відображатися вміст цієї папки.

Знак  перед папкою означає, що всередині цієї папки є вкладені в неї папки.

Знак  означає, що папка відкрита або розгорнена.

Щоб відкрити (розгорнути) папку, необхідно клацнути кнопкою миші по значку  а згорнути - по значку  .

Якщо клацнути двічі по назві папки в лівій частині вікна "Проводника WINDOWS", то її вміст буде відображатися в правій області і, крім того, в лівій частині з'являться папки, що знаходяться всередині неї.

З правою і лівою областями вікна "Проводника WINDOWS" можна працювати як із звичайними папками WINDOWS.

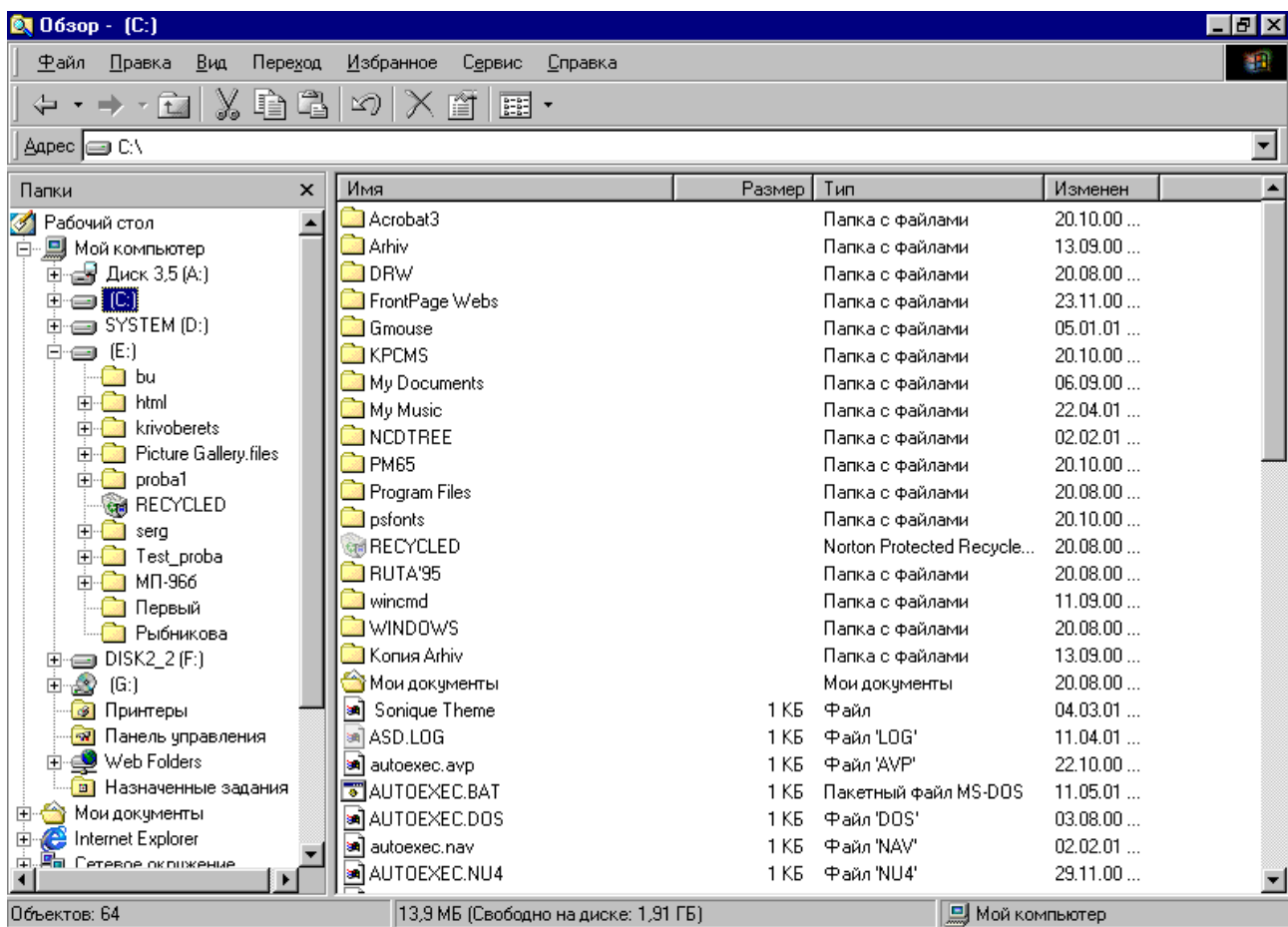


Рис. 3.10 - Вікно "Проводника WINDOWS"



### 3.14.11 Запуск програм на виконання

Здійснити запуск програм можна багатьма способами, зокрема:

- якщо у верхній частині Головного меню знаходиться назва потрібної програми, то для її запуску необхідно встановити на цю назву курсор миші і клацнути мишею;
- за допомогою команди **"Программы"** Головного меню. Команда "Программы" містить ярлики встановлених на комп'ютері програм. Для запуску будь-якого з них необхідно клацнути по її ярлику мишею;
- щоб запустити на виконання програму з якої-небудь папки, треба відкрити цю папку і двічі клацнути мишею по значку відповідного програмного файлу;
- для запуску на виконання програми за допомогою її ярлика, що знаходиться в якійсь папці, необхідно цю папку відкрити, а потім двічі клацнути мишею на значку ярлика;
- в Головному меню вибрати команду **"Выполнить"**. У діалоговому вікні **"Запуск программы"**, що з'явиться, необхідно вказати повне ім'я програмного файлу в текстовому полі **"Открыть"**, а потім клацнути мишею по кнопці **ОК**, що знаходиться в цьому вікні. Щоб не набирати повне ім'я програмного файлу з клавіатури, можна скористатися кнопкою **"Обзор"**. Натиснення миші по цій кнопці виводить на екран діалогове вікно **"Обзор"**. У списку **"Папка"** цього вікна, що розкриється, вибрати папку, що містить програмний файл. У робочій частині цього вікна з'являться значки об'єктів, що містяться у вибраній папці. Потім треба двічі клацнути мишею по значку програмного файлу. У результаті WINDOWS вміщує повне ім'я програмного файлу в рядку **"Открыть"** діалогового вікна **"Запуск программы"**. Для запуску на виконання вибраного програмного файлу клацнути мишею по кнопці **Ок** цього ж вікна;
- в пункті **"Поиск"** "Главного меню" вибрати команду **"Файлы и папки"**. У запиті, що з'явиться, вказати ім'я шуканого файлу, область пошуку, а потім клацнути по кнопці **"Найти"**. По знайденому програмному файлу клацнути двічі мишею для його запуску на виконання.

### 3.14.12 Створення ярликів

Для створення ярлика якого-небудь об'єкта (папки, файлу) існують такі способи:

1. Відкрити папку, що містить об'єкт, для якого необхідно створити ярлик. Клацнути по значку цього об'єкта правою кнопкою миші і, не відпускаючи її, перетягнути об'єкт в потрібне місце. Відпустити кнопку миші. У меню об'єкта виконати команду **"Создать ярлык"**.

2. Клацнути правою кнопкою миші в тому місці вікна, в якому необхідно створити ярлик файла або папки. У меню об'єкта виконати команду **"Создать"**, а потім команду **"Ярлык"**. У діалоговому вікні **"Создание ярлика"** в текстовому полі **"Командная строка"** з клавіатури набрати повне ім'я об'єкта, для якого створюється ярлик (для вказівки повного імені об'єкта можна скористатися кнопкою **"Обзор"**, як описано в пункті **"Запуск програм на виконання"** даного посібника). Потім клацнути мишею по кнопці **Далее** в діалоговому вікні **"Создание ярлика"**. У вікні **"Выбор названия программы"**, що з'явиться, в текстовому полі **"Укажите название ярлика"** з клавіатури набрати ім'я ярлика. Клацнути мишею по кнопці **Далее**. У діалоговому вікні, що з'явиться, можна вибрати натисненням миші вигляд значка ярлика, що створюється, а потім клацнути мишею по кнопці **Готово**, що знаходиться в цьому ж вікні.

### 3.15 Форматування дисків

Клацнути правою кнопкою миші по значку диска в папці **"Мой компьютер"**. Вибрати в меню об'єкта пункт **"Форматирование"**. Встановити у вікні, що з'явиться, (рис. 3.11) обсяг диска, необхідні режими, показані на рис. 3.11, і натиснути кнопку **"Начать"**.

Диск відформатується як системний, якщо активізувати режим **"Скопировать системные файлы"**.

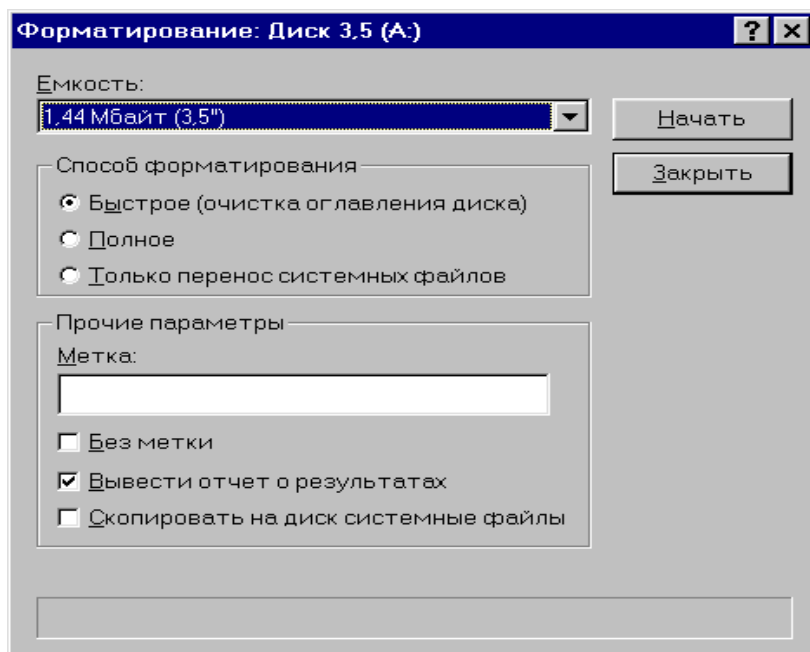


Рис. 3.12 - Діалогове вікно форматування дисків





### 3.16 Обмін даними

#### 3.16.1 Обмін даними за допомогою "Буфера обмена"

"Буфер обмена" - це тимчасове сховище даних. Його використання для копіювання і переміщення папок і файлів описане в цьому посібнику вище.

"Буфер обмена" можна використати також для копіювання або переміщення якої-небудь інформації з одного документа в інший. Для цього необхідно виконати такі дії:

1. Виділити в документі переміщуваний фрагмент, або фрагмент, що копіюється.
2. У меню **"Правка"** виконати команду **"Копировать"** або **"Вырезать"** (це ж можна виконати і командами **"Копировать"** або **"Вырезать"** з меню об'єкта, що викликається натисненням правої кнопки миші на виділеному фрагменті). При цьому команда **"Вырезать"** прибирає з документа цей фрагмент.
3. Перейти в інший документ, в який треба вставити фрагмент. Встановити курсор в місце вставки фрагмента. Вставити туди з "Буфера обмена" фрагмент командою **"Вставить"** з меню **"Вставка"** або з меню об'єкта, яке викликається правим натисненням в місці вставки фрагмента.

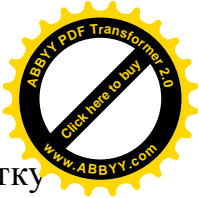
Якщо вставлений фрагмент створювався в іншому додатку, що підтримує OLE технологію, то для його редагування необхідно двічі клацнути мишею на цьому фрагменті. Викличеться додаток, що створив цей фрагмент. У ньому можна здійснити редагування фрагмента.

За допомогою "Буфера обмена" можна зберегти фрагмент в окремому файлі на диску. Для цього виділений фрагмент вмістити в "Буфер обмена", як і в попередньому випадку. Закрити додаток. У "Главном меню" виконати послідовно команди **"Программы"**, **"Стандартные"**, **"Просмотр буфера обмена"**. У меню **"Файл"** вікна "Буфер обмена" виконати команду **"Сохранить как"**, указавши ім'я файла і ім'я папки, в якій треба зберегти цей фрагмент.

#### 3.16.2 Впровадження об'єкта

Щоб впровадити об'єкт, наприклад, малюнок в документ текстового редактора Word, необхідно:

1. Виконати в меню **"Вставка"** вікна додатку WORD команду **"Объект"**.
2. З переліку можливих типів об'єктів, що з'явиться, вибрати потрібний, наприклад, **"Рисунок Paint"**.
3. Після цього запуситься вибраний додаток, наприклад, в нашому випадку графічний редактор Paint. Тепер можна або створити в цьому додатку малюнок, що вставляється, або завантажити якийсь малюнок з тих, що є на диску.
4. У меню **"Файл"** (у нашому разі додатку Paint) виконати команду **"Выйти и вернуться в документ"**.



5. Після цього об'єкт, що вставляється, з'явиться в документі додатку WORD.

Подвійне натиснення мишею на впровадженому об'єкті викликає додаток, в якому він створювався. Після цього об'єкт можна відредагувати, а потім повернутися в документ.

### 3.16.3 Створення зв'язку

При створенні зв'язку одного документа з файлом, що містить інший документ, в першому документі створюється посилання на файл другого документа. Наприклад, якщо за допомогою графічного редактора Paint створено файл з ім'ям Графік.bmp, що містить малюнок, то можна встановити зв'язок з цим малюнком з якого-небудь документа. Тоді будь-які зміни файла Графік.bmp переносяться в документ, в якому встановлений зв'язок з цим файлом.

Щоб зв'язати, наприклад, який-небудь малюнок з документом редактора Word, треба виконати такі дії:

- 1) У графічному редакторі Paint створити малюнок і зберегти його з якимось ім'ям.
- 2) Запустити на виконання редактор Word. Відкрити документ, в якому буде встановлюватися зв'язок з малюнком. Встановити курсор в документі в те місце, куди вставляється малюнок, що зв'язується.
- 3) У меню **"Вставка"** вікна редактора Word виконати команду **"Об'єкт"**.
- 4) У вікні, що відкрилося, вибрати вкладку **"Создание из файла"**. За допомогою пункту **"Обзор"** знайти об'єкт, що зв'язується, тобто файл з малюнком. Потім клацнути мишею по прапорцю **"Связь с файлом"**.
- 5) Підтвердити, клацнувши по кнопці .

## 3.17 Настроювання елементів системи

Систему WINDOWS можна настроїти на свій смак. Практично всі настройки (за винятком конфігурування **"Панели задач"** і **"Главного меню"**) виконуються за допомогою системної папки **"Панель управления"**. Цю папку легше усього відкрити, виконавши команду **"Настройка"** **"Главного меню"**, а потім команду **"Панель управления"**. У папці, що відкрилася, треба вибрати значок елемента, що настроюється, і двічі клацнути по ньому мишею. У вікні, що з'явиться, для елемента, що настроюється, будуть запропоновані варіанти настройки. Вибрати потрібні варіанти і підтвердити.

### 3.17.1 Настройка "Панели задач"

Клацнути правою кнопкою миші на вільному місці **"Панели задач"** і виконати команду **"Свойства"**. З'явиться діалогове вікно (рис. 3.12).

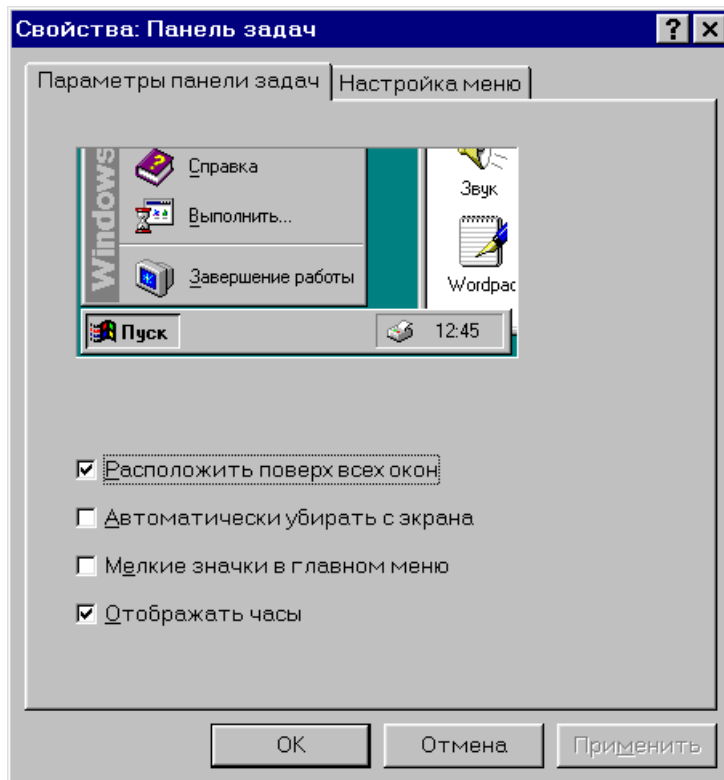


Рис. 3.12 - Діалогове вікно налаштування "Панели задач"

Щоб "Панель задач" завжди була присутня на екрані, необхідно в діалоговому вікні, що з'явилося, у вкладці "Параметры панели задач" клацнути мишею на прапорці **"Расположить поверх всех окон"**, а потім клацнути мишею по кнопці **Применить**.

Якщо в цьому вікні встановити прапорець **"Автоматически убирать с экрана"**, то "Панель задач" зникне з екрана. Вона буде з'являтися на екрані тільки в тому випадку, якщо покажчик миші буде наближатися до нижньої границі екрану.

### 3.17.2 Підключення індикатора годинника

Для підключення або відключення індикатора годинника необхідно клацнути правою кнопкою миші по вільному місцю на "Панели задач", а потім в меню об'єкта виконати команду **"Свойства"**. У вікні, що з'явиться, вибрати вкладку **"Параметры панели задач"**. У цій вкладці встановити для підключення (зняти для відключення) прапорець **"Отображать часы"**. Після цього клацнути мишею по кнопці **Применить**.

### 3.17.3 Підключення індикатора мови

У "Главном меню" виконати команду **"Настройка"**, потім команду **"Панель управления"**. У вікні панелі управління двічі клацнути мишею по значку **"Клавиатура"**. У вкладці **"Язык"** встановити прапорець **"Вывести индикатор"**.



### 3.17.4 Настроювання "Головного меню"

Клацнути правою кнопкою миші на вільному місці "Панелі задач", в меню об'єкта виконати команду **"Свойства"**. У вікні, що з'явиться, вибрати вкладку **"Настройка меню"**.

Для додавання пункту меню необхідно клацнути мишею по кнопці

У вікні **"Создание ярлыка"**, що з'явиться, в запиті **"Командная строка"**

"Командная строка"

набрати з клавіатури повне ім'я об'єкта, для якого буде створюватися пункт в "Головном меню". При заповненні командного рядка можна скористатися кнопкою . Після заповнення клацнути мишею по кнопці . З'явиться перелік папок, в яких зберігаються ярлики, відповідні пунктам "Головного меню". Указати папку, в яку треба вмістити пункт меню, що додається. Клацнути мишею по кнопці . Указати назву пункту "Головного меню" (назва ярлика), вибрати вигляд значка, а потім клацнути по кнопці .

Для видалення пункту "Головного меню" необхідно в меню об'єкта "Панелі задач" виконати команду **"Свойства"**, після появи діалогового вікна вибрати вкладку **"Настройка меню"**. У цій вкладці клацнути по кнопці . Вибрати з запропонованого списку пункт меню, що видаляється, і клацнути по кнопці .

### 3.17.5 Очищення пункту "Документи" "Головного меню"

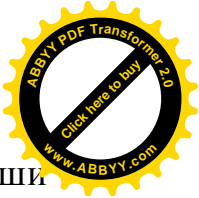
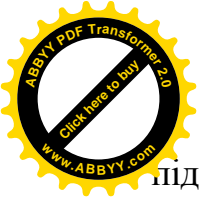
У меню об'єкта "Панелі задач" виконати команду **"Свойства"**, потім вибрати вкладку **"Настройка меню"** і клацнути по кнопці .

### 3.17.6 Підключення багатомовної підтримки

У "Головном меню" виконати команду **"Настройка"**, потім виконати команду **"Панель управления"**. Двічі клацнути мишею по значку **"Установка и удаление программ"**. У вікні, що з'явилося вибрати вкладку **"Установка WINDOWS"**. У цій вкладці в списку **"Компоненты"** встановити прапорець **"Многоязыковая поддержка"** і підтвердити, клацнувши мишею по кнопці .

### 3.17.7 Підключення мови

У "Головном меню" виконати команду **"Настройка"**, потім виконати команду **"Панель управления"**, двічі клацнути мишею по елементу **"Клавиатура"**. У вікні, що з'явиться, вибрати вкладку **"Язык"**. У діалоговому вікні клацнути по кнопці . З'явиться список мов, які можна



підключити для роботи. З нього вибрати потрібний і підтвердити, клацнувши по кнопці **Ok**.

### 3.17.8 Видалення мови з списку підключених мов

У "Главном меню" виконати команду **"Настройка"**, потім виконати команду **"Панель управления"** і двічі клацнути мишею по значку **"Клавиатура"**. У вікні, що з'явиться, вибрати вкладку **"Язык"**. Виділити мову, що видаляється, з списку підключених і клацнути мишею по кнопці **Удалить** і підтвердити, клацнувши по кнопці **Ok**.

### 3.17.9 Установка принтера

У "Главном меню" виконати команду **"Настройка"**, потім виконати команду **"Панель управления"** і двічі клацнути мишею по значку **"Принтеры"**. У вікні, що з'явиться, двічі клацнути мишею по значку **"Установка принтера"**. Вибрати з лівого списку виготовника принтера. Праворуч з'явиться перелік конкретних принтерів даного виготовника. З цього списку вибрати потрібний принтер і клацнути по кнопці **Далее**.

Указати, чи використати цей принтер за умовчанням в середовищі WINDOWS, активізувавши один з перемикачів **"Да"** чи **"Нет"** і клацнути мишею по кнопці **Далее**.

### 3.17.10 Настроювання параметрів принтера

У "Главном меню" виконати команду **"Настройка"**, потім виконати команду **"Панель управления"** і двічі клацнути мишею по значку **"Принтеры"**. Правою кнопкою миші клацнути по значку принтера, що настроюється, і виконати команду **"Свойства"**. З'явиться діалогове вікно, показане на рис. 3.13. Вибираючи вкладки **"Бумага"**, **"Графика"**, **"Параметры устройства"**, можна здійснити потрібну настройку. Розглянемо дії при виборі кожної з цих вкладок.

1) У вкладці **"Бумага"** із запропонованого переліку розмірів паперу можна вибрати потрібний (рис. 3.13). Наприклад, А4, А3 й інші. Крім того, в цьому ж вікні можна указати орієнтацію друку: **"книжная"** або **"альбомная"**.

При книжковому друці рядки виводяться по ширині листа, при альбомному - по довжині листа. Потім вибрати подачу паперу з списку, що пропонується: **"ручная"**, **"ленточный податчик"** й інші. Підтвердити свої дії, клацнувши по кнопці **Ok**.

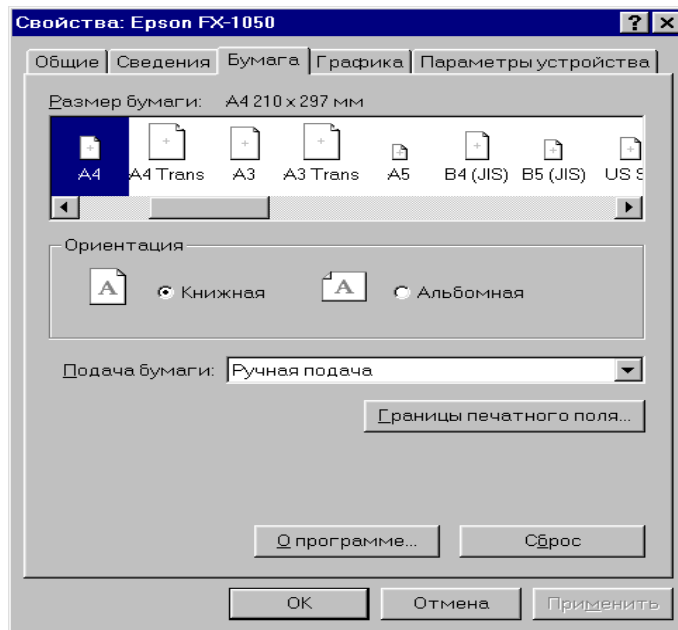


Рис. 3.13. - Принтеры - вкладка "Бумага"

2) У вкладці **"Графика"** з'явиться показане на рис. 3.14 діалогове вікно. З списку можна вибрати дозволяючу здатність друку: 240x144, 120x144 або 120x72.

Можна вказати передачу півтонів: "нет", "грубая", "точная", "контрастная", "диффузная".

У цьому ж вікні можна також вказати інтенсивність чорного кольору, встановивши в потрібне місце повзунок. Підтвердити установки, клацнувши мишею по кнопці **Ok**.

3) Вкладка **"Параметры устройства"** викликає діалогове вікно, в якому можна вибрати якість друку з списку: "высокое" або "черновое". Підтвердити установки, клацнувши мишею по кнопці **Ok**.

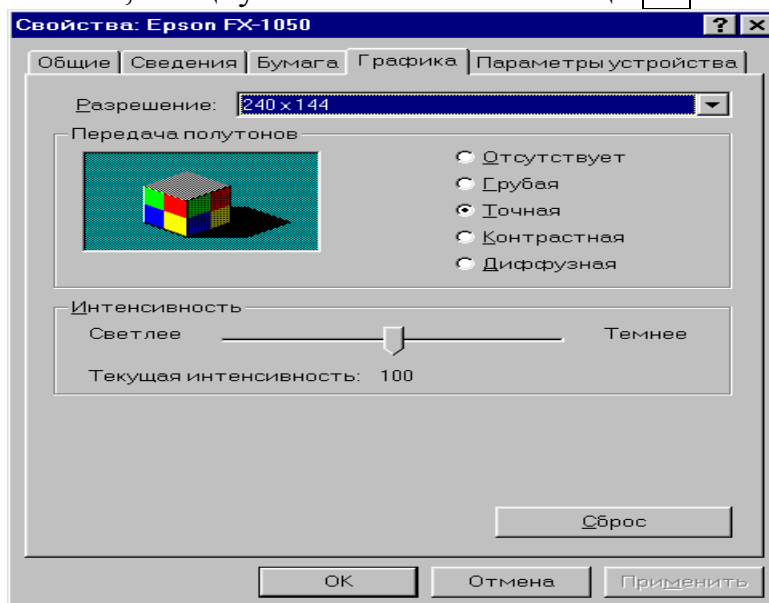


Рис. 3.14 - Принтеры - вкладка "Графика"





**!!! Зауваження.** Вигляд діалогового вікна настройки принтера залежить від типу принтера, що використовується.

### 3.17.11 Настроювання екрана

У "Главном меню" виконати команду **"Настройка"**, потім виконати команду **"Панель управления"** і двічі клацнути мишею по значку **"Экран"**. У діалоговому вікні **"Экран"**, що з'явиться, є три вкладки: **"Фон"**, **"Заставка"**, **"Оформление"**.

Це ж діалогове вікно можна викликати, виконавши команду **"Свойства"** з контекстного меню робочого стола.

При виборі вкладки **"Фон"** в діалоговому вікні з'являться переліки фонових узорів і фонових малюнків для прикраси "Рабочего стола". Вибрати з переліків потрібний фон і малюнок, а потім клацнути мишею по кнопці **OK** діалогового вікна **"Экран"**. Якщо в цьому вікні активізувати перемикач **"Размножить"**, то вибраний малюнок буде розмножений по всьому екрану, в іншому випадку він буде знаходитися тільки в центрі екрана. "Рабочий стол" можна прикрасити також своїм малюнком. Це можна здійснити двома способами:

1. Створити свій малюнок в графічному редакторі і зберегти його в файлі з розширенням bmp в системній папці WINDOWS. Тоді ім'я цього файла буде автоматично з'являтися в переліку фонових малюнків вкладки **"Фон"**.
2. Якщо файл, що містить малюнок, збережений не в системній папці WINDOWS, то за допомогою кнопки **"Обзор"** у вікні вкладки **"Фон"** вибрати потрібний файл і підтвердити, клацнувши мишею по кнопці **Ok**.

У вкладці **"Заставка"** можна вибрати із запропонованого списку вигляд заставки (охоронця екрана), яка з'являється на екрані в тому випадку, якщо протягом певного часу (встановленого в цій же вкладці) не натискалася жодна клавіша на клавіатурі і не переміщала миша.

У вкладці **"Оформление"** можна настроїти колір, розмір і шрифт будь-якого елемента вікна на свій смак.

### 3.17.12 Настроювання форми дати, часу, чисел і грошових одиниць

У "Главном меню" виконати команду **"Настройка"**, потім виконати команду **"Панель управления"** і двічі клацнути мишею по значку **"Язык и стандарты"**. У діалоговому вікні, що з'явиться, використати для настройки чисел вкладку **"Числа"**. Для настройки грошових одиниць вкладку **"Денежная единица"**, Для настройки часу вкладку **"Время"**, для настройки дати вкладку **"Дата"**.



## 4 МЕРЕЖНІ ТЕХНОЛОГІЇ

В загальному випадку мережна технологія — це узгоджений набір стандартних протоколів і програмно-апаратних засобів (адаптерів, драйверів, кабелів і т.д.), які їх реалізують.

### 4.1 Еволюція комп'ютерної технології

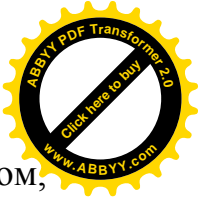
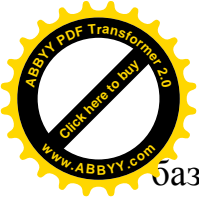
Концепція обчислювальних мереж є логічним результатом еволюції комп'ютерної технології. Перші комп'ютери 50-х років - великі, громіздкі і дорогі - призначалися для дуже невеликої кількості обраних користувачів. Такі комп'ютери не призначалися для інтерактивної роботи користувача, а використовувалися в режимі пакетної обробки.

Системи пакетної обробки, як правило, будувалися на базі мейнфрейма - потужного і надійного комп'ютера універсального призначення. Користувачі готували перфокарти, які містили в собі дані і команди програм, а потім передавали їх в обчислювальний центр. Оператори вводили ці карти в комп'ютер, а роздруковані результати користувач міг отримати тільки наступного дня.

Пакетна обробка давала можливість ефективно використовувати обчислювальну потужність комп'ютера, але користувачеві це було досить незручно, тому що він не міг сам особисто керувати процесом обробки своїх даних. Таку можливість надає тільки інтерактивний режим роботи на ЕОМ (тобто режим діалогу).

В 60-ті роки почали розвиватися інтерактивні багатотермінальні системи розподілу часу. В таких системах комп'ютер надавався в розпорядження одночасно декільком користувачам. Кожний користувач за допомогою окремого терміналу мав можливість вести діалог з комп'ютером. Термінали вийшли за межі обчислювального центру, і хоча обчислювальна потужність залишалася повністю централізованою, деякі функції — такі як введення та виведення даних — стали розподіленими. Такі багатотермінальні централізовані системи зовні були дуже схожі з локальними обчислювальними мережами. Дійсно, у користувача створювалася повна ілюзія одноособового володіння комп'ютером, але все ж таки зберігався централізований характер обробки даних.

Але поява багатотермінальних систем не вирішувала питання об'єднання комп'ютерів, які знаходилися на великій відстані. А така потреба ставала все більш актуальною. Почалося все з вирішення більш простої задачі — доступу до комп'ютера з терміналів, віддалених від нього на багато сотень, а то і тисяч кілометрів. Термінали з'єднувалися з комп'ютерами через телефонні мережі зв'язку за допомогою модемів. Потім з'явилися системи, в яких поряд з віддаленими з'єднаннями типу "термінал-комп'ютер" були реалізовані і віддалені зв'язки типу "комп'ютер-комп'ютер". Комп'ютери отримали можливість обмінюватися даними в автоматичному режимі, що, власне, і є



Базовим механізмом будь-якої обчислювальної мережі. Таким чином, хронологічно першими з'явилися **глобальні обчислювальні мережі**.

На початку 70-х років з'явилися великі інтегральні схеми. Їх порівняно невелика вартість і високі функціональні можливості привели до створення міні-комп'ютерів, які стали реальними конкурентами мейнфреймів. Міні-комп'ютери виконували задачі керування технологічним обладнанням, складами та інші задачі на рівні підрозділів підприємства. Таким чином, виникла концепція розподілу комп'ютерних ресурсів по всьому підприємству. Але всі комп'ютери однієї організації, як і раніше, продовжували працювати автономно.

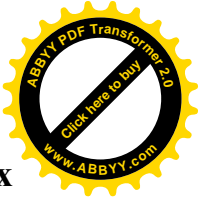
Йшов час, потреби користувачів обчислювальної техніки зростали, їм стало недостатньо власних комп'ютерів, вони вже хотіли отримати можливість обміну даними з іншими поблизу розташованими комп'ютерами. У відповідь на цю потребу підприємства і організації почали об'єднувати свої міні-комп'ютери разом і розробляти програмне забезпечення, необхідне для їх взаємодії. В результаті з'явилися перші **локальні обчислювальні мережі (ЛОМ)**.

В середині 80-х років з'явилися стандартні технології об'єднання комп'ютерів в мережу— Ethernet, Arcnet, Token Ring. Потужним поштовхом для їх розвитку стали персональні комп'ютери. Швидкість передачі даних по мережі становила до 10 МБіт/с. Розробники глобальних мереж мали змогу тільки мріяти про такі швидкості — їм доводилося користуватися тими каналами зв'язку, які були в наявності, а саме телефонними каналами зв'язку, які погано пристосовані для високошвидкісної передачі даних. Швидкість в 1200 біт/с була для них хорошим показником.

Сьогодні обчислювальні мережі продовжують розвиватися, і досить швидко. Розрив між глобальними і локальними мережами постійно скорочується в основному за рахунок появи високошвидкісних територіальних каналів зв'язку, які не поступаються якістю кабельним системам локальних мереж. В глобальних мережах з'являються служби доступу до ресурсів, такі ж зручні і прозорі, як і служби локальних мереж. Подібні приклади у великій кількості демонструє **найпопулярніша глобальна мережа - Internet**.

Змінюються також і локальні мережі. Завдяки сучасному комунікаційному обладнанню з'явилась можливість створення великих **корпоративних мереж**, які нараховують тисячі комп'ютерів і мають складну структуру. Відродилась зацікавленість великими комп'ютерами, які почали повертатися в корпоративні обчислювальні системи, але в якості повноправних потужних мережевих вузлів.

З'явилась ще одна дуже важлива тенденція, яка стосується як глобальних, так і локальних мереж. В мережах почалась обробка голосу, відеозображення, малюнків. Складність передачі такої інформації по мережі пов'язана з її чутливістю до затримок передачі, а затримки призводять до перекручення інформації. Сьогодні такі проблеми вирішуються різними способами, в тому числі і за допомогою спеціальної технології АТМ.



## 4.2 Особливості локальних, глобальних, міських та корпоративних мереж

До *локальних мереж* — Local Area Networks (LAN) — відносяться мережі комп'ютерів, які розташовані на невеликій території (в радіусі не більше 1—2 км). В загальному випадку локальна мережа являє собою комунікаційну систему, що належить одній організації. Завдяки невеликим відстаням в локальних мережах є можливість використання відносно дорогих, але високоякісних ліній зв'язку, які дозволяють досягати високих швидкостей обміну даними (100 Мбіт/с).

*Глобальні мережі* — Wide Area Networks (WAN) — об'єднують територіально рознесені комп'ютери, які можуть знаходитися в різних містах і навіть країнах. У зв'язку з тим, що проведення високоякісних ліній зв'язку на великі відстані коштує дуже дорого, в глобальних мережах часто використовуються вже існуючі і первинно призначені зовсім для інших цілей лінії зв'язку. Наприклад, багато які глобальні мережі будуються на основі телефонних каналів загального призначення.

*Міські мережі* — Metropolitan Area Networks (MAN) — є менш розповсюдженим типом мереж. Вони з'явилися зовсім недавно і призначені для обслуговування території великого міста — мегаполіса. Ці мережі займають проміжне положення між локальними і глобальними мережами, використовують цифрові магістральні лінії зв'язку зі швидкостями від 45 Мбіт/с і призначені для зв'язку локальних мереж в межах міста, а також об'єднання локальних мереж з глобальними.

*Корпоративні мережі* — це мережі в масштабі одного підприємства. Вони об'єднують велику кількість комп'ютерів на всіх територіях окремого підприємства. Вони можуть покривати місто, регіон і навіть континент. Число користувачів і комп'ютерів може вимірюватися тисячами, а відстані між мережами окремих територій можуть бути такими, що необхідним буде використання глобальних зв'язків. Тобто для корпоративної мережі характерними є:

- Масштабність - тисячі комп'ютерів, сотні серверів, великі об'єми даних, що зберігаються і передаються по лініях зв'язку.
- Висока ступінь гетерогенності - різні типи комп'ютерів, комунікаційного обладнання, операційних систем.
- Застосування глобальних зв'язків - філіали об'єднуються за допомогою телекомунікаційних засобів, в тому числі телефонних каналів, радіоканалів, супутникового зв'язку.

## 4.3 Вимоги до сучасних комп'ютерних мереж

Головною вимогою є виконання мережею її основної функції є *забезпечення користувачам потенційної можливості доступу до розподілених ресурсів всіх комп'ютерів, об'єднаних в мережу.*

Всі інші вимоги:



- *продуктивність*;
- *надійність*;
- *сумісність*;
- *розширюваність і масштабованість* - пов'язані з якістю виконання цієї головної задачі.

Існує декілька характеристик *продуктивності* мережі:

- час реакції;
- пропускна спроможність;
- затримка передачі даних.

Час реакції визначається як інтервал часу між виникненням запиту користувача до будь-якої мережної служби і отриманням відповіді на цей запит.

Пропускна спроможність відображає об'єм даних, який передається мережею за одиницю часу. Вимірюється або в бітах за секунду, або в пакетах за секунду.

Затримка передачі визначається як затримка між моментом надходження пакета даних на вхід якого-небудь мережного пристрою і моментом появи його на виході цього пристрою.

Мережу можна вважати *надійною*, якщо вона може бути використана протягом досить тривалого періоду часу. Крім того, вона має забезпечити збереження даних і захист їх від перекручувань, а також бути спроможною захистити дані від несанкціонованого доступу.

*Сумісність* означає, що мережа здатна включати в себе різноманітне програмне і апаратне забезпечення, тобто що в ній можуть співіснувати різні операційні системи і працювати апаратні засоби і додатки від різних виробників.

*Розширюваність і масштабованість* означає можливість додавання окремих елементів мережі (користувачів, комп'ютерів, додатків), а також нарощування кількості вузлів і відстанню зв'язків. Ці дії не повинні знижувати продуктивність мережі.

#### **4.4 Основні програмні та апаратні компоненти мережі**

В наш час в мережах використовують комп'ютери різних класів — від персональних комп'ютерів до мейнфреймів і супер-ЕОМ. Набір комп'ютерів в мережі має відповідати тим задачам, які розв'язує дана мережа.

*Комунікаційне обладнання* відіграє в роботі мережі не менш важливу роль, ніж комп'ютери. Це:

- кабельні системи;
- репітери;
- мости;
- комутатори;
- маршрутизатори;
- модульні концентратори.

*Мережні операційні системи* (ОС) керують всіма ресурсами мережі.





До *мережних додатків* відносять мережні бази даних, поштові системи, засоби архівації даних, системи автоматизації колективної роботи та ін.

#### 4.5 Лінії зв'язку

Будь-яка комп'ютерна мережа повинна забезпечити надійну і швидку передачу дискретних даних по лініях зв'язку. Лінія зв'язку в загальному випадку складається з:

- фізичного середовища, по якому передаються електричні інформаційні сигнали;
- апаратури для передачі даних (АПД);
- проміжної апаратури.

**Фізичне середовище передачі даних** може являти собою кабель, а також земну атмосферу або космічний простір, через які розповсюджуються електромагнітні хвилі. В залежності від середовища передачі даних лінії зв'язку розділяються на наступні:

- повітряні;
- кабельні;
- радіоканали наземного та супутникового зв'язку.

*Повітряні лінії зв'язку* — це неізольовані кабелі, що прокладені між стовпами і висять в повітрі. Традиційно по таких проводах передаються телефонні або телеграфні сигнали, але за відсутності інших можливостей ці лінії використовуються також і для передачі комп'ютерних даних. В наш час провідні (повітряні) лінії зв'язку швидко витісняються кабельними.

*Кабельні лінії зв'язку* комп'ютерних мереж створюються на основі трьох видів кабелю:

- кабелі на основі кручених пар мідних проводів;
- коаксіальні кабелі з мідною жилою;
- оптоволоконні кабелі.

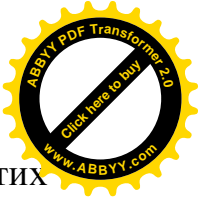
Кручена пара проводів має назву *витої пари*. Вита пара існує в двох варіантах: *екранованому*, коли пара мідних проводів обгортається в ізоляційний екран, і *неекранованому*, коли ізоляційна обгортка відсутня. Скручування проводів знижує вплив зовнішніх перешкод на корисні сигнали, що передаються по кабелю.

Коаксіальний кабель складається з внутрішньої мідної жили і обплетення, що відокремлюється від жили шаром ізоляції.

Оптоволоконний кабель складається з тонких волокон (5—60 мікрон), по яких передаються світлові сигнали. Це найбільш якісний тип кабелю — він забезпечує передачу даних з високою швидкістю (до 10 ГБіт/с і вище), до того ж краще за всі типи середовища забезпечує захист даних від зовнішніх перешкод.

*Радіоканали наземного і супутникового зв'язку* створюються за допомогою передавача і приймача радіохвиль. Існує велика кількість різних типів радіоканалів, які відрізняються частотним діапазоном і дальністю.





Супутникові канали і радіозв'язок використовується найчастіше в тих випадках, коли кабельний зв'язок використати не можна — наприклад, коли канал проходить через малонаселену місцевість, або для зв'язку з мобільним користувачем. До апаратури передачі даних відносяться, наприклад, **модеми**.

*Модем* — це пристрій, який виконує модуляцію і демодуляцію дискретних сигналів. При модуляції дискретна інформація представляється синусоїдальним сигналом заданої частоти. Демодуляція — процес зворотного перетворення.

#### 4.6 Загальні принципи побудови комп'ютерної мережі

Комп'ютери, що об'єднані в мережу, часто називають **робочими станціями**, або **вузлами мережі**. При об'єднанні комп'ютерів в мережу перш за все необхідно визначити спосіб організації фізичних зв'язків, тобто **топологію**.

*Повнозв'язкова топологія* відповідає мережі, в якій кожний комп'ютер пов'язаний зі всіма іншими. Цей варіант є громіздким і неефективним, використовується рідко.

Топологія "*загальна шина*" була до недавнього часу дуже поширеною топологією для локальних мереж. Основні переваги цієї схеми — це дешевизна і простота. А недоліками є низька надійність та невисока продуктивність, тому що при такому підключенні в кожний момент часу тільки один комп'ютер може передавати дані в мережу.

Топологія "*зірка*" передбачає підключення кожного комп'ютера окремим кабелем до загального пристрою, який називається **концентратором** і знаходиться в центрі мережі. В функції концентратора входить направлення інформації, що передається комп'ютером, одному або всім іншим комп'ютерам мережі.

Головна перевага цієї топології — більш висока надійність, тому що будь-які неприємності відносно кабелю стосуються лише того комп'ютера, до якого цей кабель приєднаний, і тільки несправність концентратора може вивести з ладу всю мережу. Недоліками є більш висока вартість, а також те, що можливості по нарощуванню кількості вузлів в мережі обмежуються кількістю портів концентратора.

В комп'ютерних мережах використовуються як індивідуальні лінії зв'язку, так і розподілені, коли одна лінія зв'язку використовується по черзі декількома комп'ютерами.

#### 4.7 Стандарти створення комп'ютерних мереж

Сучасні комп'ютерні мережі будуються згідно з двома важливими стандартами:

- стандартом Моделі Відкритої Системи Зв'язку (OSI);
- стандартом мережної технології.



### 4.7.1 Модель OSI

Міжнародна Організація по Стандартизації заснувала *Модель Відкритої Системи Зв'язку (OSI)*, що є основою при проектуванні мережі. Більшість мереж проектується відповідно до цієї моделі.

**Фізичний рівень** забезпечує фізичний зв'язок між комп'ютером і мережевим середовищем передачі даних.

**Канальний рівень** забезпечує надійність та достовірність передачі даних, а також контроль і корекцію помилок.

**Мережевий рівень** використовує послуги зв'язку нижчого рівня (канального) для організації передачі даних в мережі.

**Транспортний рівень** визначає мережеву адресацію (фізичне розміщення пристроїв в мережі) і способи з'єднання та від'єднання різних частин мережі.

**Сеансовий рівень** забезпечує взаємодію прикладних програм в мережі.

**Рівень представлення** визначає формат даних при обміні між робочими станціями. Програмне забезпечення комп'ютера-відправника конвертує дані з формату рівня додатків в проміжний формат. А на комп'ютері-одержувачі відбувається зворотне перетворення даних.

**Рівень додатків** визначає мережні прикладні програми, які обслуговують файли.

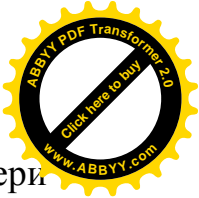
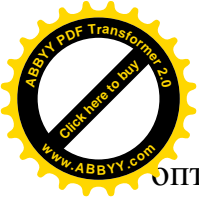
Однією з умов надійної роботи ієрархічної архітектури є узгодженість форматів передачі даних між рівнями. Рівень може нічого не знати про зміст повідомлення, але він повинен знати, що робити далі з цим повідомленням. Наприклад, якщо ви хочете відправити електронну пошту, ви пишете повідомлення і відправляєте його з прикладної програми (тобто з рівня додатків).

Рівень додатків передає повідомлення на наступний рівень і так далі через всі рівні, поки фізичний рівень не передасть його в кабель. Кожний рівень по-своєму обробляє повідомлення електронної пошти, але не знає про фактичний його зміст.

Для розв'язання цієї задачі кожний рівень просто додає до блока даних, отриманого від попереднього рівня, свою службову інформацію у вигляді додаткових блоків в початок і кінець повідомлення, щоб відповідний рівень міг правильно обробити повідомлення при прийомі. Під час цього процесу зміст повідомлення не аналізується. Вся ця процедура називається *інкапсуляцією*, тому що кожний рівень додає свою порцію інформації навколо початкового пакета даних.

### 4.7.2 Технологія Ethernet

Стандарт Ethernet був прийнятий у 1980 році. Головний принцип, який покладений в основу Ethernet, — це випадковий метод доступу до розподіленого середовища передачі даних. В якості такого середовища може використовуватися товстий або тонкий коаксіальний кабель, вита пара,



оптоволоконні кабелі або радіохвилі. В стандарті Ethernet комп'ютери підключаються відповідно до топології “загальна шина”. Управління доступом до лінії зв'язку здійснюється мережевими адаптерами Ethernet. Передача даних відбувається зі швидкістю 10 МБіт/с.

*Суть випадкового методу доступу:* комп'ютер в мережі Ethernet може передавати дані тільки якщо мережа вільна, тобто якщо жодний інший комп'ютер в даний момент часу не займається обміном інформації. Після того як комп'ютер “впевнився”, що мережа вільна, він “захоплює” середовище, починаючи передачу.

Одиниця даних, якими обмінюються комп'ютери в мережі Ethernet, називається **кадром**.

Мережа Ethernet побудована таким чином, що коли кадр потрапляє в розподілене середовище передачі даних, всі мережні адаптери одночасно починають приймати цей кадр. Всі вони аналізують адресу призначення і якщо ця адреса співпадає з їх власною адресою, кадр розміщується у внутрішньому буфері мережевого адаптера. Таким чином, комп'ютер-адресат отримує призначені для нього дані.

Іноді може виникати ситуація, коли одночасно два або більше комп'ютерів вирішують, що мережа вільна, і починають передавати інформацію. Така ситуація називається **колізією** і заважає правильній передачі даних. Після виявлення колізії мережні адаптери, які намагалися передавати свої кадри, припиняють передачу і після паузи випадкової тривалості намагаються знову отримати доступ до середовища і передати той кадр, що викликав колізію.

В мережах з досить великою кількістю комп'ютерів виникають обмеження на довжину зв'язку між вузлами, а також обмеження на кількість вузлів в мережі. Наприклад, технологія Ethernet на тонкому коаксіальному кабелі дозволяє використовувати кабель довжиною не більше 185 метрів, до якого можна підключити не більше ніж 30 комп'ютерів. Крім того, якщо комп'ютери обмінюються між собою інформацією з досить високою інтенсивністю, доводиться знижувати кількість комп'ютерів до 20—10, щоб кожному комп'ютеру діставалася допустима частка пропускної спроможності мережі. Для зняття цих обмежень використовується спеціальне обладнання, яке називають **комунікаційним**.

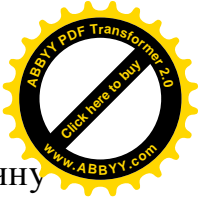
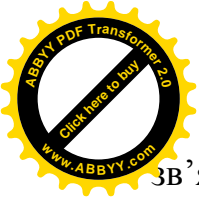
#### **4.7.2.1 Комунікаційне обладнання Ethernet**

До такого обладнання відносяться репітери, концентратори, комутатори, маршрутизатори.

**Репітер** використовується для фізичного з'єднання окремих сегментів кабелю локальної мережі з метою збільшення загальної довжини мережі. Він передає сигнали, які приходять з одного сегмента мережі в інші її сегменти .

Репітер, який має декілька портів і з'єднує декілька фізичних сегментів, часто називають **концентратором** або **хабом**.

Концентратор Ethernet повторює вхідні сигнали на всіх своїх портах, крім того, з якого ці сигнали надходять. Концентратор завжди змінює конфігурацію



зв'язків, що утворюються окремими частинами кабелю (тобто фізичну конфігурацію мережі), але залишає без зміни конфігурацію інформаційних потоків (тобто логічну конфігурацію) між комп'ютерами мережі. Таким чином, концентратори виконують *фізичну структурування мережі*.

Фізична структурування дуже корисна в багатьох відношеннях, але найважливішою проблемою, яку фізична структурування не вирішує, є проблема перерозподілу трафіку між різними фізичними сегментами мережі.

Справа в тому, що в великій мережі виникає неоднорідність інформаційних потоків. До складу мережі можуть входити декілька підмереж робочих груп, відділів, філіалів підприємства та інших адміністративних утворювань.

До недавнього часу найбільш інтенсивний обмін інформацією відбувався між комп'ютерами однієї підмережі і тільки невелика частина звернень припадала на ресурси, що знаходилися поза локальними робочими групами. (Навіть був сформульований емпіричний закон "80/20", відповідно до якого в кожній підмережі 80% трафіку є внутрішнім і тільки 20% — зовнішнім.) Але зараз, завдяки створенню на багатьох підприємствах корпоративних сховищ даних, які активно використовуються всіма співробітниками підприємства, характер навантаження мереж змінюється. І зараз все частішими стають ситуації, коли інтенсивність зовнішніх звернень вища за інтенсивність обміну між "сусідніми" машинами. Отже, для підвищення ефективності роботи мережі необхідно враховувати неоднорідність інформаційних потоків.

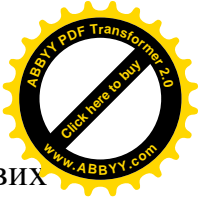
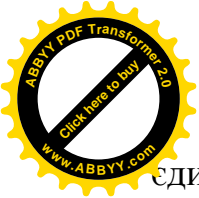
Наприклад, на підприємстві декілька відділів. Комп'ютери одного відділу спілкуються між собою, а також з комп'ютерами інших відділів. Необхідність у зв'язку між комп'ютерами двох різних відділів виникає набагато рідше, ніж між комп'ютерами одного відділу. Але в мережі з загальною шиною комп'ютери одного відділу вимушені чекати, поки закінчиться зв'язок між парою комп'ютерів іншого відділу.

Вирішення проблеми полягає в тому, щоб відмовитися від ідеї однорідного розподіленого середовища. Для логічної структурування мережі використовується таке комунікаційне обладнання, як *мости, комутатори, маршрутизатори і шлюзи*.

**Міст** ділить розподілене середовище передачі на частини (які часто називаються логічними сегментами) і передає інформацію з одного сегмента в інший тільки в тому випадку, якщо така передача дійсно необхідна, тобто якщо адреса комп'ютера призначення належить іншій підмережі.

**Комутатор** за принципом обробки кадрів нічим не відрізняється від моста. А головна його відмінність від моста в тому, що він є своєрідним комунікаційним мультипроцесором, оскільки кожний його порт обладнаний спеціалізованим процесором, який обробляє кадри за алгоритмом моста незалежно від процесорів інших портів. Можна сказати, що комутатори — це мости нового покоління, які обробляють кадри в паралельному режимі.

**Маршрутизатори** надійніше і ефективніше, ніж мости, ізолюють трафік окремих частин мережі один від одного. Маршрутизатори можуть також виконати вибір найбільш раціонального маршруту з декількох можливих. Ще однією дуже важливою функцією маршрутизаторів є їх здатність зв'язувати в



єдину мережу підмережі, побудовані із застосуванням різних мережевих технологій, наприклад Ethernet і X.25.

Крім перелічених пристроїв окремі частини мережі може з'єднувати *шлюз*. Основною причиною, по якій в мережі використовують шлюз, є необхідність об'єднати мережі з різними типами системного і прикладного програмного забезпечення.

### 4.7.3 Подальший розвиток технології Ethernet

В наш час найпоширенішою мережною технологією є технологія Ethernet. В 1997 році більш ніж 80% всіх мереж було побудовано на базі Ethernet. Всі популярні операційні системи і стеки протоколів підтримують Ethernet. Класичний 10-мегабітний Ethernet влаштував більшість користувачів протягом 15-ти років.

Але потреба у високошвидкісній і в той же час недорогій технології для підключення до мережі потужних робочих станцій призвели на початку 90-х років до появи нової технології — Fast Ethernet. Всі відмінності технології Fast Ethernet від Ethernet зосереджені на фізичному рівні.

В цій технології використовуються нові варіанти кабельних систем із застосуванням крученої пари і оптоволокна. Технологія Fast Ethernet дає можливість підвищити продуктивність мережі до 100 МБіт/с. Але реальна пропускна спроможність лінії Fast Ethernet виявляється меншою за розрахункову.

В 1999 році був прийнятий новий стандарт мережної технології — Gigabit Ethernet. Основна ідея цього стандарту перебуває в максимальному збереженні ідей класичної технології Ethernet при швидкості до 1000 МБіт/с. Цей ступінь розвитку мережевої технології дозволяє ефективно будувати крупні локальні мережі, в яких потужні сервери і магістралі нижніх рівнів мережі працюють на швидкості 100 МБіт/с, а магістраль Gigabit Ethernet об'єднує їх, забезпечуючи досить великий запас пропускнуої спроможності.

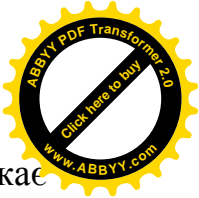
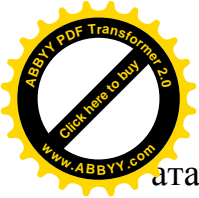
Останнім часом широко застосовується також технологія АТМ, яка дозволяє об'єднувати цифровий і аналоговий потоки інформації, тобто вирішує проблему передачі голосової і відеоінформації. Головним недоліком такої технології є складність, а як наслідок — дороге обладнання і додаткове навчання персоналу.

## 4.8 Загальні відомості про глобальну комп'ютерну мережу Internet

Internet — це глобальна комп'ютерна мережа, яка оповила всю земну кулю і пов'язує між собою урядові, військові, освітянські і комерційні структури, а також окремих громадян. Вона надає широкий вибір комп'ютерних послуг, ресурсів та інформації.

Комплекс мережних угод і загальнодоступних інструментів розроблений з метою створення однієї великої мережі, в якій комп'ютери взаємодіють на різних програмних та апаратних платформах. Internet розпочиналась як мережа під назвою ARPANET в часи холодної війни (1969 рік) і розроблялася військовими структурами як мережа, яка була б здатною пережити ядерну





атаку. А сьогодні Internet — одна з найпопулярніших технологій, яка проникає майже у всі сфери людської діяльності.

Кількість користувачів Internet бурхливо зростає з кожним роком і, згідно з дослідженнями компанії Computer Industry Almanac, складає в 2000 році вже 374 млн. чоловік.

Отже, основним “продуктом”, який можна знайти в Internet, є інформація. Ця інформація зберігається на так званих *хост-комп'ютерах* у вигляді файлів. Ключем для отримання інформації в Internet є *адреси ресурсів*. До них відносяться *поштові адреси* (mail addresses), що використовуються для пересилання повідомлень по електронній пошті, а також *адреси хост-комп'ютерів* для з'єднання з ними і отримання файлів.

Всі хост-комп'ютери Internet ідентифікуються відповідно до унікального *доменного імені* (domain name), наприклад такого, як **www.bigcorp.com**.

Доменне ім'я складається з декількох частин, які визначають конкретний хост-комп'ютер, організацію, якій він належить, а також ієрархію доменів (тобто об'єднань комп'ютерів і мереж).

*Наприклад:* **queen.ukma.kiev.ua** або **bigmashine.bigcorp.com**

Розглянемо це ім'я справа наліво. Комбінація літер **com** означає, що ця адреса зареєстрована в домені, який відведено для комерційних організацій різного профілю дії. Це — домен верхнього рівня. Взагалі, ім'я домену **edu** використовується для просвітянських організацій, **com** — для комерційних фірм, **gov** — для урядових закладів, **mil** — для організацій, які належать військовим відомствам. Існують також домени верхнього рівня, які об'єднують комп'ютери за географічним принципом, наприклад: **fr** — для Франції, **ru** — для Росії, **de** — для Німеччини, **ua** — для України.

Комбінація літер **bigcorp** визначає організацію, в мережі якої знаходиться даний комп'ютер. Коли яка-небудь організація отримує доступ до Internet, вона реєструє свою назву в спеціальних реєстраційних службах Internet.

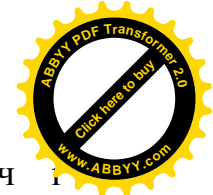
Комбінація літер **bigmashine** визначає ім'я конкретного комп'ютера в мережі даної організації.

Доменному імені хост-комп'ютера відповідає числова *IP-адреса*, яка необхідна для того, щоб програмне забезпечення, що підтримує роботу мережі Internet, було здатним передати або прийняти інформацію з хоста. IP-адреса є унікальним номером і являє собою набір з чотирьох чисел, кожне з яких лежить в діапазоні від 1 до 254 і які розділені між собою крапками, наприклад: 192.58.107.230. Програмне забезпечення автоматично транслює доменні імена хост-комп'ютерів в IP-адреси.

#### 4.8.1 Основні послуги, які надає користувачам Internet

Основою мережі Internet є набір протоколів, який має назву TCP/IP (Transmission Control Protocol/Internet Protocol, протокол управління передачею/протокол мережі Internet). Він реалізує міжмережевий обмін. Протокол TCP/IP використовується для передачі даних як в глобальній мережі Internet, так і в локальних мережах. Як вже було сказано, протокол TCP/IP — це не один протокол, а ціла група протоколів.





Різні протоколи відповідають за свій тип виконуваних задач і забезпечують свій *сервіс*. Деякі з них безпосередньо використовуються в прикладних програмах. Так, протокол FTP (File Transfer Protocol) забезпечує такий сервіс, як передача файлів між комп'ютерами, а протокол Telnet забезпечує віддалений термінальний доступ до системи.

Які ж основні послуги може надати користувачеві Internet?

До них відносяться:

- передача інформації (FTP, Gopher)
- пошук інформації (WAIS, Archie, Veronica)
- комунікаційний сервіс (E-mail, Telnet, UseNet, IRC)
- Мультимедіа інформаційні системи (World Wide Web).

#### **4.8.1.1 Передача файлів за протоколом FTP**

Інформаційний сервіс, заснований на передачі файлів з використанням протоколу **FTP** був одним із перших, розроблених в Internet. За його допомогою користувач може, використовуючи відповідну FTP-програму на своєму комп'ютері, підключитися до віддаленої машини (FTP-сервера), ознайомитися з переліком доступних файлів і скопіювати їх на свій комп'ютер. FTP дозволяє пересилати по мережі файли будь-якого типу — тексти, зображення, виконувані файли, файли з записом звукових фрагментів і т.д.

FTP є одним з прикладів систем з архітектурою *клієнт-сервер*. В системах такого типу користувач на своєму комп'ютері використовує спеціальну програму, яка називається *клієнтом*. За її допомогою він зв'язується з іншою програмою, яка працює на віддаленому комп'ютері і називається *сервером*.

#### **4.8.1.2 Доступ до ресурсів хост-комп'ютерів за допомогою Telnet**

Протокол **Telnet** забезпечує можливість входу у віддалену систему. Він дозволяє користувачеві одного комп'ютера зареєструватися на віддаленому комп'ютері, розташованому в іншій частині мережі.

При цьому користувачеві здається, що він працює за терміналом віддаленого комп'ютера. Telnet може бути в нагоді, коли ви, працюючи на "повільному" комп'ютері, хочете скористатися обчислювальними ресурсами більш потужної машини, а також якщо на віддаленому комп'ютері є необхідне вам програмне забезпечення.

Головна відмінність протоколу Telnet від протоколу FTP полягає в тому, що при підключенні до віддаленого комп'ютера по FTP-протоколу відповідний FTP-сервер дозволяє виконувати тільки ті операції, які пов'язані з передачею файлів. А якщо з'єднання встановлено за допомогою Telnet-програми, то користувач може працювати з програмним забезпеченням віддаленого комп'ютера.



### 4.8.1.3 Електронна пошта

Однією з головних і найпопулярніших послуг Internet є електронна пошта (*e-mail*). Електронні повідомлення легко посилати і читати, на них легко відповідати, ними легко керувати. У електронної пошти є багато переваг в порівнянні з такими способами передачі інформації, як традиційна пошта або факсимільний зв'язок.

Дослідження довели, що при проведенні опитувань респонденти охочіше відповідали на питання, які надійшли по електронній пошті, ніж на традиційні листи. Електронний лист може бути прочитаним або написаним в будь-який час, тому досить зайняті люди можуть використати будь-який вільний час для роботи із своєю кореспонденцією. Електронна пошта досить економічна. Відіслати E-mail коштує значно дешевше, ніж відправити звичайний лист або скористатися послугами міжміського або міжнародного телефонного зв'язку.

### 4.8.1.4 Телеконференції UseNet

Для багатьох користувачів поняття Internet тісно пов'язане з терміном *UseNet*. Це одна з найбільш популярних послуг Internet, до якої кожний день звертаються мільйони користувачів. Терміном UseNet називається система, яка дозволяє всім користувачам UseNet брати участь в групових дискусіях, так званих телеконференціях, на яких обговорюються різні питання.

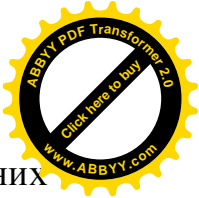
Первинно ця система була призначена для обміну технічною інформацією, але дуже скоро перетворилася на систему з набагато більш широкою областю застосування.

Зараз UseNet підтримує більш ніж 8000 телеконференцій різноманітної тематики. Як зрозуміти, що таке UseNet? Уявіть собі електронну пошту, яка відрізняється від описаної вище тим, що повідомлення відсилаються не окремим користувачам, а в спеціальним чином організовані дискусійні групи — телеконференції. Відправлені в телеконференції повідомлення стають доступними для будь-кого, хто до них звертається. В UseNet застосовується частина термінології електронної пошти. Матеріал, який пишуть для того, щоб з ним могли ознайомитися інші, називають повідомленням, або статтею. Повідомлення посилають в телеконференції і називають це публікацією в телеконференції.

Для роботи з телеконференціями використовують програми читання телеконференцій, які дозволяють вибрати телеконференцію і продивитися її зміст. Для зручного пошуку телеконференціям надаються імена. Система імен будується за ієрархічним принципом. Ім'я телеконференції складається з декількох частин різних рівнів, кожна з яких несе певну інформацію про тематику конференції.

#### Наприклад:

Телеконференція **ukr.comp.software** організована для проведення бесід на теми, пов'язані з технікою програмування. Тут недоречна реклама програмної продукції.



Телеконференція **rec.autos.driving** організована любителями старовинних автомобілів. Для того, щоб уявити собі, як ведеться дискусія в телеконференції, можна провести аналогію з будівлею, в якій є багато кімнат, а кожна кімната — це телеконференція. Кожна кімната має табличку з назвою телеконференції і коротким описом предмету обговорення. В деяких кімнатах можна знайти невеликі групи людей, які обговорюють певні проблеми, певні теми. Можна зайти, задати питання і приєднатися до дискусії. В інших кімнатах можна знайти гомінливий натовп, який обговорює “гарячі” новини. Кожний в цій групі висловлює свою думку, не звертаючи уваги на інших людей. Можна спробувати теж приєднатися до дискусії, але може вийти так, що ніхто не зверне на вас уваги. Ці два типи дискусій зустрічаються в UseNet.

#### **4.8.1.5 Конференції Internet Relay Chat**

Internet Relay Chat (IRC) як мережевий сервіс у складі Internet був розроблений в кінці 80-х років. IRC дозволяє багатьом людям безперервно “розмовляти” один з одним шляхом введення слів через клавіатуру. IRC побудований згідно з архітектурою клієнт-сервер.

Користувачі, які бажають поспілкуватися один з одним, мають запусити на своїх машинах клієнтську частину програмного забезпечення IRC і приєднатися до IRC-сервера. На сервері користувач вибирає *канал*, по якому має вестися “розмова”.

Коли користувач працює у вибраному каналі IRC, він може друкувати на клавіатурі повідомлення для інших учасників дискусії і одночасно спостерігати, що вони відповідають. Це досить цікавий спосіб спілкування в режимі реального часу, але його темп не дуже високий тому, що слово друкується набагато довше, ніж вимовляється. Але кожний учасник бере участь в дискусії на рівних правах.

#### **4.8.1.6 World Wide Web (WWW)**

World Wide Web (в перекладі - всесвітня павутина) надає можливість працювати з документами, в яких об’єднані текст, графічні ілюстрації, звукові фрагменти і навіть анімація, що робить їх надзвичайно виразними і полегшує сприйняття інформації.

Документи системи WWW містять в собі посилання на інші, пов’язані за змістом, документи, до того ж останні можуть знаходитися на віддалених комп’ютерах. Система WWW не тільки надає зручний графічний інтерфейс для доступу до гіпермедіа - документів, але й дозволяє використовувати той же інтерфейс для роботи з іншими сервісами Internet, такими як FTP, Gopher, телеконференції UseNet.



## 5 ЗАСТОСУВАННЯ ІНТЕРНЕТУ В ЕКОНОМІЦІ

Виділяють три основні технології електронної комерції: Інтернет, WAP і EDI.

**Інтернет.** Основною технологією, що використовується в електронній комерції, є технологія Інтернету. Поява Інтернету значно здешевила ведення електронної комерції за рахунок низької собівартості передачі інформації і привела до виникнення її якісно нових форм, де ключовим моментом є взаємодія із споживачем через веб-сайт.

Компанії, що ведуть бізнес через Інтернет, в основному діють таким чином. Компанія-продавець розміщує на своєму веб-сервері інтерфейс, за допомогою якого кінцевий споживач або партнер може, наприклад, сформувати і розмістити замовлення в інформаційній системі компанії-продавця. Після цього спеціальне програмне забезпечення компанії-продавця обробляє замовлення.

Таким чином, технологія Інтернет:

- дозволяє здійснювати маркетинг і продаж товарів і послуг;
- зрівнює можливості компаній;
- сприяє організації безпаперового документообігу;
- забезпечує якісніше обслуговування споживачів. Так, на більшості веб-сайтов крупних компаній формується зворотний зв'язок із споживачем у вигляді книги пропозицій і коментарів або форумів; створюється список питань, що часто ставляться. Відповіді на ці питання розміщуються у вигляді простого списку на веб-сайті комерційної організації для того, щоб користувачі могли знайти рішення своїх проблем у будь-який час а при появі нового питання відповідь на нього вводиться в даний список. Використовування електронної пошти також позбавляє персонал від необхідності бути постійно прив'язаним до телефону;
- забезпечує виконання управлінських функцій і розповсюдження ідей.

До основних недоліків технології Інтернету відноситься загальна уразливість комп'ютерної мережі.

**WAP.** Окремим випадком використання технології Інтернету є технологія WAP. Це стандартний протокол, що забезпечує інтерактивний радіодоступ в мережу Інтернет з мобільного телефону. Він є спрощеним і економічнішим в порівнянні із звичним протоколом IP. WAP можна розглядати як мобільний Інтернет, а комерційну діяльність, що проводиться з використанням цього протоколу, мобільною комерцією.

**EDI.** Технологія EDI (електронний обмін даними) - найстаріша технологія, що використовується в електронній комерції. Дотепер велика частина трафіку між компаніями здійснюється за допомогою електронного обміну даних. EDI дозволяє одній обчислювальній системі зв'язуватися з іншою, використовуючи для цього стандартний формат електронного обміну діловою інформацією. Як ділова інформація можуть служити підтвердження

замовлень, дані про стан операцій або графіки поставок і т.д. Такі дані представляють 75% загального трафіку між комерційними підприємствами.

Основною гідністю технології EDI є стандартизація всіх процедур документообігу між компаніями. Встановивши у себе EDI-шлюз, компанія дістає можливість вести електронний бізнес з будь-якою іншою компанією, що має такий же шлюз. На сьогоднішній день EDI-шлюзи інтегровані у всі найвідоміші системи управління підприємством, а також у всі відомі пакети додатків для електронної комерції.

Але електронний обмін даними також не позбавлений недоліків, основними з них можна назвати:

- необхідність доробки програмного забезпечення інформаційних систем компаній для перетворення документів в сумісний формат;
- необхідність узгодження способу формування EDI-документів, оскільки стандарти допускають деяке свавілля в цьому відношенні. Для такої роботи потрібен високооплачуваний кваліфікований персонал;
- для того, щоб упровадження цієї технології окупилося, число транзакцій повинне бути дуже великим, а це характерно лише для крупних компаній.

### 5.1 Сектори електронної комерції

Залежно від типу учасників на ринку електронної комерції прийнято виділяти наступні сектори (рис. 5.1):



Рис. 5.1 - Сектори електронної комерції





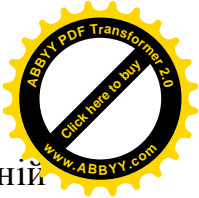
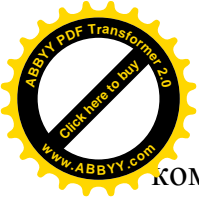
- *B2B (Business to Business)* – сектор включає всі рівні взаємодії між компаніями.
- *B2C (Business to consumer)* - сектор орієнтований на кінцевого споживача. Основу цього напрямку складає електронна роздрібна торгівля;
- *C2B (Consumer to Business)* – до цього сектора відносяться системи обробки цінових заявок, по яких кінцеві споживачі хотіли б придбати товар і послуги;
- *B2A (Business to administration)* – сектор орієнтований на операції між компаніями і урядовими організаціями (адміністраціями, починаючи від місцевих властей і закінчуючи міжнародними організаціями).
- *C2A (Consumer to administration)* – сектор орієнтований на організацію взаємостосунків між фізичними особами (кінцевими споживачами) і державними службами (адміністрацією).
- *C2C (Consumer to Consumer)* – сектор орієнтований на взаємодію між великою кількістю кінцевих споживачів. Цей напрям включає можливість взаємодії споживачів для обміну комерційною інформацією, обміну досвідом придбання того або іншого товару, обміну досвідом взаємодії з якою-небудь компанією і т.д. До цієї ж області відноситься і форма аукціонної торгівлі між фізичними особами. На сьогоднішній день найрозвиненішими є сектори *B2C* і *B2B*. Інші сектори електронній комерції знаходяться на початковій стадії розвитку. Розглянемо докладніше деякі з них.

**Електронна комерція B2C.** Це бізнес між комерційним підприємством і споживачем. Перші спроби залучення покупців і використання електронних комунікацій для вибору товару замість відвідин магазинів почалися з невеликих веб-сайтів, метою яких було привернути потенційних покупців в існуючий традиційний магазин. На такому веб-сайті звичайно указувалися контактні телефони, адреси і схема проїзду до магазину.

Пізніше на веб-сайтах почали з'являтися прайс-листи на товари і послуги компаній, що періодично обновляються. Так з'явилися каталоги онлайн, які зараз є достатньо барвистою і інформативною заміною вітринам традиційних магазинів. Далі послідував етап, що забезпечив повноцінне існування електронних магазинів як повністю самостійних організацій. Тепер клієнт дістав можливість не тільки проглядати каталоги, але і замовляти і оплачувати товари і послуги не відходячи від комп'ютера. Таким чином, відбувся розрив між реальними магазинами і їх інтернет-версіями, тобто електронний магазин зміг існувати без свого фізичного аналога.

У електронній комерції *B2C* присутні два основні учасники - компанія і кінцевий споживач (покупець). Покупець здійснює пошук товару, його замовлення і оплату через мережу Інтернет, зокрема через веб-сайт. В даному випадку взаємозв'язок між бізнесом і споживачем відбувається в наступних областях: просування товару; замовлення товару; оплата замовлення; доставка товару; післяпродажна підтримка. Для виконання цих функцій в структурі





компанії звичайно існують відповідні відділи. Наприклад, в структуру компаній електронної комерції *B2C* можуть входити наступні складові:

- 1) інтерактивний веб-сайт. Такі веб-сайти можна класифікувати таким чином:
  - інформаційні - що містять інформацію про продукцію, товари і послуги, прайс-листи і каталоги;
  - комерційні - дозволяючі клієнту не тільки ознайомитися з наявними товарами і послугами, але і оформити замовлення;
  - спеціалізовані - орієнтовані на певну групу користувачів;
- 2) адміністрація компанії і технічний персонал;
- 3) служба доставки (може бути відсутнім, якщо компанія продає цифрові товари або надає і інформаційних послуги консультантів);
- 4) підрозділ по роботі з постачальниками (може бути відсутнім, якщо компанія надає послуги силами власного персоналу);
- 5) служба прийому платежів;
- 6) маркетингова служба і відділ реклами;
- 7) служба технічної підтримки і т.д.

У залежності об того, який сегмент ринку обслуговує компанія, які послуги і яким чином надає, бізнес-моделі в електронній комерції *B2C* можна класифікувати таким чином: чисті інтернет-компанії і змішані (тобто використовуючі як традиційні методи ведення бізнесу, так і новітні технології). Треба відзначити, що і у інтернет-компаній також є офіси, але велика частина ділових операцій відбувається в мережі Інтернет.

До інтернет-компаній можна віднести:

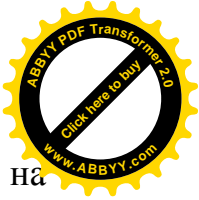
- 1) портали, що в основному надають пошукові механізми і рекламу. Вони привертають покупців, і основні доходи одержують за рахунок продажу рекламних блоків;
- 2) повнофункціональні електронні магазини;
- 3) компанії по обробці цінових заявок.

Змішані компанії максимально використовують Інтернет в своїй діяльності. Такі компанії можуть творчо використовувати можливості, що надаються електронною комерцією, для залучення покупців і зміцнення позицій на ринку.

Серед компаній змішаного типу можна виділити компанії:

- 1) використовуючі нові комунікації як альтернативу старим (наприклад, електронні магазини, що мають і фізичний аналог);
- 2) використовуючі Інтернет тільки для просування товарів (наприклад, електронні вітрини);
- 3) здійснюючі в Інтернеті тільки підтримку покупців.

**Електронна комерція *B2B*.** Це бізнес між комерційними підприємствами (тобто між покупцями і продавцями в особі комерційних організацій). За деякими оцінками, ринок комерції *B2B* в 10 разів крупніший за ринок комерції *B2C*. Наприклад, в США в 2002 р. питома вага сектора *B2B* в загальному об'ємі електронної комерції склала 93%, тоді як питома вага сектора *B2C* - 7%.



У електронній комерції *B2B* об'єднуються компанії, що працюють на міжкорпоративному ринку, де одні юридичні особи надають послуги і продають товари іншим юридичним особам. До цієї категорії відносяться оптові магазини, інтернет-біржі, торгові майданчики, компанії - виробники і продавці сировини, устаткування, товарів і послуг, компанії, що займаються веб-хостингом, і т.д.

Електронна комерція *B2B* є ринком глобального масштабу. Нові правила міжнародної торгівлі, прийняті Всесвітньою торговою організацією (ВТО), відкрили місцеву економіку у всьому світі для міжнародних постачальників. Поточна ситуація в бізнесі характеризується як гіперконкуренція - збільшується число постачальників продукції, збільшується вибір для споживачів, які стають все більш вимогливими. Вимогливість клієнтів вимушує постачальників знижувати ціну і надавати поліпшений сервіс. Підприємства вимушені скорочувати експлуатаційні і навантажувально-розвантажувальні витрати і в той же час прискорювати дію ланцюжка поставки.

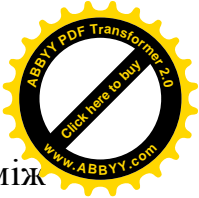
Мета електронної комерції *B2B* полягає в економії засобів на закупівлях, які здійснюються практично миттєво. При цьому постачальники використовують веб-сайти закупляючих організацій для внесення пропозицій про ціни на товари і продажі надлишків товарних запасів. Заміна бюрократичної тяганини, пов'язаної із закупівлями, на оперативні посилання означає економію засобів, підвищення ефективності замовлень, скорочення помилок і оперативну доставку, що зводить до мінімуму складські витрати.

Нові комунікації створюють можливості для реорганізації ланцюжка поставки. Наприклад, покупець може напряму контактувати з виробником. Таким чином, Інтернет надає таку платформу для бізнесу, в якій кожен учасник ланцюжка може співробітничати з іншим учасником напряму, минувши посередників.

Відмінності між *B2B* і *B2C* більш істотні, ніж між оптовою і роздрібною торгівлею. З погляду бізнесу це означає економію усередині організації, готовність вступати у вигідні союзи з постачальниками, скорочення витрат при оперативній доставці товарів і послуг і т.д.

Структура *B2B-компаній* багато в чому схожа із структурою компаній *B2C*. Її компонентами можуть бути:

- 1) інтерактивний веб-сайт, що містить інформацію про продукцію, товари і послуги, прайс-листи і каталоги, інформацію про оптові знижки і т.д.;
- 2) адміністрація компанії і технічний персонал (виконують ті ж функції, що і в компаніях електронної комерції, плюс *B2C* документообіг з ділерами, партнерами і покупцями);
- 3) служба доставки (треба відзначити, що в компаніях *B2B* служба доставки звичайно менш актуальна, часто доставкою товару займається незалежна транспортна компанія);
- 4) підрозділ по роботі з постачальниками;
- 5) маркетингова служба і рекламний відділ і т.д.



На відміну від електронної комерції *B2C* система розрахунків між компаніями звичайно не має на увазі використання систем електронних платежів.

Як і у *B2C* бізнес-моделі у *B2B* можна розділити на чисті інтернет-компанії і змішані.

Серед чистих інтернет-компаній можна виділити електронні біржі, галузеві торгові майданчики, компанії, що здійснюють веб-хостинг, а серед змішаних компаній - веб-постачальників продукції і компанії, використовуючі нові комунікації для проведення рекламних кампаній, замовлень, логістики, післяпродажного обслуговування.

До сектора *C2B* відносяться компанії обробки цінових заявок, по яких споживачі хотіли б придбати товари і послуги. Функціонує така модель таким чином. Відвідувачі сайту залишають на ньому заявку на придбання якогось товару за ціною не вище визначеної. Ця заявка доступна всім компаніям, зареєстрованим в каталозі компанії. Якщо хтось з продавців вважає прийнятною пропонувану ціну, то він зв'язується з покупцем, і вони скоюють операцію.

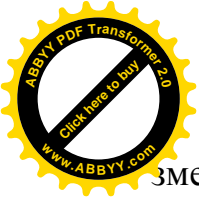
Сектор *C2C* ще знаходиться у стадії становлення, проте його роль дуже важлива. Він включає можливість взаємодії споживачів для обміну комерційною інформацією. Це може бути обмін досвідом придбання того або іншого товару, обмін досвідом взаємодії з фірмою і т.д. До сектора *C2C* відносяться електронні аукціони і торгові майданчики, де фізичні особи торгують один з одним.

До сектора *B2A* відноситься бізнес, що оперує операціями між компаніями і урядовими організаціями (адміністрацією). Він включає ділові зв'язки комерційних структур з державними організаціями, починаючи від місцевих властей і закінчуючи міжнародними організаціями. Інформація про деякі плановані урядом комерційні операції (закупівлях, замовленнях і т.д.) публікується в Інтернеті. При цьому через Інтернет приймаються всі заявки від всіх компаній, охочих взяти участь в цих операціях. По суті, уряд проводить *on-line-тендер*. Цей підхід покращує кількість інформації, сприяє зростанню попиту і пропозиції, дозволяє швидше здійснювати необхідні контакти, знижує транзакційні витрати.

Останнім часом уряди розвинених країн і керівництва міжнародних організацій прагнуть активно використовувати Інтернет для проведення закупівель товарів і послуг.

До сектора *C2A* відносяться всі види господарської діяльності, побудовані на організації взаємовідносин фізичних осіб і державних служб. Цей сектор також знаходиться у стадії розвитку, але має достатньо високий потенціал, який може бути використаний для організації взаємодії державних структур і споживачів, особливо в соціальній і податковій сфері.

У США налічується 80 тис. органів місцевої влади, які щорічно збирають близько 500 млрд дол. у вигляді штрафів і зборів за різні дозволи (забудова, ліцензії і т.п.) через Інтернет. До цього ж сектора відносяться збори податкових відомств. Транзакційні витрати при цих зборах не менше 10% і можуть бути



зменшені в 3-4 рази технологіями електронної комерції, не говорячи вже про підвищення зручності для місцевого населення.

Треба відзначити, що багато компаній прагнуть використовувати в своїй діяльності відразу декілька моделей. Наприклад, електронний магазин, що розміщує на своєму веб-сайті, рекламу іншої компанії або підприємства, можна віднести до секторів *B2B* і *B2C* одночасно.

## **5.2. Основні види бізнесу в ІНТЕРНЕТІ**

Компанії електронної комерції використовують різні види і форми бізнесу: електронні магазини, електронні вітрини, електронні біржі, електронні аукціони, електронні торгові майданчики, електронні рекламні або туристичні агентства, електронні центри консультантів і т.п.

На сьогоднішній день основу електронної комерції складає електронна торгівля. Найпоширенішим критерієм для класифікації компаній електронної комерції є рівень технологій, що застосовуються для організації торгового процесу. На практиці системи для торгівлі рідко бувають повністю автоматизованими, і по ступеню автоматизації такі системи можна класифікувати як веб-вітрини і електронні магазини.

### **5.2.1 Електронна вітрина**

*Електронна вітрина* - спеціалізований веб-сайт, на якому опублікована інформація про пропоновані товари і послуги, але відсутні функції автоматичного оформлення покупки я механізми прийому електронних платежів. Звично це достатньо прості і недорогі сайти, що представляють товари у вигляді стандартного каталога. Іноді вони можуть виробляти операції оформлення замовлення і виставляння рахунку, але на цьому етапі робота із замовленням переходить до менеджера з продажу.

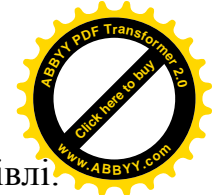
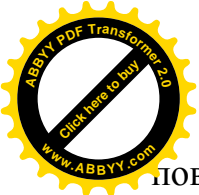
Далі менеджер організовує зв'язок з складом, доставку товару покупцю і прийом оплати за покупку. Паралель з операціями, що відносяться безпосередньо до продажу товарів, потрібне ретельне вивчення попиту, проведення рекламних заходів і т.д.

Таким чином, електронна вітрина не вимагає розробки і установки на сервері якого-небудь спеціального програмного забезпечення, і це дозволяє підготувати і запустити подібний проект за декілька днів. Проте така система вимагає найму додаткового кваліфікованого персоналу, а також не завжди зручна для покупців.

Проте електронна вітрина може стати оптимальним рішенням для компаній, торгуючих товарами різних спеціалізованих категорій, товарів, що не купуються в Інтернеті, або для компаній, що виробляють свій товар на замовлення під конкретні вимоги покупця.

### **5.2.2 Електронний магазин**

На відміну від електронних вітрин повнофункціональні електронні магазини звичайно оснащені спеціалізованим програмним забезпеченням,



повністю обслуговуючим технологічний процес електронної торгівлі. Електронний магазин здійснює демонстрацію товару відвідувачу, формування замовлення, підрахунок його повної вартості і автоматично взаємодіє з базами даних, що містять інформацію про наявність товарів на складі. Також він приймає платежі від утримувачів банківських карт і користувачів електронних платіжних систем, тобто виконує повний супровід операції від вибору товару до контролю за отриманням оплати.

Програмне забезпечення електронного магазину звичайно дозволяє організувати ряд допоміжних функцій, таких, як пошук по категоріях товарів, отримання додаткової інформації про товар, технічна підтримка користувачів і т.д.

Більшість сучасних програмних пакетів, призначених для створення електронного магазину, дозволяє виконати повний комплекс функцій магазину - від ідентифікації і авторизації користувача до підготовки бухгалтерської документації; вони мають механізми взаємодії з бухгалтерськими додатками, що помітно полегшує процес обліку продажів і підготовки податкової звітності.

Крім програмного забезпечення справжній електронний магазин повинен мати внутрішню інфраструктуру, приховану від очей покупця. У таку інфраструктуру можуть входити служба доставки товарів, склад і відділ постачання, технічний відділ, що здійснює цілодобову підтримку веб-сайта і програмного забезпечення, власна бухгалтерія і відділ реклами.

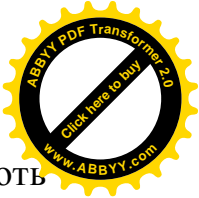
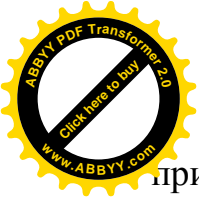
Таким чином, електронний магазин - це достатньо велике підприємство з розгалуженою внутрішньою структурою, повністю орієнтоване на отримання прибутку за рахунок прямих продажів якої-небудь продукції користувачам Інтернету.

Останнім часом сильно зросла популярність сервісних служб, що надають всім зацікавленим організаціям готові електронні магазини в оренду. В більшості випадків власникам магазину достатньо лише завантажити на сервер прайс-листи, опис пропонованих товарів, а також додаткові ілюстровані і текстові матеріали, після чого вони можуть приймати поступаючі замовлення. Безумовно, кількість пропонованих орендарям сервісних можливостей залежить від вартості оренди електронного магазину, використовуваного на сервері програмного і апаратного забезпечення і інших характеристик технічного майданчика.

### **5.2.3 Електронний аукціон**

Електронні аукціони - це спеціалізовані сервери, що надають своїм відвідувачам можливість здійснювати торги в режимі реального часу за допомогою мережі Інтернет. Для того, щоб стати учасником такого аукціону, достатньо реєструватися в базі даних серверу аукціону, після чого можна виставляти що-небудь на торги. При відкритті лота для нього встановлюється стартова ціна і тривалість торгів (звичайно вона складає часовий інтервал в 3, 5, 7, 10, 15, 20 або 30 днів). Після цього покупці підвищують мінімальну ставку з метою перебити ціну конкурентів. Як тільки закінчиться термін тривалості торгів, сервер автоматично направить продавцю і покупцю, що





призначив найвищу ціну, координати один одного, після чого вони вирішують питання обміну товару на гроші самостійно. Учасники торгів можуть обмінятися враженнями і обговорити операції, що вже відбулися, на спеціалізованих форумах аукціону.

У електронній комерції, також як і в традиційній, використовуються такі типи аукціонів, як стандартний, зворотний, німецький і т.д. Комерційна вигода організаторів електронного аукціону найчастіше полягає в продажу рекламної площі на сторінках сайту, оскільки відвідувана таких ресурсів вельми висока. Як правило, крім аукціону подібні сервери надають своїм клієнтам цілий ряд додаткових послуг, наприклад служби новин або доступ до електронної пошти, що помітно збільшує притоку відвідувачів і, відповідно, прибуток від реклами.

### 5.3 Електронні платіжні системи

*Платіжна система в Інтернеті* - це система проведення розрахунків між компанією і користувачем мережі Інтернет в процесі покупки/продажу товарів і послуг через Інтернет. У електронній комерції виділяють наступні основні механізми оплати товарів і послуг:

- оплата пластиковою картою, тобто переказ грошей з банківського рахунку покупця на рахунок продавця;
- дебетові схеми (некарткові платіжні системи);
- електронні гроші;
- традиційні методи оплати, наприклад оплата готівкою або поштовим переказом.

Кожний з цих механізмів оплати може бути реалізований декількома методами.

#### 5.3.1 Оплата пластиковою картою

На сьогоднішній день пластикові карти є поширеним засобом оплати. Така схема оплати схожа на звичну покупку в магазині: клієнт купує товар або послугу і передає продавцю для оплати номер своєї кредитної карти, гроші з якою знімаються і поступають на рахунок продавця.

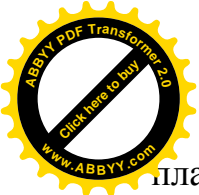
Компанія може організувати прийом платежів з пластикових карт за допомогою різних механізмів:

- 1) через власний рахунок продавця;
- 2) через платіжного посередника;
- 3) через платіжних посередників для роботи з фізичними лицами.

Перш ніж компанія-продавець дістане можливість приймати на своєму сайті пластикові картки, вона повинна відкрити в банку спеціальний рахунок, званий рахунком продавця. Рахунок продавця - це транзитний рахунок в банку, куди поступають гроші, одержані продавцем з пластикових карт покупців. Потім гроші переводяться на поточний (розрахунковий) рахунок продавця. Без такого рахунку прийом платежів пластиковими картками неможливий.

Відкрити власний рахунок продавця достатньо складно. Це доступно тільки юридичним особам. Банки з підозрою відносяться до тих, хто приймає





платежі через Інтернет, вони дуже ретельно перевіряють претендентів на отримання такого рахунку, оскільки це пов'язано з великими ризиками для самого банку; отже, невеликим компаніям одержати такий рахунок самостійно практично неможливо. У зв'язку з цим росте кількість фірм, які беруть на себе відкриття нерезидентам США рахунок продавця в американських банках. Докладна інформація про ці компанії, їх послуги і ціни представлена на веб-сайтах. Більшість таких компаній не тільки допомагає одержати рахунок продавця, **але також** займається реєстрацією підприємства за межею, відкриттям рахунків в банках.

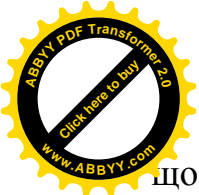
Механізм прийому оплати через спеціального платіжного посередника більш зручний, оскільки компанії-продавцю не потрібно думати про авторизацію пластикових карт, здійснювати власноручно транзакції, вирішувати проблеми оподаткування (податки платить тільки платіжний посередник, але не продавець, що користується його послугами) і т.д. Всім цим займається посередник. Недолік цього варіанту - вищий відсоток за обслуговування.

Платіжні посередники в електронній торгівлі - це фірми, які вже мають власний рахунок продавця і за певну платню надають іншим фірмам можливість проводити свої платежі через нього і забирати потім одержані гроші. Реєстрація в таких системах, як правило, стоїть від 50 до 200 дол. (деякі посередники не стягують ні платні за реєстрацію, ні абонентної платні, а тільки певний відсоток за кожну транзакцію).

В основному процедура оплати товарів в Інтернеті через спеціального посередника відбувається таким чином. Покупець підключається до веб-серверу магазину, формує корзину товарів. Магазин розміщує на своїх веб-сторінках спеціальні кнопки, наприклад з написом «Сплатити»; після ухвалення рішення про покупку користувач натискає на цю кнопку і переміщається на сервер посередника, на сторінку магазину, спеціально створену там посередником. Покупець вибирає тип карти, за допомогою якої буде вироблена оплата, вводить її реквізити і заповнює спеціальну форму, а після натиснення кнопки про підтвердження оплати автоматично повертається на сайт магазину. Платіжний посередник виробляє авторизацію карти: перевіряє, чи не закінчився термін придатності карти і чи досить грошей на рахунку. Якщо авторизація пройшла успішно, гроші покупця перераховуються на торговий рахунок посередника, а посередник, одержавши гроші, переводить їх на банківський рахунок магазину, і магазин відпускає товар. Звичайно ця процедура займає декілька хвилин.

Різні посередники встановлюють різні обмеження на те, що можна продавати з їх допомогою. Більшість з них готова працювати тільки з матеріальними товарами, оскільки ризик обману в цьому випадку менше, ніж при продажу нематеріальних товарів, таких, як програмне забезпечення або послуги консультантів, що пересилаються по електронній пошті або одержані з веб-сайта продавця.

Деяко спрощеним варіантом платіжних посередників є системи *Person-to-person-платежів*. Вони відрізняються від звичних платіжних посередників тим,



що призначені для роботи з фізичними лицями. Використовування подібних систем дозволяє одержати гроші на свій рахунок протягом 2-3 днів, тобто не довше, ніж при роботі із звичними платіжними посередниками.

### **5.3.2 Дебетові схеми (некарткові платіжні системи)**

Некарткові платіжні системи - це такі системи, в яких покупець спочатку реєструється, відкриває рахунок, переводить на нього гроші, а потім вже ними користується. В цьому випадку користувачу не треба вводити параметри своєї пластикової карти при кожній покупці, як це робиться при оплаті пластиковими картами. У некарткових системах забезпечується вищий ступінь безпеки і існує можливість встановлювати меншу платню за транзакції клієнтів, ніж в карткових системах (2-4% від суми платежу). Насправді переказів грошей з рахунку на рахунок в таких системах просто немає. Всі гроші всіх учасників комерційної операції лежать на єдиному рахунку в банку, і замість реальних транзакцій ведеться журнал їх переміщень між різними учасниками.

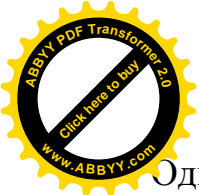
Незручність для покупця в цьому випадку полягає у тому, що на відміну від оплати кредитною картою, перш ніж купити що-небудь, покупець повинен спочатку реєструватися, створити свій обліковий запис в системі і внести на неї гроші, тобто покупцю доведеться вносити на свій рахунок гроші не як платня за товар або послугу, а авансом, з розрахунку на майбутнє.

### **5.3.3 Електронні гроші і смарт-карти**

*Електронні гроші* є спеціальними файлами, циркулюючими між покупцем, продавцем і банком, які виконують роль купюр і монет. Системи електронних грошей також можна віднести до дебетових схем. Як і в некарткових платіжних системах, покупець повинен спочатку внести гроші в систему, а потім їх витратити. У системах електронних грошей безпосереднє внесення грошей на транзитний рахунок в платіжній системі замінюється придбанням валюти, яку потім можна витратити при оплаті товарів або послуг через Інтернет.

У таких системах творці відмовилися від самої ідеї банківського рахунку. Роль грошей виконують файли - зобов'язання інтернет-банків. При цьому за допомогою сучасних криптографічних методів такі файли володіють всіма властивостями справжніх грошей. Зокрема, їх неможливо виготовити кому-небудь, окрім банку-емітента. Їх достовірність легко перевіряється. Спеціальні процедури виключають можливість копіювання електронних грошей. Іншими словами, це аналоги паперових банкнот, в яких роль водяних знаків і захисних смужок виконують протоколи захисту даних. Обороти таких грошей дуже дешевий і швидкий. Електронні купюри можна передавати по мережі, оплачуючи товари і послуги, скоюючи банківські операції, позичати, і т.д.

Для зберігання електронних грошей і проведення розрахунків звичайно потрібно встановити програму-гаманець, яка безкоштовно надається на сайтах власників електронних платіжних систем. Звичайно ці системи дозволяють зберігати електронні гроші в декількох валютах.



Одна з найпопулярніших систем електронних грошей - WebMoney дозволяє використовувати грошові знаки, еквівалентні рублю, долару, євро, а також переказувати кошти з однієї валюти в іншу за допомогою спеціальних програмних засобів. Для підключення до системи необхідно скористатися спеціальним програмним забезпеченням WebMoneyKeeper (гаманець).

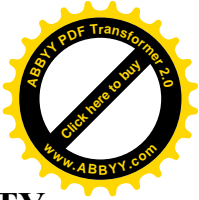
Перевагою електронних грошей є те, що їх можна застосовувати для мікроплатежів (менше 1 долара), що неможливо зробити при оплаті пластиковою картою. Використовуючи цифрову готівку, що знаходиться на жорсткому диску комп'ютера, покупець може робити покупки миттєво, причому з нього не стягують відсотки за транзакції. Але головна перевага електронних грошей - конфіденційність розрахунків. Коли покупець розплатується цифровою готівкою, продавець встановлює лише достовірність грошей, але не може визначити, хто саме зробив покупку.

Недоліком електронних грошей є те, що зберігаються вони на жорсткому диску комп'ютера, який може бути заражений комп'ютерним вірусом або підданий атаці хакера, внаслідок чого цілісність системи буде зруйнована і електронні гроші можуть бути втрачені.

*Смарт-карта* - це пластикова карта з своїм процесором, пам'яттю, програмним забезпеченням і системою введення-висновку інформації. Електронні гроші на базі смарт-карт можуть забезпечити необхідний рівень конфіденційності і анонімності, а також не вимагають зв'язку з центром для підтвердження оплати. У зв'язку з цим вартість транзакцій з використанням смарт-карт прагне до нуля, з'являється можливість заплатити один цент, не плативши за переказ ще 10-15 центів.

У міру розповсюдження прочитуючих пристроїв для карт такі системи можуть стати найпопулярнішими в Інтернеті і поза ним.

Електронні гроші зараховуються на смарт-карти за допомогою зняття засобів з реальних рахунків клієнтів і заміни їх на цифрові гроші на смарт-карті. Після зарахування електронних грошей на карту подальше відстежування їх переміщень між користувачами стає неможливим, тобто система, по суті, пропонує повний електронний еквівалент справжніх грошей. Передача електронних грошей між користувачами може здійснюватися по телефону.



## 6 ОРГАНІЗАЦІЯ КОМП'ЮТЕРНОЇ БЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

Те, що інформація має цінність, люди усвідомили дуже давно. Інформацію продукують, зберігають, транспортують, купують, продають, а отже, фальшують і крадуть. Саме тому інформацію потрібно захищати.

Сучасне суспільство стає все більше інформаційно залежним. Успіх багатьох видів діяльності безпосередньо залежить від володіння певними даними й відсутності цих даних у конкурентів. І чим більше проявляється такий ефект, тим нагальнішою стає потреба захисту інформації.

Щодо інформаційних систем, то в них дані також є найбільш цінною компонентою. Збитки від втрати даних або розголошення їх, затрати на повторне введення даних можуть набагато перевищувати вартість апаратних та програмних засобів, що використовуються для їх зберігання. Відомо немало випадків закриття компаній через втрату або крадіжку інформаційних ресурсів. Навіть інформація, що не є конфіденційною, може бути випадково знищена користувачем або свідомо зловмисником, може бути втрачена під час стихійного лиха. Тому одним з найважливіших завдань, що мусить бути вирішено при побудові інформаційної системи, є організація її безпеки.

Для кожного з нас забезпечення житла, автомобіля, офісу є звичною і необхідною справою. Інакше виглядає ситуація, коли мова йде про безпеку інформаційної системи. Часто лише після того як диск уражений вірусом або пошкоджений, користувачі починають задумуватись над організацією безпеки своїх систем, резервним копіюванням даних. Проте жодна система захисту не гарантує повної безпеки, а будь-який захист пов'язаний з певними затратами та незручностями для користувача.

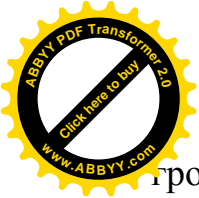
Тому актуальним є питання аналізу можливих загроз та ризиків для конкретної системи, вибору рішень щодо адекватного захисту системи за мінімуму затрат і незручностей.

### 6.1 Інформаційна політика

На нинішній час захист інформації - достатньо розвинута галузь науки і техніки, що пропонує на ринку широкий спектр різноманітних заходів й засобів для захисту даних. Проте жоден з них окремо взятий не в змозі гарантувати адекватну безпеку інформаційної системи. Надійний захист можливий лише за умови проведення **комплексу** взаємодоповняльних заходів. Можна виділити три основні складові такого комплексу:

- нормативно-правові засоби;
- адміністративні заходи;
- спеціальне обладнання та програмне забезпечення.

*Нормативно-правові засоби захисту* визначаються законодавчими акти держави, які регламентують правила використання, обробки та передачі інформації обмеженого доступу та встановлюють ступінь відповідальності за порушення цих правил. У ст. 34 Конституції України визначається право



громадян України на інформацію, забезпечення інформаційних процесів.

З липня 2003 року в Україні введена кримінальна відповідальність за незаконне втручання в роботу комп'ютерів і комп'ютерних мереж, а також за поширення комп'ютерних вірусів, що привело до спотворення, зникнення, блокування інформації чи її носіїв. Законом передбачено, що мінімальне покарання за правопорушення такого типу - це штраф у розмірі до 70-ти неоподатковуваних мінімумів доходів громадян, а максимальне - позбавлення волі терміном до 5-ти років.

До уваги також мусимо брати морально-етичні аспекти захисту, які реалізуються у вигляді різноманітних норм, що традиційно сформувалися в державі або в суспільстві. : *Сукупність адміністративних заходів та вибір спеціального обладнання і програмного забезпечення повинні здійснюватись для конкретної інформаційної системи. Слід завважити, що безпеку інформаційних систем не можна купити, її треба постійно підтримувати: контролювати, модернізувати та оновлювати. Займатися цими питаннями в першу чергу повинні керівники підприємств, відділи інформаційної безпеки, IT-менеджери або принаймні системні адміністратори.*

## 6.2 Політика безпеки

Незважаючи на те, що політика безпеки повинна розроблятися індивідуально для кожної конкретної інформаційної системи, існує ряд рекомендацій, які можна вважати загальними і керуватися ними, організуючи захист будь-яких інформаційних систем.

**Політика безпеки** - це формальний виклад правил, яких повинні дотримуватися особи, що одержують доступ до корпоративних технологій та інформації. Можна виділити такі чотири **етапи побудови політики безпеки**:

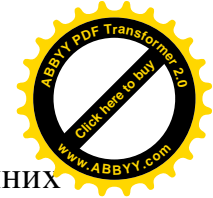
- реєстрація всіх ресурсів, які повинні бути захищені;
- аналіз та створення списків можливих загроз для кожного ресурсу;
- оцінка ймовірності появи кожної загрози;
- прийняття рішень, які дозволять економічно ефективно захистити інформаційну систему.

Далі розглянемо кожен з цих етапів.

Отже, першим кроком до створення політики безпеки є вибір об'єктів, що потребують захисту. Деякі з них є очевидними, наприклад, сервер баз даних або маршрутизатор. Нижче наведено перелік об'єктів, що можуть підлягати захисту:

- апаратне забезпечення: сервери різних служб, маршрутизатори, комутатори, концентратори, канали зв'язку, термінали, робочі станції, персональні комп'ютери, принтери, дисководи;
- програмне забезпечення: прикладне програмне забезпечення, різні | системні утиліти, лістинги коду програм, що використовуються;
- дані: збережені в реальному режимі, резервні копії і архіви, системні записи бази даних, паролі та записи обліку користувачів;
- персонал: користувачі, системні адміністратори, техніки;





- документація: інструкції до різного обладнання, журнали системних адміністраторів, контракти, угоди, технічні завдання;
- витратні матеріали: папір, друковані бланки, магнітні та оптичні носії інформації.

Деякі об'єкти цього списку не є очевидними. Пояснюється це тим, що лише незначна частина злочинів, пов'язаних з інформацією, здійснюється за допомогою високотехнологічних засобів. Здебільше крадіжка інформації відбувається через халатність персоналу або незнання базових правил безпеки.

Нижче наводиться список класичних загроз:

- несанкціонований доступ до ресурсів і/або інформації;
- ненавмисне розкриття інформації (наприклад, випадкове ознайомлення з паролем);
- різні види атак, що дозволяють проникнути в мережу або перехопити управління мережею;
- комп'ютерні віруси, в тому числі — мережеві хробаки, що модифікують і знищують інформацію або блокують роботу обчислювальних систем;
- логічні бомби - набори команд, що записуються в програму і спрацьовують за певних умов, наприклад, через визначений інтервал часу;
- „троянські коні" - програми, що виконують певні дії без відома власника зараженої системи, наприклад, відсилають через Інтернет із зараженого комп'ютера різну інформацію, включаючи паролі зареєстрованих користувачів;
- засоби пригальмовування обміну даними в мережах;
- природні катаклізми.

Щодо природних катаклізмів, то для кожного регіону вони мають свої особливості, проте переважно єдиним ефективним захистом від них є створення резервних копій цінної інформації, які потрібно зберігати в надійному місці (найкраще кілька копій в різних географічно віддалених місцях). Очевидно, що наведений вище список може бути доповнений загрозами, що є характерними для конкретної інформаційної системи. Крім того, потрібно пам'ятати, що методи та засоби вторгнень в інформаційні системи постійно розвиваються. В результаті можна стверджувати, що стовідсотковий захист від усіх типів загроз забезпечити неможливо.

Далі потрібно оцінити ймовірність появи кожної загрози та можливі втрати (збитки) за її появи. В результаті такого аналізу всі можливі загрози можна класифікувати як малонебезпечні, небезпечні та критичні для цієї інформаційної системи.

Сама собою система безпеки інформаційної системи не приносить прибутку, проте її відсутність може бути причиною великих збитків. Прикладами можливих збитків є:

- втрата конфіденційності (несанкціонований доступ до інформації);





- втрата даних (спотворення або знищення файлів);
- відмова в обслуговуванні користувачів системою;
- втрата репутації.

Останнє може призвести до відмови користувачів бути клієнтами цієї інформаційної системи. Найважче оцінити фінансові збитки від того, що якась кількість клієнтів після злому покинуть компанію, а потенційні клієнти не оберуть її. В результаті компанії рідко повідомляють про вторгнення в їхні інформаційні системи та крадіжки даних.

На останньому етапі здійснюється вибір загроз, від яких буде реалізовуватись захист, обсягів та черговості робіт.

Визначаючи загрози, для яких буде створюватись захист, необхідно брати до уваги і затрати на його реалізацію. Серед них, наприклад, матеріальні затрати на купівлю обладнання та програмного забезпечення, затрати на шифрування й дешифрування. Витрати на захист від кожної загрози мають бути адекватні можливим наслідкам цієї загрози з врахуванням ймовірності їх появи.

Вартість засобів захисту від певного виду загроз не повинна перевищувати втрат, до яких може спричинити ця загроза, в тому числі і втрат на відновлення інформаційної системи.

Тільки реалізуючи повною мірою необхідний внесок у захист на кожному перерахованому вище етапі, можна впевнено почувати себе при роботі з інформаційними ресурсами.

Коли говорять про безпеку інформаційних систем, звичайно мають на увазі захист даних від зловмисників або ненавмисного порушення конфіденційності. Проте безпека даних практично в кожній інформаційній системі суттєво залежить і від використання джерел безперебійного живлення, і від наявності надійних резервних копій. Резервне копіювання - один з найбільш ефективних способів, що дозволяє застрахуватися від втрати даних, причиною якої є збої апаратного забезпечення чи природні катаклізми.

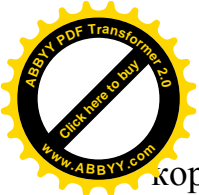
### **6.3 Сервіси безпеки та механізми її порушень**

**Сервіс безпеки** - це сукупність механізмів, процедур та інших заходів управління для зменшення ризиків, пов'язаних з загрозою втрати або розкриття даних. Одні сервіси забезпечують захист від загроз, інші - виявляють слабкі місця в системі безпеки.

Основними сервісами безпеки є:

- сервіс автентифікації;
- сервіс конфіденційності;
- сервіс цілісності;
- сервіс дотримання зобов'язань.

Першим етапом на шляху захисту ресурсів і інформаційної системи є організація перевірки, чи є користувач, який входить в систему, тим за кого він себе видає. Сама процедура перевірки носить назву *автентифікації*



користувача. Як правило, процедура автентифікації складається з двох кроків: ідентифікації та верифікації.

Під *ідентифікацією* розуміють процедуру представлення користувача системі. Звичайно це відбувається шляхом уведення імені, під яким користувач зареєстрований в цій системі.

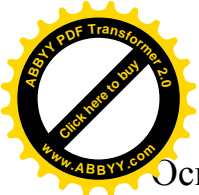
Другий крок автентифікації - верифікація. *Верифікація* - це процедура, яку система виконує для того, щоб переконатися, що користувач, який входить в систему, є саме тим, чиє ім'я він ввів при ідентифікації. Для цього користувачу пропонується ввести пароль, який буде порівняний з паролем в записі обліку цього користувача.

Використання для автентифікації клієнта двох елементів утруднює нелегальне проникнення в систему, оскільки для успішної атаки необхідно подолати два незалежні бар'єри. Якщо користувач успішно пройшов ідентифікацію (вказав ім'я, що є зареєстроване в системі) та верифікацію (ввів пароль, що відповідає цьому імені), автентифікація закінчена. Кожному користувачу, який пройшов процедуру автентифікації, надаються певні права доступу до ресурсів системи.

Обліковий запис користувача може передбачати часові обмеження, наприклад, вхід в систему лише в робочий час та обмеження місця роботи, приміром, дозвіл працювати лише на машинах відповідного департаменту. У разі виявлення, помилки під час автентифікації автоматично виконуються спеціальні процедури. Це може бути виведення на екран повідомлення про некоректність введених даних та запрошення повторно ввести інформацію. Якщо кількість невдалих спроб входу в систему перевищила певне число, користувачу може бути відмовлено в повторних спробах автентифікації. Для відновлення прав на вхід в систему може бути запропоновано відповісти на секретне запитання, відповідь на яке зафіксована в системі при реєстрації або перереєстрації користувача. Параметром, що служить для прийняття рішень про неможливість повторних спроб входу в систему, може бути час, що виділяється користувачу на здійснення успішного входу в систему. Право на повторний вхід в систему може бути також поновлено після особистого звернення до адміністратора.

Серед альтернативних методів автентифікації варто відзначити біометричні. В наш час вони стають поширеними не лише у надсекретних організаціях, а й в системах масового користування. Наприклад, сучасні ноутбуки та кишенькові комп'ютери часто обладнуються сканерами відбитків пальців.

*Біометрія* - це наука, що займається вимірюванням характеристик людського організму. Такі фізичні характеристики, як відбитки пальців, рисунок сітківки та райдужної оболонки ока, голос та інші, можуть бути використані як засоби для автентифікації. У процесі реєстрації нового користувача певна фізична характеристика зчитується і записується в базу даних для наступного використання. Ця записана характеристика називається шаблоном. При виконанні автентифікації відповідна характеристика особи зчитується, перетворюється в цифровий код і порівнюється з шаблоном.



Оскільки біометричні вимірювання майже завжди мають варіації, під час порівняння вимагається попадання в певну зону значень, а не повний збіг з шаблоном. На основі результатів порівняння приймається рішення про те, чи користувач, особа якого перевіряється, є дійсно тим, за кого себе видає.

Існують такі методи розпізнавання особи за біометричними характеристиками:

- розпізнавання за відбитками пальців;
- оптичне розпізнавання сітківки та райдужної оболонки ока;
- розпізнавання за голосом;
- розпізнавання підпису;
- розпізнавання за рисами обличчя;
- розпізнавання за динамікою введення тексту.

Остання технологія базується на тому, що кожній людині властива своя манера роботи на клавіатурі. В пам'ять комп'ютера вноситься не тільки ім'я і пароль користувача, а і його „клавіатурний портрет“, який визначається динамікою вводу символів.

Конфіденційність означає, що доступ до інформації може бути наданий тільки тим суб'єктам, що мають на це право. Якщо необхідно, для забезпечення конфіденційності використовується шифрування даних.

Сервіс цілісності даних забезпечує захист від навмисної або випадкової зміни даних. Захисту потребує, наприклад, інформація, що зберігається в базі даних або передається по каналах зв'язку. Електронний характер цих даних має ряд властивостей, що ускладнюють їх захист.

Електронний документ - це послідовність двійкових бітів, тому встановити, чи ця послідовність є оригіналом чи його спотвореною копією. Зміна бітів в пам'яті комп'ютера або в потоці, що передається в каналі зв'язку, не залишає фізичних слідів. Для захисту електронних документів можуть використовуватися криптографічні контрольні суми. Сервіс цілісності даних дозволяє виявити факт зміни, часткового вилучення і доповнення даних.

Сервіс дотримання зобов'язань гарантує, що учасники інформаційного обміну не зможуть заперечити факт своєї участі в ньому. Тобто, наприклад, відправник не може відмовитися від факту передачі даних, адресат - від факту їх прийому. Цей сервіс може бути реалізований за рахунок використання цифрового підпису, що буде розглянуто дещо пізніше. Залежно від профілю інформаційної системи на першому місці можуть бути різні вимоги:

- забезпечення конфіденційності інформації;
- забезпечення цілісності даних, наприклад, неможливості внесення змін в | платіжні доручення;
- забезпечення безвідмовної роботи системи, наприклад, надання певних послуг.

Рис. 6.1 ілюструє нормальний потік інформації та механізми порушень його безпеки.

Виділяють такі чотири механізми порушень безпеки даних.

- **Роз'єднання.** Ресурс системи знищується або стає недоступним для використання (рис. 6.1б). При цьому порушується доступність даних.

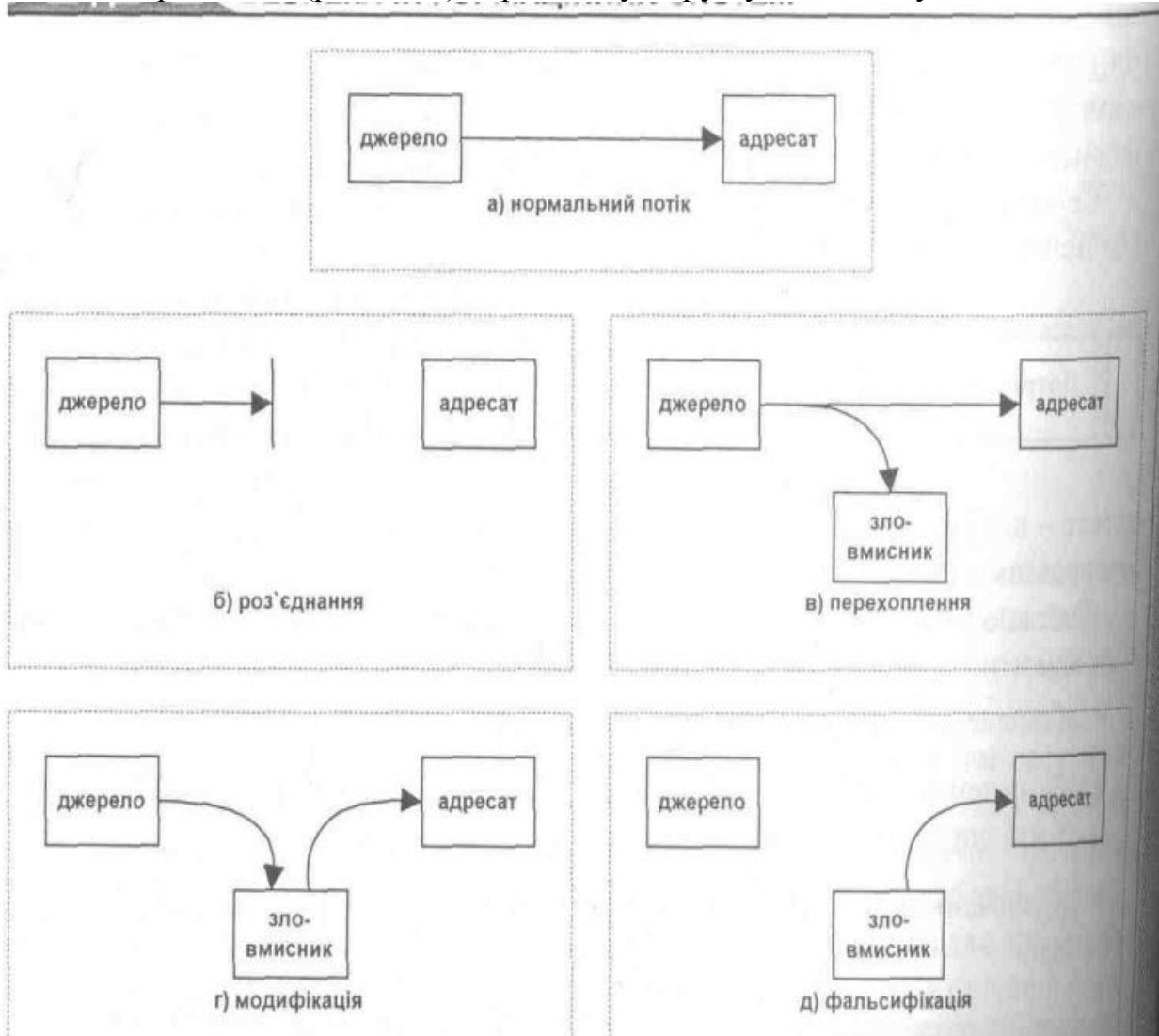
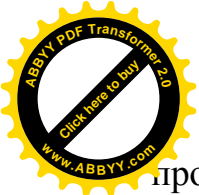


Рис. 6.1 - Механізми порушень безпеки даних

- **Перехоплення.** До ресурсу відкривається несанкціонований доступ (рис. 6.1в). В цьому випадку порушується конфіденційність даних.
- **Модифікація.** До ресурсу не тільки відкривається несанкціонований доступ, | зловмисник може також змінити ресурс (рис. 6.1г). В такому випадку порушується цілісність даних. Адресат може сприйняти модифікований зловмисником файл як оригінал.
- **Фальсифікація.** Зловмисником в систему вноситься підставний об'єкт (рис. 6.1д). В цій ситуації порушується автентичність.

#### 6.4. Шифрування даних

Унеможливити читання даних сторонніми особами дозволяють



процедури шифрування.

**Шифрування** - це перетворення даних у форму, що не дає можливості безпосереднього сприйняття зашифрованої інформації. ||i

Шифрування здійснюється з використанням криптографічного ключа. З використанням ключа здійснюється і зворотна процедура *дешифрування*, за допомогою якої відбувається приведення зашифрованої інформації до первинного варіанта.

*Криптографія* - це наука засекречування інформації, тобто перетворення її в форму, яка недоступна для прочитання без наявності ключа. Наука, що займається розробкою методик і алгоритмів читання зашифрованих текстів без знання ключа, називається криптоаналізом.

Спроба криптоаналітика порушити конфіденційність обміну інформації називається криптографічною атакою. Відповідно результатом успішного проведення криптографічної атаки є злом шифру.

Якщо атака базується на прямому переборі ключів, то час, необхідний для успішного подолання шифру, пропорційний довжині ключа. Сьогодні найбільш популярною довжиною ключа є 128 біт.

*Криптографічна стійкість (криптостійкість)* - це стійкість шифросистеми до всіх відомих видів криптоаналізу. Прикладом застосування криптографії є банківські послуги і платежі в І діалоговому режимі. Електронний бізнес може існувати тільки за умови конфіденційності інформації, одним з найкращих методів забезпечення якої є криптографія.

#### 6.4.1 Симетрична криптографія

Якщо відправник і одержувач користуються одним і тим же ключем, тобто шифрування і дешифрування використовується один ключ, то така методика і шифрування називається симетричною криптографією. Симетричне шифрування зручне для обміну інформацією в Інтернеті. В інформаційних системах, у яких відправник і одержувач не є віддалені один від одного і мають можливість конфіденційно домовитись про ключ шифрування. Ключі також треба зберігати в безпеці. Захищене зберігання ключів і забезпечення їх доступності для використання - це окрема задача, для розв'язання якої використовується спеціальна методика, що називається управління ключами. Особливість полягає в тому, що шифрування використовується для великих файлів (наприклад, мегабайт даних), а ключ має невелику довжину, наприклад, 16 байт.

Для зберігання ключів існують і апаратні засоби. Це мініатюрні пристрої, серед яких найбільш поширені пластикові смарт-картки, що приєднуються до портів комп'ютерів. Деякі з цих пристроїв дуже прості, повільно працюють, мають малий об'єм пам'яті та обмежений набір функцій, інші з них є більш потужними. Перевагою застосування пристроїв для зберігання ключів є те, що злодіє не має можливості атакувати їх 24 години на добу, а лише кілька секунд при підключенні пристрою до комп'ютера. Пристрої для зберігання ключів можуть використовуватися і для зберігання паролів.



## 6.4.2 Асиметрична криптографія

Асиметрична криптографія передбачає використання для шифрування і дешифрування двох різних ключів (рис.6.2).

Для шифрування використовується загальний (відкритий) ключ. Цей ключ не є таємним. Загальний ключ повинен бути відомий всім, хто хоче відправити зашифроване повідомлення. Дешифрування може бути здійснене тільки з використанням таємного ключа. Загальний (відкритий) та таємний (закритий) ключі зв'язані між собою математичною функцією. Якщо володар таємного ключа зберігає свій ключ в таємниці, зашифрований за допомогою відкритого ключа текст залишається в безпеці.

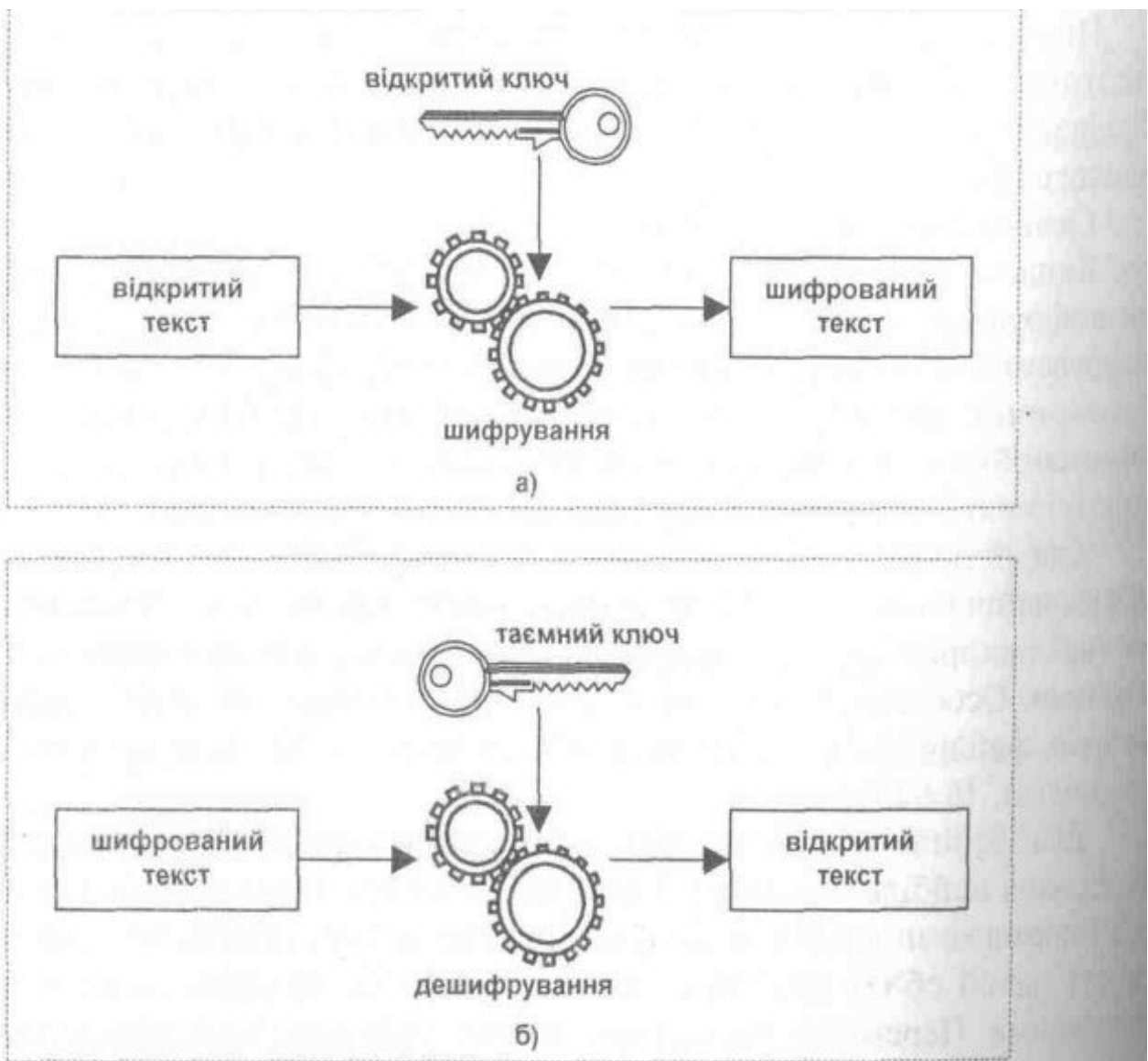
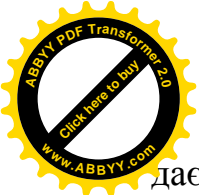


Рис. 6.2 - Асиметрична криптографія: а) шифрування відкритим ключем; б) дешифрування таємним ключем

Загальний ключ може бути відісланий через Інтернет для використання при шифруванні, оскільки його перехоплення не становить загрози - він не





дає можливості розшифрувати інформацію.

Сучасні системи захисту інформації часто поєднують симетричну та асиметричну схеми шифрування. Наприклад асиметричні алгоритми використовують для розповсюдження ключів, а симетричні - для безпосереднього шифрування даних. Останнє зумовлено, в першу чергу, тим, що симетричні алгоритми мають значно вищу швидкодію, ніж асиметричні.

### 6.4.3 Цифровий підпис

Алгоритми асиметричного шифрування можуть використовуватись для створення цифрових підписів. Ідея використання цифрового підпису базується на двох фундаментальних положеннях:

- таємний ключ захищений і доступ до нього має тільки власник;
- єдиний спосіб створити цифровий підпис - це використати секретний ключ.

В режимі шифрування даних (рис. 6.2) відкритий ключ використовується для шифрування, а закритий - для дешифрування. В режимі створення цифрового підпису - навпаки: на етапі шифрування використовують таємний ключ, а на етапі дешифрування - відкритий (рис. 6.3).

Оскільки відкритий ключ є вільно доступним, то будь-хто зможе прочитати інформацію, зашифровану таємним ключем. Це не дає можливості засекречувати дані, проте дозволяє засвідчити цілісність змісту повідомлення.

Зашифрований текст розшифровують за допомогою відкритого ключа. Якщо відкритий ключ розшифрував дані, то вони були зашифровані зв'язаним з ним таємним ключем. Якщо при дешифруванні відкритим ключем повідомлення не розшифровується (не стає читабельним), то повідомлення не було зашифроване зв'язаним з ним таємним ключем або було спотворене після шифрування. Такий прийом дозволяє здійснювати надійну автентифікацію повідомлень. Звідси випливає, що все, що ви зашифрували за допомогою вашого таємного ключа, є вашим цифровим підписом.

Цифровий підпис є унікальним для повідомлення. Кожна комбінація повідомлення і таємного ключа буде видавати різні підписи.

Оскільки асиметричні алгоритми працюють повільно, то часто з мотивів : продуктивності підписують не все повідомлення, а дані, які це повідомлення представляють. Дані, які представляють повідомлення, формують за певними правилами і називають *дайджестом повідомлення*.

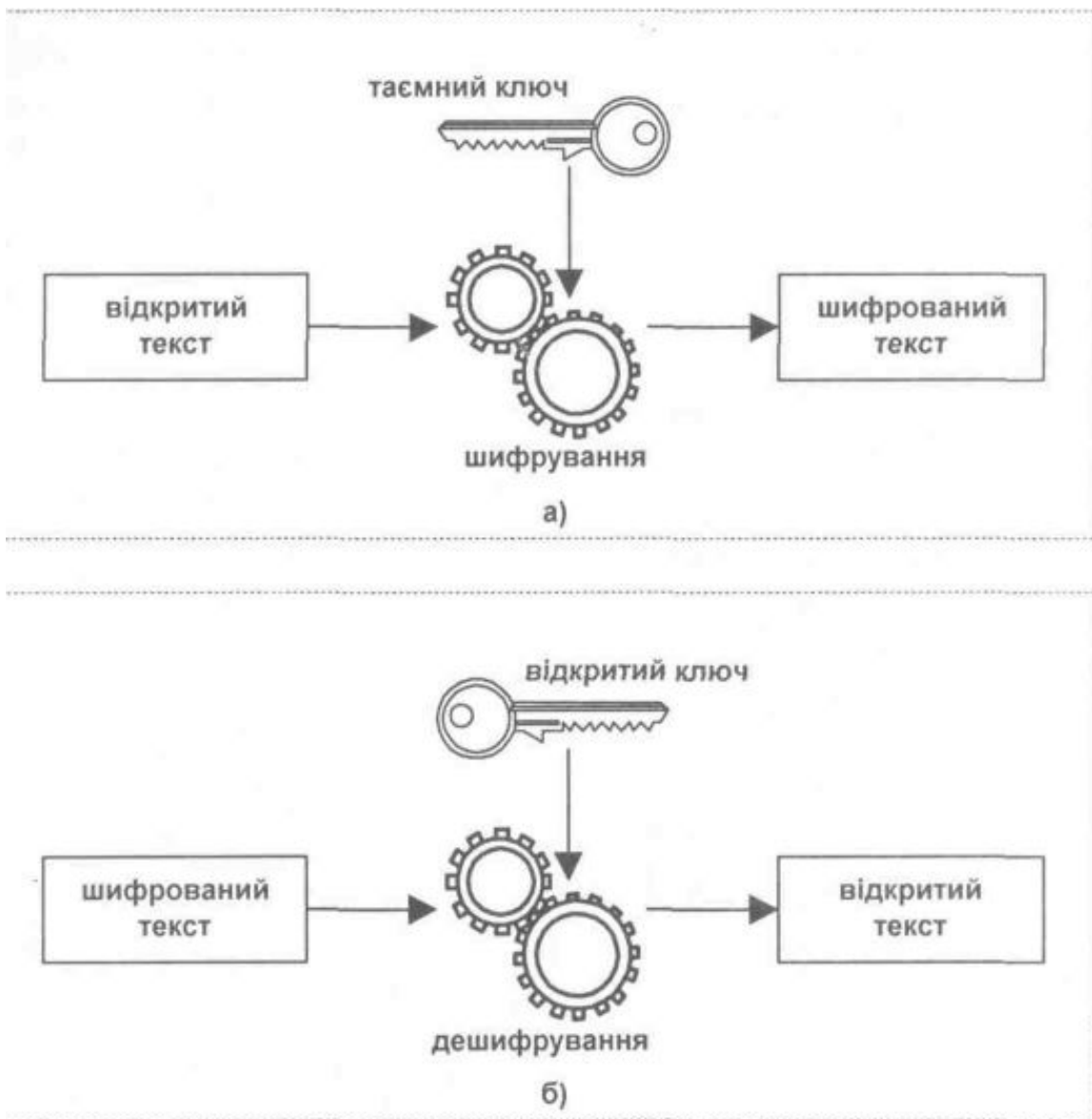


Рис. 6.3 - Схема використання асиметричної криптографії для цифрового підпису

Цифровий підпис - це конкретна реалізація криптографічної системи з відкритим ключем. Існує ще поняття електронного підпису. *Електронний підпис* - це будь-який знак або процедура, реалізовані електронними засобами та призначені для зв'язання запису з зобов'язаннями або завірення справжності запису. Прикладами електронного підпису може бути вихідний сигнал складного біометричного пристрою, який працює на розпізнаванні відбитків пальців, а може бути просто ім'я, введене наприкінці повідомлення, що буде відправлене електронною поштою.

#### *Цифровий сертифікат*

Для того, щоб бути переконаним, що відкритий ключ є власністю суб'єкта, який про це декларує, можна скористатись *цифровим сертифікатом*. Цифровий сертифікат зв'язує ім'я з відкритим ключем. Він створюється для

того, щоб можна було виявити несанкціоновану заміну імені або відкритого ключа. Перегляд цифрового сертифікату повинен дати можливість виявити факт підміни. *Цифровий сертифікат* - це ім'я, відкритий ключ та підпис (рис. 6.4). При цьому повідомленням вважається ім'я та відкритий ключ, які підписуються. Підпис практично завжди ставить вповноважений сертифікаційний центр.

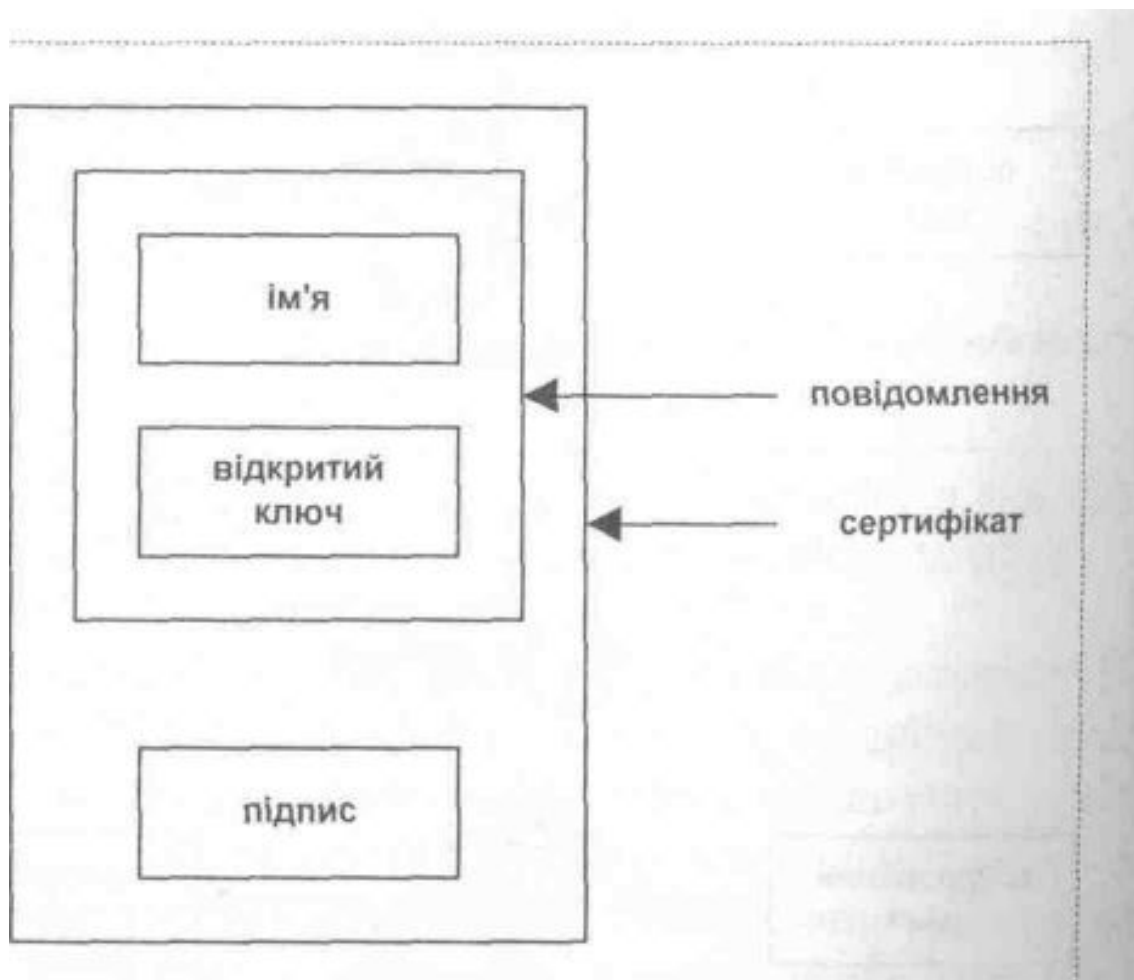
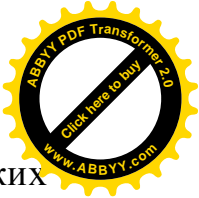


Рис. 6.4 - Цифровий сертифікат

## 6.5 Комп'ютерні віруси

**Комп'ютерним вірусом** називається програма, здатна створювати свої копії (не обов'язково цілком співпадаючі з оригіналом) і впроваджувати них у різні об'єкти або ресурси комп'ютерних систем, мереж і так далі без ведення користувача. При цьому копії зберігають здатність подальшого поширення. На сьогоднішній день відомо 6 основних типів вірусів: файлові, завантажувальні, примари (поліморфні), невидимки, скрипт-віруси і макро-віруси. Варто відрізнити віруси від шкідливих кодів. До них відносяться Інтернет-хробаки і програми, що одержали назву "Троянські коні".



Основні симптоми вірусної поразки: уповільнення роботи деяких програм, збільшення розмірів файлів (особливо виконуваних), поява не існуючих раніше підозрілих файлів, зменшення обсягу доступної оперативної пам'яті (у порівнянні зі звичайним режимом роботи), раптово виникаючі різноманітні відео і звукові ефекти. При всіх перерахованих вище симптомах, а також при інших дивних проявах у роботі системи (хитлива робота, часті самостійні перезавантаження та інше) варто негайно зробити перевірку системи на наявність вірусів.

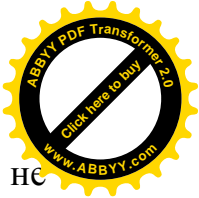
### **6.5.1 Зародження комп'ютерних вірусів**

Про появу першого комп'ютерного вірусу багато різних думок. Додоповільно тільки відомо, що на машині Чарльза Бэббиджа, що вважається винахідником першого комп'ютера, його не було, а на Univax 1108 і IBM 360/370, у середині 1970-х років вони вже були. Цікаво, що ідея комп'ютерних вірусів з'явилася набагато раніш самих персональних комп'ютерів. Крапкою відліку можна вважати праці відомого вченого Джона фон Неймана по вивченню математичних автоматів, що самовідтворюються, про які стало відомо в 1940-х роках. У 1951 році він запропонував спосіб створення таких автоматів. А в 1959 році журнал Scientific American опублікував статтю Л.С. Пенроуза, присвячену механічним структурам, що самовідтворюються. У ній була описана найпростіша двовимірна модель механічних структур, що самовідтворюються, здатних до активації, розмноженню, мутаціям, захопленню. Пізніше інший учений Ф.Ж. Шталь реалізував дану модель на практиці за допомогою машинного коду на IBM 650.

### **6.5.2 Шляхи проникнення вірусів у комп'ютер і механізм розподілу вірусних програм**

Основними шляхами проникнення вірусів у комп'ютер є знімні диски (гнучкі і лазерні), а також комп'ютерні мережі. Зараження твердого диска вірусами може відбутися при завантаженні програми з дискети, що містить вірус. Таке зараження може бути і випадковим, наприклад, якщо дискету не вийняли з дисководу А и перезавантажили комп'ютер, при цьому дискета може бути і не системної. Заразити дискету набагато простіше. На неї вірус може потрапити, навіть якщо дискету просто вставили в дисковод зараженого комп'ютера і, наприклад, прочитали її зміст.

Вірус, як правило, впроваджується в робочу програму таким чином, щоб при її запуску керування спочатку передалося йому і тільки після виконання всіх його команд знову повернулося до робочої програми. Одержавши доступ до керування, вірус насамперед переписує сам себе в іншу робочу програму і заражає неї. Після запуску програми, що містить вірус, стає можливим зараження інших файлів. Найбільше часто вірусом заражаються завантажувальний сектор диска і файли, що виконуються, що мають розширення EXE, COM, SYS, BAT. Украй рідко заражаються текстові файли.



Після зараження програми вірус може виконати яку-небудь диверсію, не занадто серйозну, щоб не залучити уваги. І нарешті, не забуває повернути керування тій програмі, з якої був запущений. Кожне виконання зараженої програми переносить вірус у наступну. Таким чином, заразиться все програмне забезпечення.

Для ілюстрації процесу зараження комп'ютерної програми вірусом має сенс уподібнити дискову пам'ять старомодному архівові з папками на тасьмі. У папках розташовані програми, а послідовність операцій по упровадженню вірусу буде в цьому випадку виглядати в такий спосіб.

### **6.5.3 Ознаки появи вірусів**

При зараженні комп'ютера вірусом важливо його знайти. Для цього варто знати про основні ознаки прояву вірусів. До них можна віднести наступні:

1. припинення роботи або неправильна робота раніше що успішно функціонували програм
2. повільна робота комп'ютера
3. неможливість завантаження операційної системи
4. зникнення файлів і каталогів або перекручування їхнього вмісту
5. зміна дати і часу модифікації файлів
6. зміна розмірів файлів
7. несподіване значне збільшення кількості файлів на диску
8. істотне зменшення розміру вільної оперативної пам'яті
9. висновок на екран непередбачених повідомлень або зображень
10. подача непередбачених звукових сигналів
11. часті зависання і збої в роботі комп'ютера

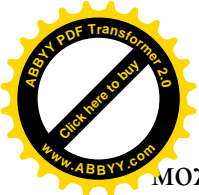
Слід зазначити, що перераховані вище явища необов'язково викликаються присутністю вірусу, а можуть бути наслідком інших причин. Тому завжди утруднена правильна діагностика стану комп'ютера.

### **6.5.4 Методи боротьби з вірусами**

Способи протидії комп'ютерним вірусам можна розділити на трохи груп: профілактика вірусного зараження і зменшення передбачуваного збитку від такого зараження; методика використання антивірусних програм, у тому числі знешкодження і видалення відомого вірусу; способи виявлення і видалення невідомого вірусу.

Найбільш ефективні в боротьбі з комп'ютерними вірусами антивірусні програми. Однак відразу хотілося б відзначити, що не існує антивірусів, що гарантують стовідсотковий захист від вірусів, і заяви про існуванні таких систем можна розцінити як або несумлінну рекламу, або непрофесіоналізм. Таких систем не існує, оскільки на будь-який алгоритм антивірусу завжди





можна запропонувати контр-алгоритм вірусу, невидимого для цього антивірусу (зворотне, на щастя, теж вірно: на будь-який алгоритм вірусу завжди можна створити антивірус). Більш того, неможливість існування абсолютного антивірусу була доведена математично на основі теорії кінцевих автоматів, автор доказу - Фред Коэн.

Варто також звернути увагу на кілька термінів, застосовуваних при обговоренні антивірусних програм:

*"Помилкове спрацьовування"* (False positive) - детектування вірусу в незараженому об'єкті (файлі, секторі або системній пам'яті). Зворотний термін – *"False negative"*, тобто недетектування вірусу в зараженому об'єкті.

*"Сканування по запиті"* ("on-demand") - пошук вірусів по запиті користувача. У цьому режимі антивірусна програма неактивна доти, поки не буде викликана користувачем з командного рядка, командного файлу або програми-розкладу (system scheduler).

*"Сканування на-летові"* ("real-time", "on-the-fly") - постійна перевірка на віруси об'єктів, до яких відбувається звертання (запуск, відкриття, створення і т.п.). У цьому режимі антивірус постійно активний, він присутній у пам'яті "резиденте" і перевіряє об'єкти без запиту користувача.

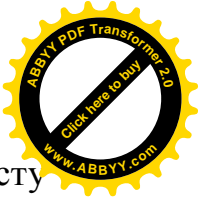
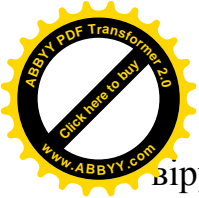
## **6.6 Класифікація антивірусів**

### **6.6.1 Сканери**

Принцип роботи антивірусних сканерів заснований на перевірці файлів, секторів і системної пам'яті і пошуку в них відомому і нових (невдомих сканерів) вірусів. Для пошуку відомих вірусів використовуються так звані "маски". Маскою вірусу є деяка постійна послідовність коду, специфічна для цього конкретного вірусу. Якщо вірус не містить постійної маски, або довжина цієї маски недостатньо велика, то використовуються інші методи. Прикладом такого методу є алгоритмічна мова, що описує усі можливі варіанти коду, що можуть зустрітися при зараженні подібного типу вірусом. Такий підхід використовується деякими антивірусами для детектування поліморфних вірусів.

У багатьох сканерах використовуються також алгоритми "евристичного сканування", тобто аналіз послідовності команд в об'єкті, що перевіряється, набір деякої статистики й ухвалення рішення ("можливо заражений" або "не заражений") для кожного об'єкта, що перевіряється. Оскільки евристичне сканування є багато в чому імовірнісним методом пошуку вірусів, то на його поширюються багато законів теорії імовірностей. Наприклад, чим вище відсоток вірусів, що виявляються, тим більше кількість помилкових спрацьовувань. Сканери також можна розділити на двох категорій – "універсальні" і "спеціалізовані". Універсальні сканери розраховані на пошук і знешкодження всіх типів вірусів поза залежністю від операційної системи, на роботу в якій розрахований сканер. Спеціалізовані сканери призначені для знешкодження обмеженого числа вірусів або тільки одного їхнього класу, наприклад макро-вірусів. Спеціалізовані сканери, розраховані тільки на макро-





віруси, часто виявляються найбільш зручним і надійним рішенням для захисту систем документообігу в середовищах MS Word і MS Excel.

Сканери також поділяються на «резиденті» (монітори), що роблять сканування "на-летові", і "нерезидентні", що забезпечують перевірку системи тільки по запиті. Як правило, "резиденті" сканери забезпечують більш надійний захист системи, оскільки вони негайно реагують на появу вірусу, у той час як "нерезидентний" сканер здатний пізнати вірус тільки під час свого чергового запуску.

До достоїнств сканерів усіх типів відноситься їхня універсальність, до недоліків - розміри антивірусних баз, що сканерам приходится «тягати за собою», і відносно невелику швидкість пошуку вірусів.

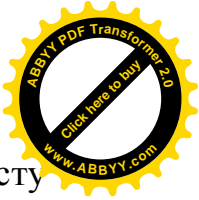
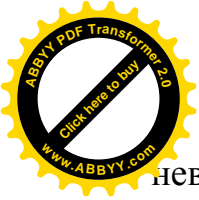
### **6.6.2 CRC-сканери**

Принцип роботи CRC-сканерів заснований на підрахунку CRC-сум (контрольних сум) для присутніх на диску файлів (системних секторів). Ці CRC-суми потім зберігаються в базі даних антивірусу, як, утім, і деяка інша інформація: довжини файлів, дати їхньої останньої модифікації і т.д. При наступному запуску CRC-сканери звіряють дані, що утримуються в базі даних, з реально підрахованими значеннями. Якщо інформація про файл, записана в базі даних, не збігається з реальними значеннями, то CRC-сканери сигналізують про те, що файл був змінений або заражений вірусом. CRC-сканери, що використовують анти-стелс алгоритми, є досить сильною зброєю проти вірусів: практично 100% вірусів виявляються виявленими майже відразу після їхньої появи на комп'ютері. Однак у цього типу антивірусів є уроджений недолік, що помітно знижує їх ефективність. Цей недолік полягає в тому, що CRC-сканери не здатні піймати вірус у момент його появи в системі, а роблять це лише через якийсь час, уже після того, як вірус розійшовся по комп'ютері. CRC-сканери не можуть визначити вірус у нових файлах (в електронній пошті, на дискетах, у файлах, відновлюваних з backup або при розпакуванні файлів з архіву), оскільки в їхніх базах даних відсутня інформація про ці файли. Більш того, періодично з'являються віруси, що використовують цю "слабкість" CRC-сканерів, заражають тільки знову створювані файли і залишаються, таким чином, невидимими для них.

### **6.6.3 Блокуючи антивіруси**

Блокуючи антивірус антивіруси - це резидентні програми, що перехоплюють "вірусно-небезпечні" ситуації і, що повідомляють про це користувачеві. До "вірусно-небезпечних" відносяться виклики на відкриття для запису у виконувани файли, запис у boot-сектори дисків або MBR вінчестера, спроби програм залишитися резидентне і т.п., тобто виклики, що характерні для вірусів у моменти розмноження.

До достоїнств блокуючих антивірусів відноситься їхня здатність виявляти і зупиняти вірус на самій ранній стадії його розмноження, що, до речі, буває дуже корисно у випадках, коли давно відомий вірус постійно «виповзає



невідомо відкіля». До недоліків відносяться існування шляхів обходу захисту блокуючих антивірусів і велика кількість помилкових спрацьовувань, що, видимо, і послужило причиною для практично повного відмовлення користувачів від подібного роду антивірусних програм. Необхідно також відзначити такий напрямок антивірусних засобів, як блокуючи антивіруси, виконані у виді апаратних компонентів комп'ютера. Найбільш розповсюдженим є вбудований в BIOS захист від запису в MBR вінчестера. Однак, як і у випадку з програмними блокуючими антивірусами, такий захист легко обійти прямим записом у порти контролера диска, а запуск DOS-утиліти FDISK негайно викликає «помилкове спрацьовування» захисту.

Існує трохи більш універсальних апаратних блокуючих антивірусів, але до перерахованих вище недоліків додаються також проблеми сумісності зі стандартними конфігураціями комп'ютерів і складності при їхній установці і налаштуванні. Усе це робить апаратні блокуючи антивіруси вкрай непопулярними на тлі інших типів антивірусного захисту.

#### 6.6.4 Імунізатори

Імунізатори поділяються на два типи: імунізатори, що повідомляють про зараження, та імунізатори, що блокують зараження яким-небудь типом вірусу. Перші звичайно записуються в кінець файлів (за принципом файлового вірусу) і при запуску файлу щораз перевіряють його на зміну. Недолік у таких імунізаторів всього один, але він є летальним: абсолютна нездатність повідомити про зараження стелс-вірусом. Тому такі імунізатори практично не використовуються в даний час. Другий тип імунізації захищає систему від поразки вірусом якогось визначеного виду. Файли на дисках модифікуються таким чином, що вірус приймає них за вже заражені. Для захисту від резидентного вірусу в пам'ять комп'ютера заноситься програма, що імітує копію вірусу. При запуску вірус натикається на неї і вважає, що система вже заражена. Такий тип імунізації не може бути універсальним, оскільки не можна провести імунізацію файлів від усіх відомих вірусів: одні віруси вважають уже зараженими файли, якщо час створення файлу містить мітку 62 секунди, а інші - 60 секунд. Однак незважаючи на це, подібні імунізатори в якості півзаходу можуть цілком надійно захистити комп'ютер від нового невідомого вірусу аж до того моменту, коли він буде визначатися антивірусними сканерами.

Якість антивірусної програми визначається за наступними позиціями, що приведені у порядку убавання їхньої важливості:

1. Надійність і зручність роботи - відсутність «зависань» антивірусу і інших технічних проблем, що вимагають від користувача спеціальної підготовки.
2. Якість виявлення вірусів усіх розповсюджених типів, сканування усередині документів Office, упакованих і архівованих файлів. Відсутність «помилкових спрацьовувань». Можливість лікування заражених об'єктів.
3. Можливість лікування заражених об'єктів.
4. Швидкість роботи.



## 7 ОСНОВИ ВЕБ-ДИЗАЙНУ

### 7.1 Коротка характеристика мови HTML

**HTML-документ** – файл текстової або нетекстової природи (звук, відео, зображення), створений за допомогою *мови гіпертекстової розмітки HTML* (Hyper Text Mark-up Language).

Застосування універсальної мови HTML зумовлене необхідністю представлення інформації для глобального використання комп'ютерами різноманітних типів, що працюють під управлінням різноманітних операційних систем. Існують стандарти мови HTML, які підтримуються практично всіма провідними компаніями-розробниками програмного та апаратного забезпечення. Ці стандарти передбачають, що HTML-документ буде однаково та коректно представлений будь-якому користувачеві мережі WWW.

Мова HTML надає розробникам засоби для:

- Публікації HTML-документів з заголовками, текстом, таблицями, списками, рисунками і т.і.
- Навігації по HTML-документам за допомогою гіперпосилань.
- Розробки форм для обміну даними з відділеними службами, що можуть бути використані при пошуку інформації, в електронній комерції і т.і.
- Включення електронних таблиць, відео та аудіо кліпів та інших додатків безпосередньо в HTML-документи.

Текстовий HTML-документ представляє собою звичайний текстовий файл з розширенням HTML або HTM. Відзначимо, що в назві файлу можливо використовувати тільки символи латинського алфавіту, арабські цифри, символи дефісу ("-") та підкреслення ("\_").

В цьому документі може міститись інформація двох типів: звичайний текст та команди (*теги*) мови HTML. Зміст HTML-документу інтерпретуються браузером під час завантаження. Як правило, наслідком інтерпретації тексту є його відображення у вікні браузера. Результати інтерпретації тегів залежать від їх призначення. Наприклад, це може бути форматування тексту або відображення у вікні браузера певного рисунку.

Всі теги мови HTML-документу виділяються обмежувачими символами (< та >), між якими записується *ідентифікатор (ім'я) тегу*. Розрізняють *одинарні* та *парні теги*.

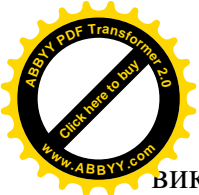
Наведемо приклад одинарного тегу для відображення у вікні браузера горизонтальної лінії <hr>.

На відміну від одинарного для визначення парного тегу (*тегу-контейнеру*) використовуються *відкриваючий* та *закриваючий тег*. Закриваючий тег відрізняється наявністю символу / перед іменем тегу.

Наприклад:

```
<td> Зміст тегу </td>
```

Все, що записано між відкриваючим та закриваючим тегом називають *змістом тегу*. В деяких випадках закриваючий тег можливо не



використовувати. Наприклад тег–контейнер `<li>` можливо використовувати без закриваючого тегу `</li>`.

Теги можуть мати **параметри (атрибути)**. Наприклад, для визначення розміру тексту в тезі `<font>` можливо використати атрибут `size`:

```
<font size='12'>
```

Кількість та номенклатура параметрів індивідуальні для кожного тегу. Якщо параметрів декілька, то вони відокремлюються між собою за допомогою пробілів. Наприклад:

```
<font size=12 color='red'>
```

Відзначимо, що як імена тегів, так і імена параметрів не чутливі до регістру символів, за допомогою яких вони визначені. Таким чином, визначення тегів `<br>` та `<BR>` рівнозначне. Розрізняють параметри тегів, які потребують запису значення та параметри, які використовуються без значення.

Якщо параметр тегу не потребує запису значення, то використовується тільки назва параметру. Наприклад:

```
<br noshade>
```

Якщо параметр тегу потребує запису значення, то воно вказується після назви параметру після символу `"=`". Значення параметру може записуватись як в одинарних або подвійних лапках, так і взагалі без лапок. Наприклад:

```
<td align='left'>
```

```
<td align="left">
```

```
<td align=left>
```

Проте рекомендується записувати значення параметру в одинарних або подвійних лапках.

Крім одинарних та парних тегів розрізняють **теги рівня блоку** та **послідовні теги**. Різниця між ними полягає в тому, що теги рівня блоку можуть містити послідовні теги та інші теги рівня блоку. Послідовні теги можуть містити тільки данні та інші послідовні теги. Наведемо приклади:

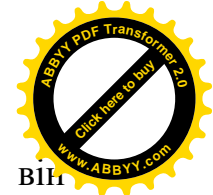
- Теги рівня блоку: `<table></table>`, `<p></p>`.
- Послідовні теги: `<img>`, `<br>`.

Важливою особливістю мови HTML є те, що неправильно записані теги або параметри тегів ігноруються браузером. При цьому браузер не показує ніякого повідомлення та переходить до інтерпретації інших частин HTML-документу.

Мова HTML передбачає можливість використання кольорового форматування HTML-документу. При цьому кольори визначаються або за допомогою англійських назв, або в форматі **RGB** (*Red Green Blue*).

Приклади визначення кольорів за допомогою англійських назв:

- *green* – зелений.
- *red* – червоний.
- *blue* – синій.
- *white* – білий
- *black* – чорний.



Хоча такий спосіб визначення кольору досить зручний, але він обмежений кількістю відповідних назв. Тому більш поширений є спосіб визначення кольорів у форматі RGB, який дозволяє використовувати більш ніж 1,5 млн. кольорів.

Передумовою використання формату RGB є те, що будь-який колір можливо визначити за допомогою трьох кольорів: червоного (red), зеленого (green) та синього (blue). В форматі RGB кожному із цих кольорів може відповідати число від 0 до 256, записане в шістнадцятичному форматі. Відсутність даного кольору відповідає числу 0 (00), а максимальна насиченість числу 256 (ff). Синтаксис RGB передбачає використання знаку "#".

Приклади визначення кольорів в форматі RGB:

- Червоний – #ff0000.
- Зелений – #00ff00.
- Синій – #0000ff.
- Чорний – #000000.
- Білий – #ffffff.

Для створення HTML-документу достатньо використовувати текстовий редактор, який не вставляє в текст документу власних команд. Прикладом може бути редактор "Блокнот", що входить в стандартну комплектацію операційної системи Windows.

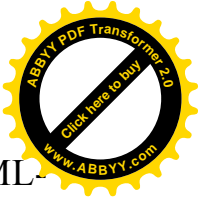
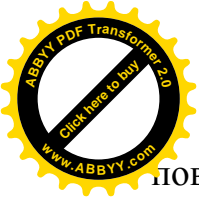
Технологія створення HTML-документу в випадку застосування "Блокноту" полягає в тому, що Web-програміст визначає текст HTML-документу та теги, що в ньому використовуються. Після цього HTML-документ зберігається у файлі формату html або htm та тестується за допомогою браузеру. Після виправлення помилок HTML-документ може бути розміщений на Web-сервері.

Хоча вказана технологія є досить простою та не потребує спеціального програмного забезпечення, але вона є не оптимальною з точки зору трудомісткості, виявлення помилок при визначенні тегів та високих вимог до кваліфікації Web-програміста. Високі вимоги пояснюються тим, що Web-програмісту необхідно пам'ятати великий обсяг інформації, пов'язаний з номенклатурою та синтаксисом тегів. Тому на сучасному етапі використовують спеціальні HTML-редактори, які дозволяють суттєво підвищити якість програмного коду та прискорити процес створення HTML-документів. Найбільш популярними серед вітчизняних Web-програмістів є HTML-редактори – Macromedia Homesite та Macromedia Dreamweaver. Популярність цих редакторів пояснюється дружнім інтерфейсом користувача, наявністю вбудованої в редактори бібліотеки тегів та синтаксичного аналізатора правильності програмного коду HTML-документу, а також можливістю застосування візуальної технології створення HTML-документів.

## 7.2 Визначення загальних параметрів

Першим тегом, з якого слід починати опис HTML-документу, є тег <html>. Він завжди повинен починати опис документу, а завершати опис





повинен тег `</html>`. Тобто тег `<html>` є тегом контейнером для всієї HTML-сторінки. Сам документ є звичайним текстовим ASCII-файлом і може складатися із двох розділів – заголовка та змістовної частини.

Розділ заголовка відкривається тегом `<head>` і закривається тегом `</head>`. Задачею заголовку є надання браузеру інформації, необхідної для коректного відображення HTML-сторінки. Теги, що знаходяться всередині розділу заголовку (крім назви документу), на екрані не відображаються. Назву документу визначають за допомогою тегу контейнеру `<title>`. Наприклад:

```
<title> Назва HTML-сторінки </title>
```

Змістова частина документу, що не містить фреймів, визначається тегом контейнером `<body>`. Тобто весь текст та всі теги змістовної частини документу знаходиться між тегами `<body>` та `</body>`.

Таким чином, для визначення простої HTML-сторінки досить написати:

```
<html><head><title> Заголовок моєї Web-сторінки</title>
</head><body>
  Зміст Web-сторінки
</body></html>
```

Відображення цієї сторінки в вікні браузера показано на рис. 7.1.

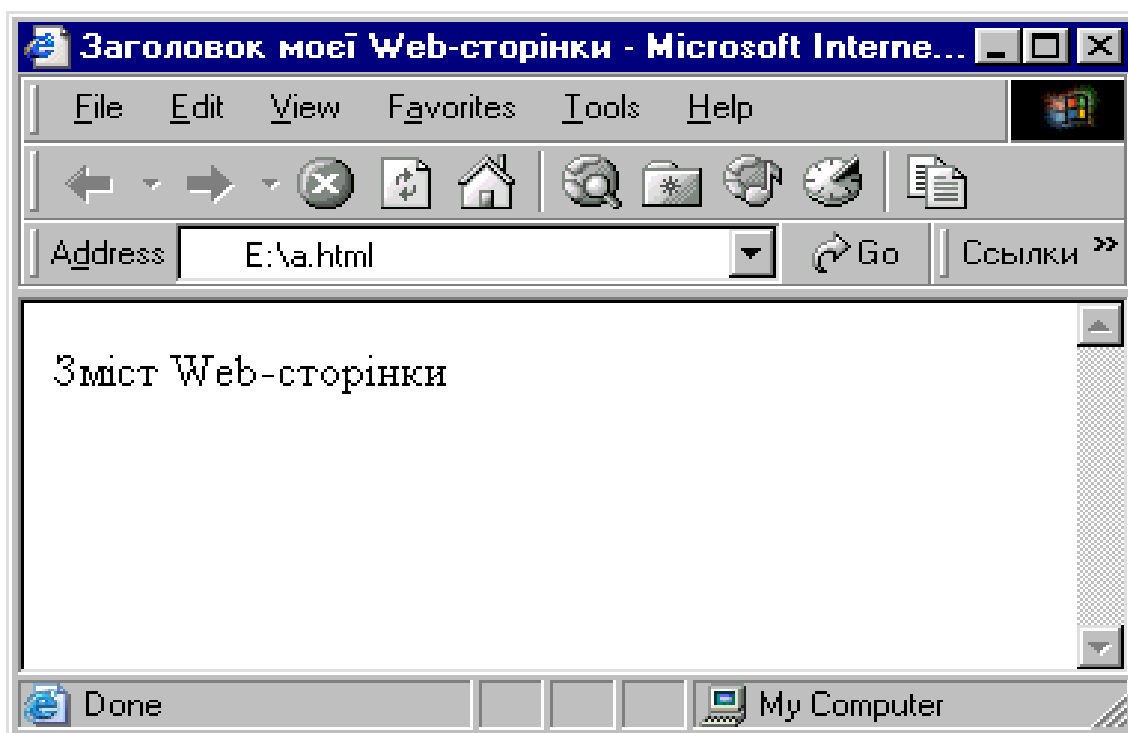
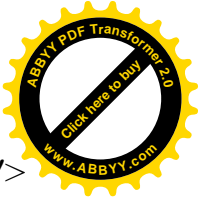


Рис.7.1 - Відображення простої HTML-сторінки браузером

Відзначимо, що в розділі заголовку досить часто використовується тег `<meta>`. За допомогою цього тегу вирішують деякі типові задачі:

1. Визначення кодування тексту HTML-сторінки. Наприклад, визначення того, що кодування тексту HTML-сторінки, задане в найпоширенішому серед вітчизняних користувачів форматі windows-1251, можливо здійснити так:





`<META http-equiv=Content-Type content="text/html; charset=windows-1251">`

2. Визначення ключових слів, що використовуються пошуковими системами. Наприклад, визначення ключових слів "економіка" та "торгівля" можливо здійснити так:

`<META name="keywords" content="економіка, торгівля">`

3. Визначення опису HTML-сторінки в пошуковій системі. Наприклад:

`<META name="description" content = "сайт, присвячений економіці та торгівлі ">`

4. Відображення в браузері визначеної HTML-сторінки. Наприклад, для переходу на HTML-сторінку 1.html через 20 секунд після відображення поточної HTML-сторінки необхідно:

`<meta http-equiv="refresh" content="20;URL=untitled1.html">`

Також необхідно відзначити, що тег `<body>` може мати ряд необов'язкових параметрів (див. табл.2.1), які визначають форматування всієї HTML-сторінки.

Таблиця 7.1- Параметри тегу `<body>`

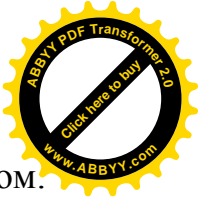
Параметр	Призначення
alink	Визначає колір активного гіперпосилання
background	Вказує на адресу фонового зображення
bottommargin	Визначає межу нижнього поля документу
bgcolor	Визначає колір фону документу
bgproperties	Дозволяє або забороняє прокрутку фонового зображення
leftmargin	Визначає межу лівого поля документу
link	Визначає колір не переглянутого гіперпосилання
rightmargin	Визначає межу правого поля документу
scroll	Встановлює або забороняє смуги прокрутки вікна браузера
text	Визначає колір тексту
topmargin	Визначає межу верхнього поля документу
vlink	Визначає колір переглянутого гіперпосилання

Наприклад, для визначення документу з текстом зеленого кольору та відсутністю смуг прокрутки слід написати:

`<body text="green" scroll="0">`

### 7.3 Форматування тексту

Форматування тексту засобами HTML реалізується за допомогою спеціальних тегів. Всі теги форматування тексту використовуються тільки в розділі `<body>` і є парними тегами. Найчастіше використовуються такі теги:



Тег `<b>` – призначений для відображення тексту напівжирним шрифтом.

Наприклад:

`<b>Це напівжирний шрифт</b>`

Тег `<i>` – призначений для відображення тексту курсивом.

Наприклад:

`<i>Це курсив</i>`

Тег `<u>` – призначений для відображення тексту підкресленим.

Наприклад:

`<u>Це підкреслений текст</u>`

Тег `<s>` – призначений для відображення тексту перекресленими символами.

Наприклад:

`<s>Це перекреслені символи</s>`

Тег `<sub>` – призначений для відображення тексту в вигляді верхнього індексу.

Наприклад:

`<sub>Верхній індекс </sub>`

Тег `<sup>` – призначений для відображення тексту в вигляді нижнього індексу.

Наприклад:

`<sup>Нижній індекс </sup>`

Теги `<sub>` та `<sup>` зручно використовувати для виводу формул.

Всі теги форматування тексту можуть бути вкладені один в інший. Це дозволяє проводити комбінований вивід тексту.

Наприклад, для того, щоб текст був виведений курсивом та був підкресленим необхідно:

`<i><u>Приклад підкресленого курсиву</u></i>`

Порядок вкладення тегів один в інший значення не має.

Наприклад, попередній приклад коректно записати і так:

`<u><i>Приклад підкресленого курсиву</i></u>`

Зміна зовнішнього вигляду тексту в вікні браузера при використанні наведених тегів форматування показана на рис. 7.2.

Тег `<font>` – призначений для визначення параметрів шрифту. За допомогою цього тегу можливо визначити колір символів, їх розмір та тип шрифту. Зміна параметрів шрифту відбувається за допомогою змін параметрів тегу `color`, `size` та `face`.

Параметр **color**. Дозволяє змінювати колір символів. Наприклад, визначення зеленого кольору символів можливо реалізувати так:

`<font color="green">Цей текст зеленого кольору</font>`

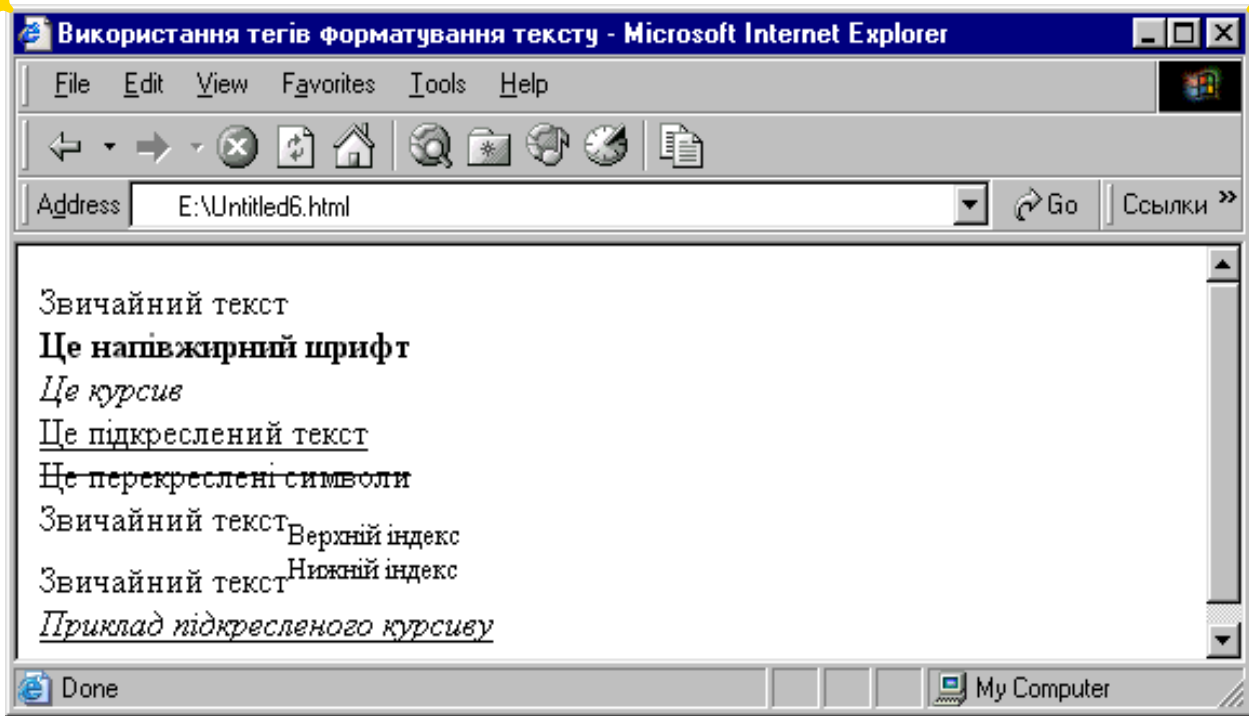
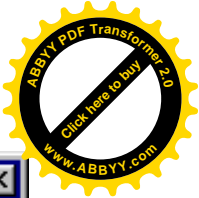


Рис. 7.2 - Використання тегів форматування тексту

Відзначимо, що колір тексту, визначений за допомогою цього тегу, має пріоритет над кольором, визначеним за допомогою тегу `<body>`. Тобто, якщо в тегу `<body text="red">` визначено, що колір тексту червоний, а в тегу `<font color="yellow">` заданий жовтий колір, то на всій HTML-сторінці текст буде червоний, але блок, обмежений тегом `<font>`, буде жовтого кольору.

Параметр **face**. Дозволяє задати тип шрифту, яким браузер буде виводити текст. Для цього в якості значення параметру необхідно вказати назву шрифту. Наприклад, вивід тексту шрифтом Courier New, можливо реалізувати так:

```
<font face="Courier New"> Використовуємо шрифт Courier New</font>
```

Відзначимо, якщо на комп'ютері користувача такого шрифту немає, то браузер буде використовувати стандартний шрифт. Крім того можливо задати не одну, а декілька назв шрифтів, розділених комами. Якщо на комп'ютері користувача не буде першого шрифту, то браузер намагатиметься знайти шрифт за номером два і так далі. Наприклад:

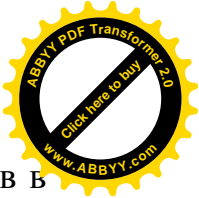
```
<font face="Arial,Helvetica, sans-serif"> Використовуємо шрифт Arial, якщо такого нема, то Helvetica і так далі.</font>
```

Параметр **size**. Дозволяє визначити розмір символів. Розмір задається в умовних одиницях від 1 до 7. Фактичний розмір умовних одиниць залежить від характеристик екрану комп'ютера користувача та настройок браузера. Цифра один відповідає найменшим символам, а 7 – найбільшим. Розмір символів при стандартних установках 3. Розмір символів можливо вказувати як в абсолютних, так і в відносних величинах.

Синтаксис запису наступний:

```
<font size=x>Визначення символів в абсолютних величинах</font>
```

```
<font size=±x>Визначення символів в відносних величинах</font>
```



В першому випадку  $x$  – це цифра від 1 до 7, тобто величина символів в умовних одиницях. В другому випадку  $x$  може бути як додатнім, так і від’ємним числом. Причому знак  $+$  або  $-$  треба вказувати обов’язково. Величина  $x$  буде показувати на скільки умовних одиниць повинна змінитись величина символів відносно поточного значення.

Наприклад, вивід тексту з розміром символів 7 та 1 можливо реалізувати так:

`<font size=7>Визначення символів розміром 7</font>`

`<font size=1>Визначення символів розміром 1</font>`

Зменшення символів на одну умовну одиницю відносно поточного значення:

`<font size=-1> Зменшення символів на 1 умовну одиницю </font>`

Збільшення символів на одну умовну одиницю відносно поточного значення:

`<font size=+1> Збільшення символів на 1 умовну одиницю </font>`

Приклади використання параметрів `face` та `size` теги `<font>` показані на рис. 7.3.

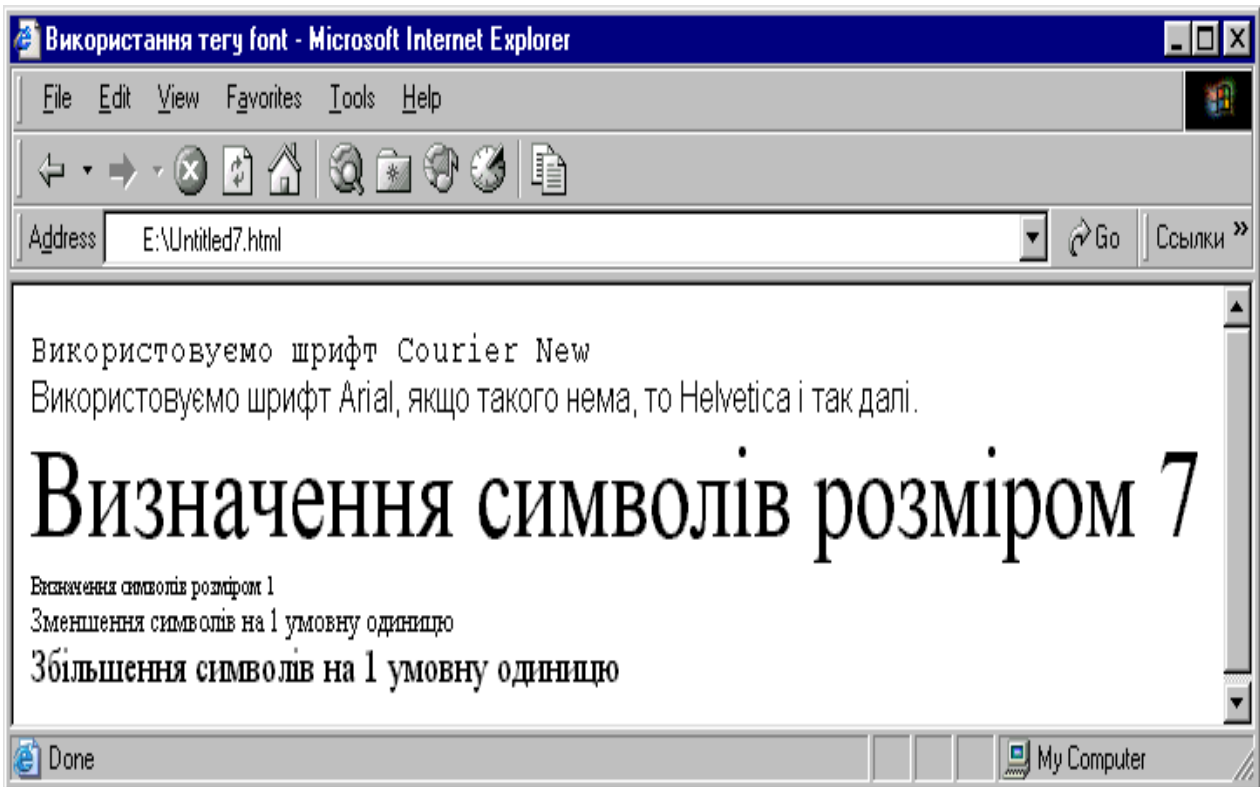
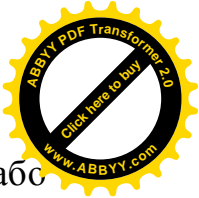
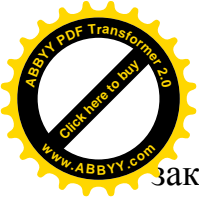


Рис. 7.3 Приклади використання параметрів `face` та `size` теги `<font>`

Тег **<basefont>** використовується для визначення параметрів шрифту, які будуть використовуватись в якості стандартних при перегляді HTML-сторінки. Параметри теги `<basefont>` ті ж самі, що і для теги `<font>`. Зміна параметрів `<basefont>` розповсюджується на ту частину HTML-сторінки, яка розміщена нижче теги `<basefont>`. При сумісному використанні теги `<basefont>` з `<font>` пріоритет має тег `<font>`. На відміну від `<font>` тег `<basefont>` не має



закриваючого тегу. Його дія розповсюджується до кінця HTML-сторінки або до ще одного тегу `<basefont>`. Наприклад, визначення шрифту розміром 4 за допомогою тегу `<basefont>` можна здійснити так:

```
<basefont size=4>Використання тегу basefont
```

Відзначимо, що теги `<font>` та `<basefont>` відносяться до послідовних тегів, а тому не можуть включати в себе теги рівня блоку, наприклад, `<table>` або `<tr>`.

#### 7.4 Використання тегів заголовків

Теги заголовків (`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`) призначені для виділення на екрані користувача певних фрагментів тексту HTML-сторінки. Виділення тексту реалізується за рахунок зміни розмірів та "жирності" тексту. Тегу `h1` відповідає найбільший розмір тексту, а `h6` – найменший розмір. Особливістю даних тегів є вставка порожнього рядка до і після виділеного фрагменту тексту.

Наприклад, для виділення в тексті HTML-сторінки трьох фрагментів тексту як заголовків 1, 3 та 6 рівнів необхідно:

```
<body>
```

```
Звичайний текст. <h1> Заголовок 1 рівня </h1>
```

```
Звичайний текст. <h3> Заголовок 3 рівня </h3>
```

```
Звичайний текст. <h6> Заголовок 6 рівня </h6>
```

```
Звичайний текст.
```

```
</body>
```

Відповідне відображення HTML-сторінки в вікні браузера показано на рис.

7.4.

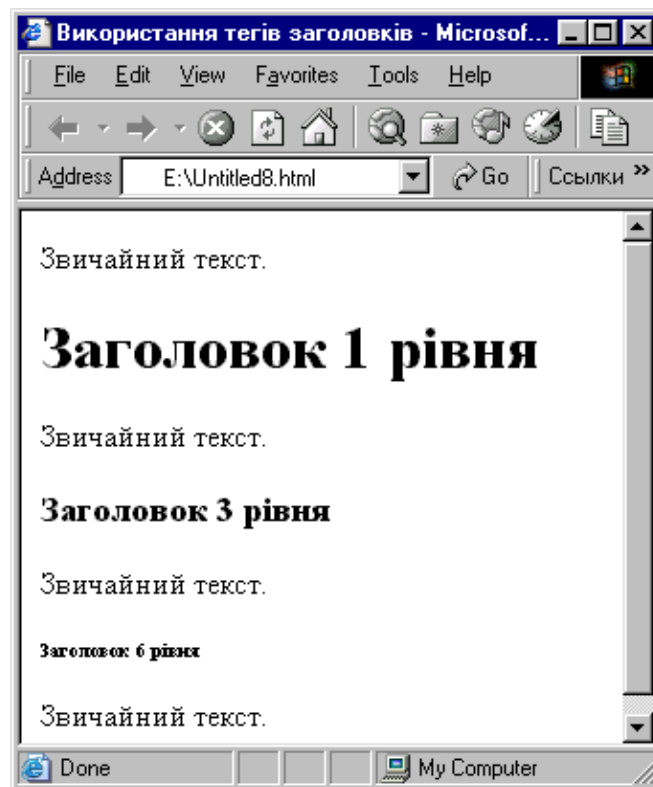


Рис. 7.4 - Використання тегів заголовків



Всі вказані теги є парними та можуть використовуватись з необов'язковим параметром align, який служить для горизонтального вирівнювання тексту. Можливі значення параметру align показані в табл. 7.2.

Таблиця 7.2 - Значення параметру align

Значення параметру	Наслідки використання
left	Текст вирівнюється по лівій стороні вікна браузера
center	Текст вирівнюється по центру вікна браузера
right	Текст вирівнюється по правій стороні вікна браузера
justify	Текст вирівнюється по ширині вікна браузера. Візуально відрізняється від left тільки в тому випадку, коли текст заголовку містить більш ніж два рядки тексту.

Відзначимо, якщо параметр align не заданий, то використовується вирівнювання заголовку по центру вікна браузера.

Розглянемо приклад визначення на HTML-сторінці чотирьох заголовків третього рівня з різними параметрами вирівнювання. Для цього в розділі необхідно записати:

*<body> Звичайний текст.*

*<h3 align="center"> Заголовок вирівнюється по лівій стороні вікна браузера</h3>*

*Звичайний текст.*

*<h3 align="center"> Заголовок вирівнюється по центру вікна браузера</h3>*

*Звичайний текст.*

*<h3 align="right"> Заголовок вирівнюється по правій стороні вікна браузера</h3>*

*Звичайний текст.*

*<h3 align="justify"> Заголовок вирівнюється по ширині вікна браузера.*

*Візуально відрізняється від left тільки в тому випадку, коли текст заголовку містить більш ніж два рядки тексту. </h3>*

*Звичайний текст. </body>*

Відповідне відображення в вікні браузера показане на рис. 7.5.



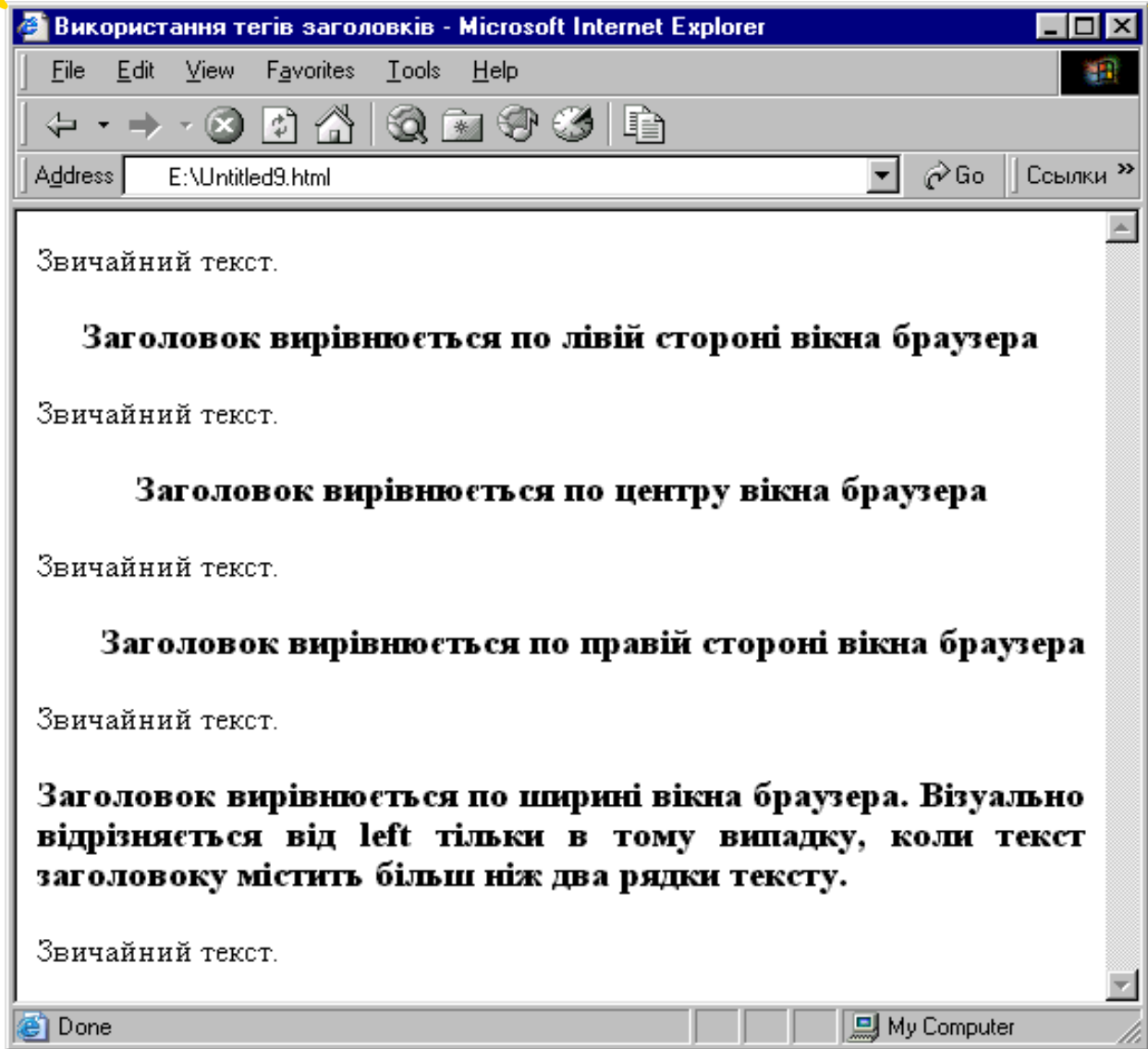
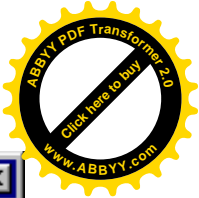


Рис. 7.5 - Використання тегу заголовку третього рівня з різними параметрами вирівнювання.

## 7.5 Форматування HTML-документу

Тег `<br>` використовується для примусового переведення рядка. Необхідність використання даного тегу пояснюється тим, що при відображенні текстових документів перенесення тексту із одного рядка в інший відбувається автоматично та залежить в основному від розмірів шрифту та розмірів вікна браузера. При цьому переведення рядка може здійснюватись тільки по символам, що розділяють окремі слова.

Розглянемо приклад запису виразів "Перший рядок" та "Другий рядок" в двох різних рядках. Відповідний HTML-код такий:

```
<body>
```

```
Перший рядок <br> Другий рядок
```

```
</body>
```

Парний тег `<nobr>` на відміну від тегу `<br>` використовується для заборони переводу рядка. Наприклад, для гарантованого розміщення в одному рядку виразу "Цей текст повинен відображатись в одному рядку" необхідно:

```
<nobr> Цей текст повинен відображатись в одному рядку </nobr>
```

Якщо даний текст не буде вміщуватись в одному рядку, то в вікні браузера з'явиться горизонтальна смуга прокрутки.

Тег `<p>` призначений для розділу тексту HTML-сторінки на окремі абзаци. Даний тег є тегом-контейнером та може використовуватись з необов'язковим параметром `align`. Значення параметру `align` та наслідки його використання для абзацив ті ж самі, що і для заголовків, див. табл. 2.2. Відзначимо, що візуально в вікні браузера абзац буде відділятися від іншого тексту за допомогою порожніх рядків та можливої зміни горизонтального вирівнювання. Наведемо приклад використання тегу `<p>`:

```
<body>
```

*Звичайний текст*

```
<p align="center"> Абзац з горизонтальним вирівнюванням - "по центру" </p> </body>
```

Відповідне вікно браузера показано на рис. 7.6.

Парний тег `<center>` призначений для горизонтального вирівнювання елементів HTML-сторінки посередні вікна браузера. Наведемо приклад його використання:

```
<body>
```

*Звичайний текст*

```
<center> Текст з горизонтальним вирівнюванням - "по центру" </center>
```

```
Звичайний текст </body>
```

Відзначимо, що в специфікаціях діючих стандартів HTML тег `<center>` позначений як не бажаний для використання, хоча і підтримується практично всіма розповсюдженими браузерами.

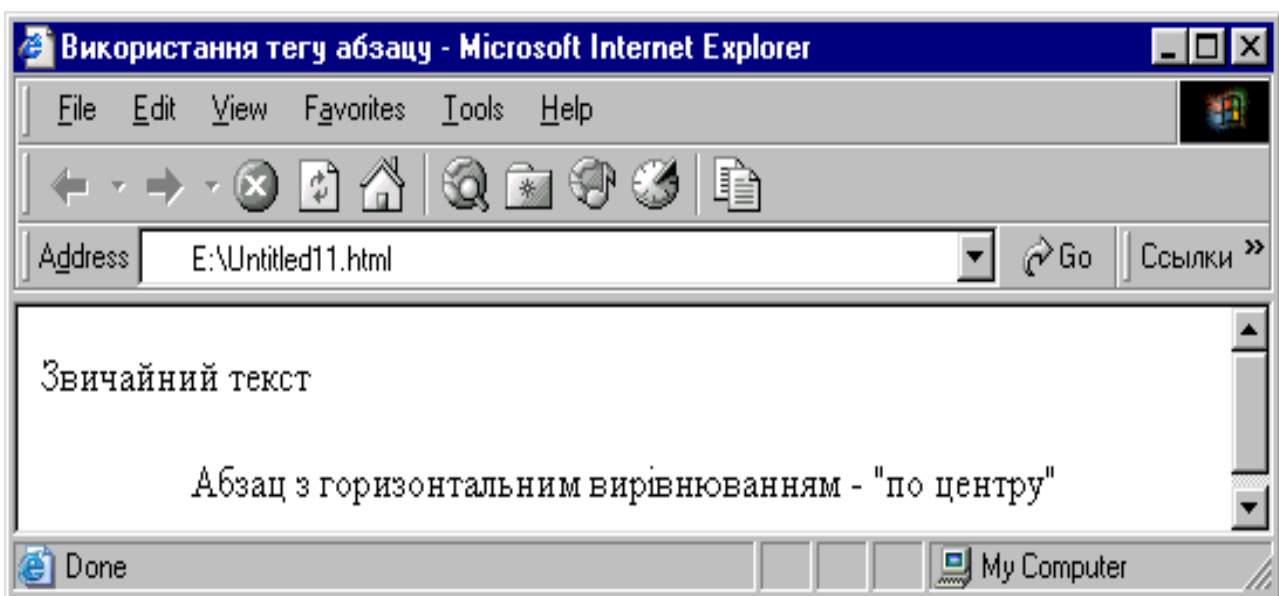


Рис. 7.6 - Використання тегу абзацу



Тег-контейнер `<pre>` використовується для визначення попередньо відформатованого тексту. Текст, розміщений між тегами `<pre>` та `</pre>`, буде виведений у вікні браузера із збереженням деяких параметрів форматування, виконаних за допомогою текстових редакторів. Відзначимо, що в середині контейнеру `<pre>` не можна використовувати деякі важливі та широко вживані теги. Тому використання цього тегу досить обмежене. Тег `<hr>` призначений для виведення в вікні браузера горизонтальної лінії, до і після якої вставляються порожні рядки. Необов'язкові параметри тегу `<hr>` та їх призначення наведені в табл. 7.3.

Таблиця 7.3 - Параметри тегу `<hr>`

Параметр	Призначення
align	Визначення горизонтального вирівнювання лінії. Можливі значення: left ("по лівому краю"), center ("по центру"), right ("по правому краю"). Якщо параметр align не використаний то застосовується вирівнювання "по центру"
color	Встановлює колір лінії
noshed	Відмінняє рельєфність лінії
size	Визначає товщину лінії
width	Встановлює довжину лінії

Наприклад, для визначення лінії червоного кольору, товщиною 2 пікселі, довжиною 400 пікселів, без рельєфу необхідно:

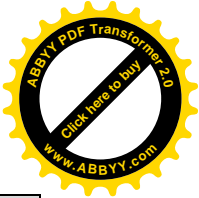
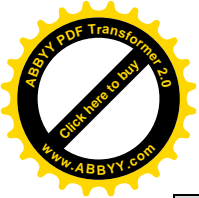
```
<hr color="red" size=2 width=400 noshade>
```

## 7.6 Використання коментарів та спеціальних символів

В деяких випадках необхідно не показувати у вікні браузера частину HTML-коду сторінки, тобто "за коментувати" цей код. Для цього використовуються теги коментарів `<!--` та `-->`. Наприклад:

```
<!-- Цей текст не буде відображатись у вікні браузера -->
```

Як відомо, при створенні HTML-сторінки повинні застосовуватись тільки символи, що входять до базової частини таблиці кодів ASCII. Проте деякі спеціальні символи (наприклад, пробіл) не входять до цієї частини таблиці. В випадку використання вказаних спеціальних символів необхідно застосовувати спеціальні HTML-коди, див. табл. 7.4.



Таблиця 7.4 - Коди спеціальних символів

Спеціальний символ	HTML-код
Амперсанд (&)	&amp;
Знак copyright (©)	&copy;
Знак більше (>)	&gt;
Знак менше (<)	&lt;
Пробіл	&nbsp;
Знак плюс/мінус (±)	&plusmn;
Знак одна друга (½)	&frac12;
Знак одна друга (¼)	&frac14;
Знак одна друга (¾)	&frac34;
Знак "»"	&raquo;
Знак "§"	&sect;
Знак "«"	&laquo;

## 7.7 Рисунки

Для вставки рисунка в HTML-сторінку використовується тег `<img>`. Основним параметром цього тегу є параметр *src*, за допомогою якого вказується адреса файлу, в якому і зберігається рисунок. Цей же тег також можливо використовувати для показу відео-роликів. В цьому випадку використовується параметр *dynsrc*, за допомогою якого вказується адреса відео файлу. Найпоширенішими форматами файлів-рисунків є формати *jpg*, *gif*, *png*, а відео файлів - *avi*.

Відзначимо, що адреса файлу-рисунок може бути вказана в абсолютному та відносному форматі. При використанні абсолютного формату в адресі повністю визначено комп'ютер, каталог та файл рисунок.

Наприклад, для вставки рисунка, що зберігається в файлі *logo.gif* за адресою [www.picture.com.ua/picture](http://www.picture.com.ua/picture) необхідно:

```

```

При використанні відносного формату визначення місцезнаходження файлу-рисунок відбувається з урахуванням місцезнаходження даної HTML-сторінки.

Наприклад, для вставки в HTML-сторінку *prim.html* рисунка, який зберігається в файлі *logo.gif*, що знаходиться в одному каталозі з файлом *prim.html* необхідно:

```

```

Графічні файли зручно зберігати в окремому каталозі, наприклад, в каталозі з іменем *img*. В цьому випадку для вставки в HTML-сторінку *prim.html* рисунка *logo.gif* необхідно:

```

```

Відзначимо, що каталог *img* та файл *prim.html* повинні знаходитись в одному каталозі. Крім обов'язкового параметру *src* тег `<img>` має декілька необов'язкових параметрів, див. табл. 7.5.

Таблиця 7.5 - Параметри тегу <img>

Назва параметру	Призначення
alt	Визначення альтернативного тексту
width	Ширину рисунку в пікселях
height	Висоту рисунку в пікселях
hspace	Відступ від рисунку до інших об'єктів на HTML-сторінці по вертикалі
vspace	Відступ від рисунку до інших об'єктів на HTML-сторінці по горизонталі
border	Товщина рамки навколо рисунку
lowsrc	Задає файл з альтернативним зображенням
align	Вирівнювання рисунку відносно інших об'єктів на HTML-сторінці
src	Адреса графічного файлу
dynsrc	Адреса відео файлу
start	Задає момент початку прокрутки відео ролика. Можливі значення: fileopen – ролик починається відразу після завантаження відео файлу, mouseover – ролик починається при наведенні миші на зображення.
loop	Задає кількість повторів відео - ролика. Якщо значення дорівнює -1, то програвання відбувається безперервно

Розглянемо призначення вказаних параметрів докладніше. Досить часто при перегляді HTML-сторінок користувачі використовують браузер, що працює в режимі відключення завантаження зображень. Як відомо, цей режим дозволяє зменшити час відображення HTML-сторінки в вікні браузера. В цьому випадку замість зображення на екрані з'являється альтернативний текст, заданий в параметрі **alt**. Крім цього альтернативний текст відображається в якості підказки при знаходженні курсору "миші" в області рисунку. Приклад використання параметру *alt*:

```

```

Параметри *width* та *height* призначені для визначення розмірів рисунку у вікні браузера. Відзначимо, що при зміні розмірів зображення розмір файлу не змінюється, тобто час завантаження рисунку не змінюється. Наведемо приклад застосування параметрів *alt*, *width* та *height* для одного і того ж рисунку:

```

```

```

```

``

Відповідне вікно браузера показано на рис. 7.7.

Слід зазначити, що зміна розмірів рисунку може призвести до втрати якості зображення, особливо за рахунок втрати пропорцій між початковою висотою та шириною. Практичний досвід показує, що навіть при дотриманні пропорцій збільшення початкових розмірів зображень-фотографій більш ніж на 10-15% призводить до чуттєвої втрати якості. При цьому зменшення розмірів зображень позначається на якості їх відображення набагато менше, але їх використання не оптимальне з точки зору часу завантаження.

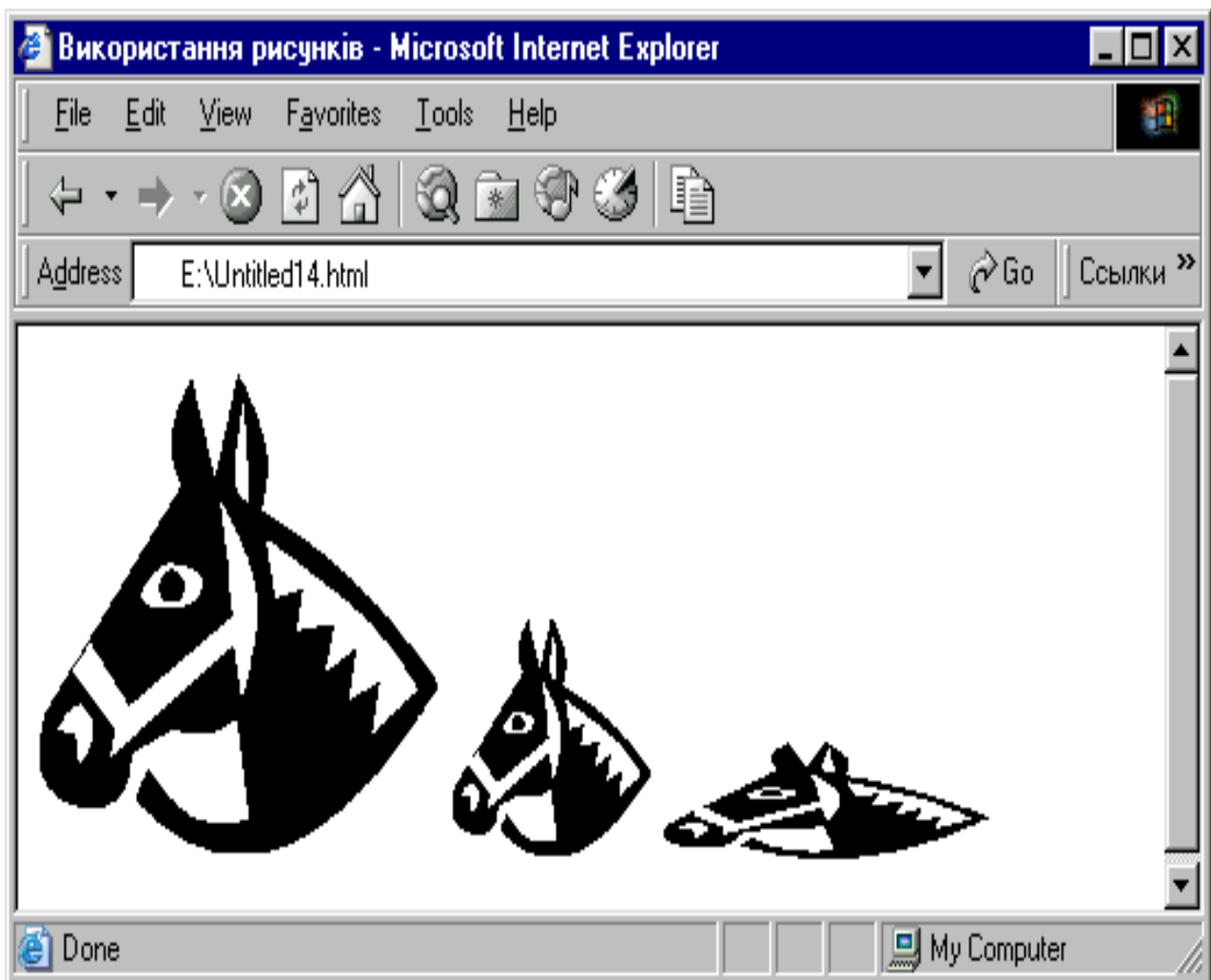


Рис. 7.7 - Використання параметрів alt, width та height тегу <img>

Параметр **border** дозволяє визначити товщину рамки навколо рисунку. Застосуємо цей параметр із значенням 2 для рисунків попереднього прикладу:

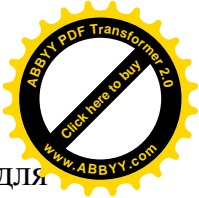
``

``

``.

Параметри **hspace** та **vspace** дозволяють в пікселях задавати відступи від рисунку до інших об'єктів на HTML-сторінці.





Покажемо застосування цих параметрів із значеннями 20 та 20 для попереднього прикладу:

```

```

```

```

```

```

Параметр **lowsrc** дозволяє визначити файл з альтернативним зображенням, яке з'являється у вікні браузера до появи основного зображення. Альтернативне зображення, як правило, це рисунок того ж змісту, що і основний, але менш якісний. По цій причині файл альтернативного зображення менший за розміром, а тому завантажується набагато швидше ніж основний. Це дозволяє швидко показати загальні риси зображення.

Приклад використання:

```

```

Параметр **align** використовується для визначення горизонтального або вертикального вирівнювання рисунку відносно інших об'єктів на HTML-сторінці. Параметри вирівнювання залежать від значень параметру align. При використанні цього параметру із значеннями **left** або **right** рисунок розміщується відповідно до лівої або правої сторони вікна браузера, а текст "обтікає" зображення з протилежної сторони. Відзначимо, що в цьому випадку текст може займати декілька рядків.

Приклад використання параметру align:

```
<body>
```

```

```

*Приклад використання параметру align="right". Рисунок розміщений біля правої сторони вікна браузера. Текст обтікає зображення з лівої сторони.*

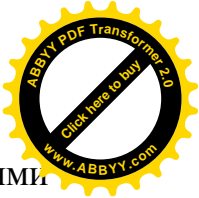
```
</body>
```

Крім **left** та **right** параметр **aling** може приймати наступні значення:

- **top** – верхня межа зображення вирівнюється по найбільш високому елементу рядка;
- **texttop** – верхня межа зображення вирівнюється по найбільш високому текстовому елементу рядка;
- **middle** – середина зображення вирівнюється по базовій лінії рядка;
- **absmiddle** – середина зображення вирівнюється по середині рядка;
- **bottom** – нижня межа зображення вирівнюється по базовій лінії рядка;
- **absbottom** – нижня межа зображення вирівнюється по нижній межі рядка.

В цих випадках зображення можна розглядати як звичайний елемент рядка.

Відзначимо, що базова лінія тексту – це нижня лінія рядка тексту без врахування нижньої частини деяких символів (p, j, y).



Наведемо приклад використання параметру align тегу <img> з значеннями top, middle та bottom:

```
<body>
```

```

```

Приклад використання параметру align="top". <br>

```

```

```
</body>
```

### 7.8 Звук

Для програвання звуку на Web-сторінках можливо використовувати тег <embed>, основним параметром якого є *src*, що дозволяє визначити адресу звукового файлу. Наприклад, для того, щоб при перегляді в браузері Web-сторінки прозвучав звук, записаний в файлі *1.wav*, необхідно записати такий HTML-код: <embed src=1.wav >. Основні параметри цього тегу показані в табл. 7.6.

Таблиця 7.6 - Парметри тегу <embed>

Назва параметру	Призначення
width	Ширину звукового програвача в пікселях
height	Висоту звукового програвача в пікселях
hspace	Відступ від звукового програвача до інших об'єктів на HTML-сторінці по вертикалі
vspace	Відступ від звукового програвача до інших об'єктів на HTML-сторінці по горизонталі
hidden	Дозволяє показувати/не показувати (false/true) звуковий програвач у вікні браузера
lowsrc	Задає файл з альтернативним зображенням
align	Вирівнювання рисунку відносно інших об'єктів на HTML-сторінці
src	Адреса графічного файлу
dynsrc	Адреса відео файлу
autostart	Задає момент початку прокрутки відео ролика. Можливі значення: true – програвання починається відразу після завантаження звукового файлу, false – програвання починається після команди користувача.
loop	Задає кількість повторів відео - ролика. Можливі значення: true – програвання відбувається безперервно, false – кількість повторів визначається користувачем.



## 7.9 Гіперпосилання

Як відомо, гіперпосилання призначені для зв'язування одного Web-ресурсу з іншим. Гіперпосилання складається із двох частин. Перша частина – це об'єкт, який користувач бачить у вікні браузера, і вибір якого призводить до переходу на цільовий Web-ресурс, URL-адреса якого вказана в другій частині гіперпосилання. В якості цього об'єкту можуть бути текст або/та зображення. В залежності від цього класифікують текстові та графічні гіперпосилання. При типових настройках браузера гіперпосилання виділяються на HTML-сторінці за допомогою кольору. Крім того текст гіперпосилання виділений за допомогою підкреслення. Для визначення гіперпосилання в коді HTML-сторінки використовують парний тег `<a>`. Основним параметром цього тегу є параметр *href*, що задає URL-адресу гіперпосилання. Синтаксис запису гіперпосилання такий:

`<a href="URL-адресу"> Текстовий або графічний об'єкт </a>`

Приклади запису текстового та графічного гіперпосилання показані нижче:

`<a href="www.meta.ua"> Тексто́ве гіперпосилання </a>`

`<a href="www.meta.ua">  </a>`

Вказана URL-адреса може бути відносна або абсолютна. При використанні відносної адреси повний шлях до цільового ресурсу не заданий. В цьому випадку визначення місцезнаходження цільового ресурсу відбувається з урахуванням місцезнаходження HTML-сторінки, в якій використане гіперпосилання. Наприклад, якщо в файлі 1.html визначене гіперпосилання:

`<a href="2.html"> Текстовий або графічний об'єкт </a>`

то мається на увазі, що цільова HTML-сторінка 2.html знаходиться в тому ж каталозі того ж комп'ютера, в якому знаходиться і HTML-сторінка 1.html.

Абсолютною називається URL-адреса, в якій повністю визначено комп'ютер, каталог та файл цільового ресурсу. Наприклад:

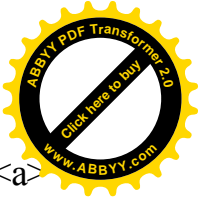
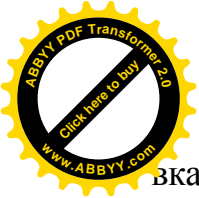
`<a href="www.example.com/html/3.html"> Абсолютне гіперпосилання </a>`

Важливим параметром тегу `<a>` є параметр *target*, за допомогою якого можливо вказати вікно браузеру, в яке повинен бути завантажений цільовий ресурс. Для завантаження цільового ресурсу в нове вікно браузера значення цього параметру має бути `"_blank"`. Наприклад, для визначення текстового гіперпосилання, вибір якого призводить до відображення в новому вікні браузера файлу 2.html, необхідно:

`<a href="2.html" target="_blank"> Текст </a>`

Відзначимо, що при відсутності параметру *target* цільовий ресурс буде завантажуватись в вікно браузера, в якому відображається HTML-сторінка з гіперпосиланням.

Цікавою є можливість гіперпосилань задати в якості цільового ресурсу певну частину власної HTML-сторінки. Наприклад, зручно читати великий документ, коли він має зміст з гіперпосиланнями на різні розділи. В цьому випадку перехід по гіперпосиланню до деякого розділу (який може знаходитись в кінці документу) призведе до відображення цього розділу на екрані. Такі гіперпосилання називаються внутрішніми. Для їх побудови необхідно за допомогою тегу `<a>`



вказати місце переходу (покажчик) та за допомогою параметру *name* тега `<a>` присвоїти йому ім'я.

Наприклад, для переходу на 12 розділ документу на початку тексту цього розділу розмістимо покажчик з іменем "go12":

```
<a name="go12"></a>Текст розділу 12
```

Відзначимо, що в цьому випадку параметр `href` не використовується, а ім'я покажчика повинно бути записане за допомогою літер англійського алфавіту та цифр.

Особливістю гіперпосилання для переходу в середині HTML-сторінки є використання в параметрі `href` імені покажчика з перфіксом `#`. Наприклад:

```
<a href="#go12">Перехід до розділу 12</a>
```

## 7.10 Списки

Списки – широко розповсюджена форма показу даних як в електронних так і в друкованих документах. Мовою HTML передбачено використання трьох стандартних видів списків: маркірованого, нумерованого та списку визначень. Відзначимо, що від інших елементів на HTML-сторінці стандартні списки відділяються порожніми рядками.

### 7.10.1 Маркіровані списки

Для визначення маркірованого списку використовується тег-контейнер `<ul>`, в середині якого розміщуються всі елементи списку. При цьому кожен пункт списку повинен починатись тегом `<li>`. Наведемо приклад HTML-коду маркірованого списку:

```
<body> Звичайний текст
<ul>
<li> Перший пункт маркірованого списку
<li> Другий пункт маркірованого списку
<li> Третій пункт маркірованого списку
</ul>
Звичайний текст
</body>
```

Відображення маркірованого списку у вікні браузера показано на рис. 7.8.

Як в тезі `<ul>`, так і в тезі `<li>` можливо використовувати необов'язковий параметр *type*, за допомогою якого визначається тип маркера списку. Можливі значення цього параметру:

- **disc** – маркери відображаються заповненими колами;
- **circle** – маркери відображаються не заповненими колами;
- **square** – маркери відображаються заповненими квадратами.

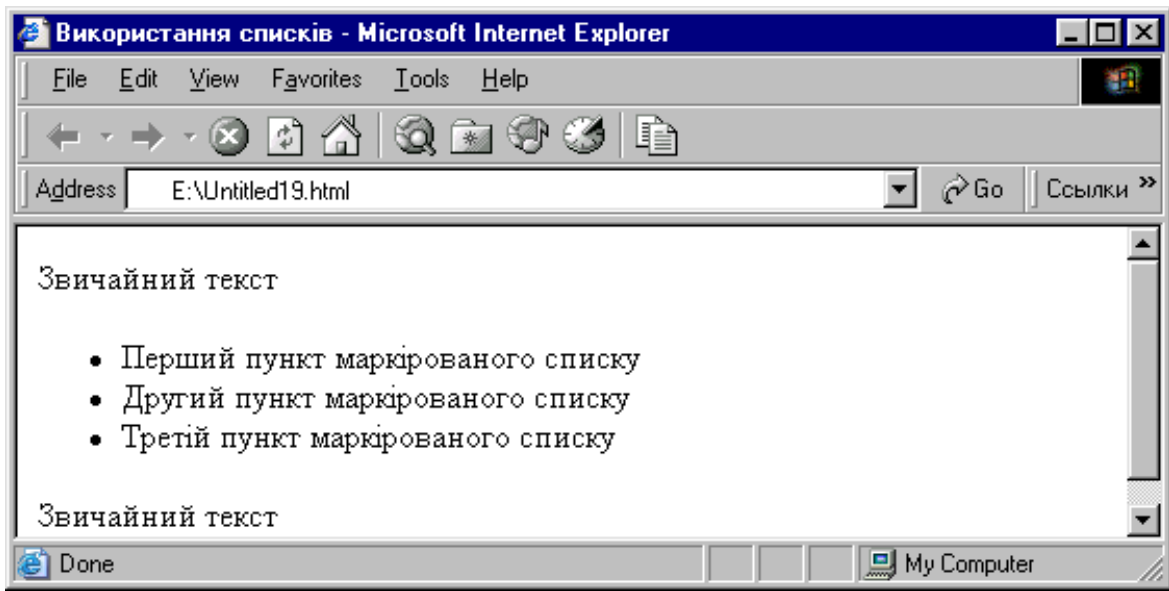


Рис. 7.8 - Відображення маркірованого списку

Приклад використання різних значень параметру `type` в тезі `<li>` маркірованого списку:

```
<ul>
<li type="disc"> Перший пункт маркірованого списку. type="disc"
<li type="square"> Другий пункт маркірованого списку. type="square"
<li type="circle"> Третій пункт маркірованого списку. type="circle"
</ul>
```

Відповідне відображення у вікні браузера показано на рис. 7.9.

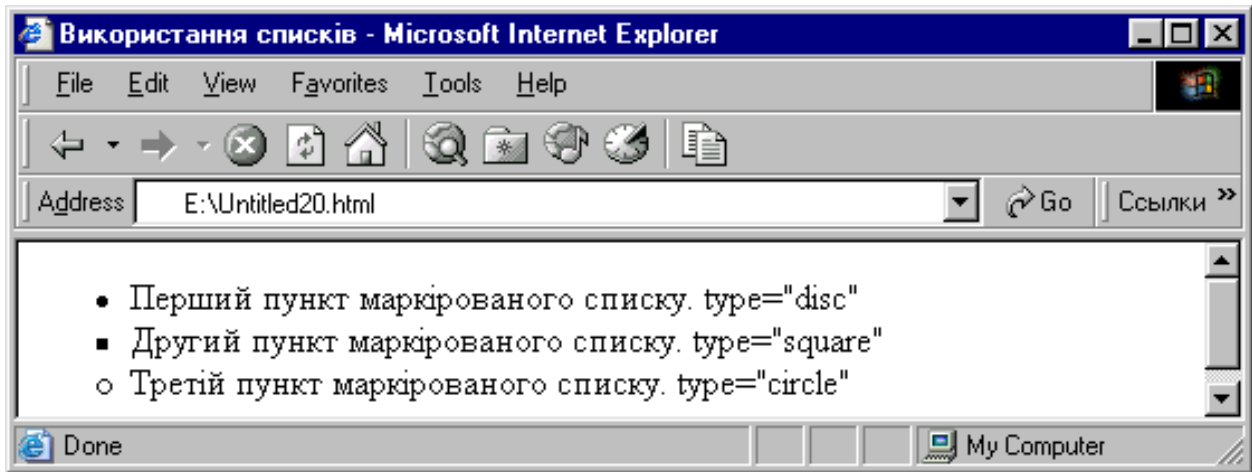


Рис. 7.9 - Використання різних значень параметру `type` в тезі `<li>`

Якщо параметр `type` в тегах `<ul>` та `<li>` не використовується, то для всього списку використовується маркер типу "disc".

## 7.10.2 Нумеровані списки

Для визначення нумерованого списку використовується тег-контейнер `<ol>`, в середині якого розміщуються всі елементи списку. При цьому кожен пункт списку повинен починатись тегом `<li>`. Наведемо приклад HTML-коду нумерованого списку:

```
<ol>
<li> Перший пункт нумерованого списку
<li> Другий пункт нумерованого списку
<li> Третій пункт нумерованого списку
</ol>
```

Відображення такого нумерованого списку у вікні браузера показано на рис. 7.10.

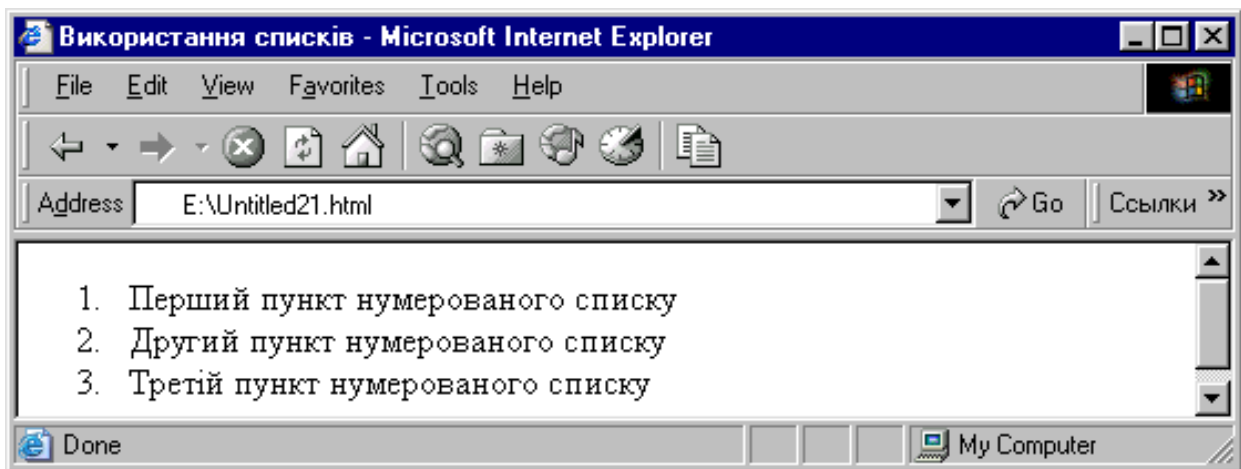


Рис. 7.10 - Відображення нумерованого списку

Досить часто тег `<ol>` використовується з необов'язковими параметрами *type* та *start*. При цьому параметр *type* використовується також в тезі `<li>`. Його призначенням є визначення одного з стандартних типів маркера. Відзначимо, що для нумерованого списку існує п'ять стандартних типів маркерів. Використання цього параметру в тезі `<ol>` розповсюджується на весь список, а використання в тезі `<li>` розповсюджується тільки на поточний пункт списку.

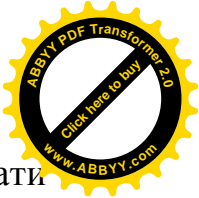
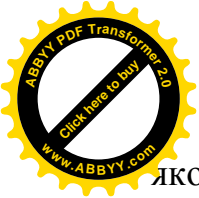
Відзначимо, що нумерація пунктів нумерованого списку не залежить від типу маркеру. Якщо параметр *type* в тегах `<ol>` та `<li>` не зазначено, то для всього списку використовується маркер типу "1".

Параметр *start* призначений для зміни початку нумерації пунктів списку і може використовуватись тільки в тезі `<ol>`. Для зміни нумерації пунктів в середині списку в тезі `<li>` використовується параметр *value*. Значеннями параметрів *start* та *value* можуть бути тільки натуральні числа.

## 7.10.3 Списки визначень

Список визначень використовується для розміщення на HTML-сторінці тексту, за змістом подібного до енциклопедії або словника. Тобто тексту, в





якому потрібно навести термін та пояснення цього терміну, що може займати декілька рядків. На відміну від інших типів списків кожен пункт списку визначень складається із двох частин.

В першій частині пункту записується термін, що потребує визначення.

В другій частині пункту записується текст, що пояснює зміст терміну.

Список визначень задається за допомогою тегу-контейнеру `<dl>`. Для визначення терміну використовується тег `<dt>`, а для визначення пояснення використовується тег `<dd>`.

Наведемо приклад списку визначень для розміщення на HTML-сторінці деяких стандартних термін в галузі Web-програмування:

`<dl>`

`<dt>`*Інтернет (мережа Інтернет)*

`<dd>`*Сукупність мереж та обчислювальних засобів, які використовують стек протоколів TCP/IP (Transport Control Protocol/Internet Protocol), спільний простір імен та адрес для забезпечення доступу до інформаційних ресурсів мережі будь-якій особі;*

`<dt>`*Гіпертекстове посилання*

`<dd>`*Адреса іншого мережевого інформаційного ресурсу у форматі URL, який тематично, логічно або будь-яким іншим способом пов'язаний з документом, у якому це посилання визначене.*

`</dl>`

Відображення цього списку визначень у вікні браузера показано на рис.

7.11.

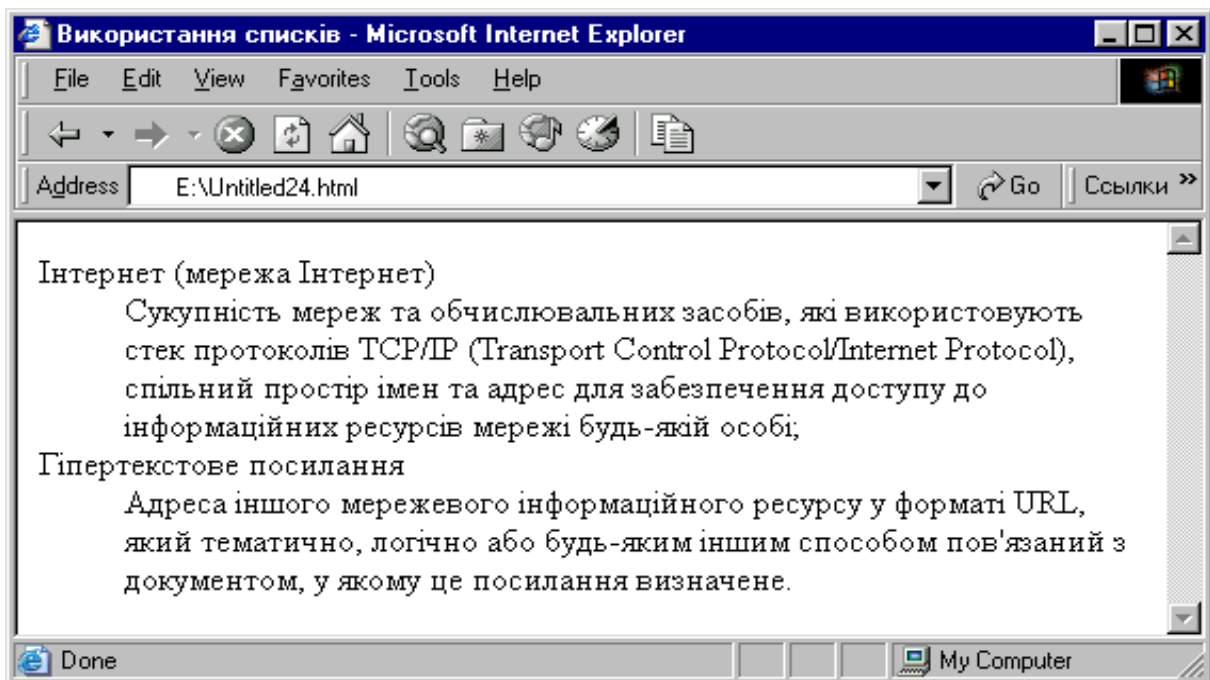


Рис. 7.11 - Список визначень



## 7.11 Таблиці

Особливість використання таблиць полягає в тому, що вони є основним засобом форматування HTML-документів. Розміщення інформації в HTML-документі, виконане за допомогою інших засобів (тегів), має або набагато менші можливості, або не однаково відображається браузерами різних типів.

Для побудови таблиці необхідно використати теги-контейнери: тег таблиці `<table>`, тег рядка `<tr>` та тег комірки даних `<td>` або тег комірки заголовка `<th>`. При цьому таблиця має бути розміщена в тілі HTML-документу, будується вона рядками, причому кількість комірок в рядках повинна бути однаковою. Таким чином одній таблиці відповідає один тег `<table>` та стільки тегів `<tr>`, скільки рядків має таблиця. Якщо в таблиці немає об'єднання комірок, то сумарна кількість тегів `<td>` та `<th>` дорівнює добутку кількості рядків на кількість колонок таблиці. Різниця між тегами `<td>` та `<th>` полягає в форматуванні та вирівнюванні розміщеного в них тексту. В тезі `<td>` текст відображається стандартним шрифтом з горизонтальним вирівнюванням "по лівому краю". В тезі `<th>` текст відображається напівжирним шрифтом з горизонтальним вирівнюванням "по середині". Для розміщення інформації (тексту або графіки) в середині таблиці необхідно помістити її в тег комірки. Таблиця може мати заголовок, якому відповідає парний тег `<caption>`, він має бути розміщений після тегу `<table>` до першого тегу `<tr>`.

Наведемо приклад HTML-коду для визначення таблиці, яка складається із заголовку та трьох рядків, в кожному із яких має бути дві комірки:

```
<table>
<caption>Заголовок таблиці </caption>
<tr><th>Рядок 1. Комірка 1</th><th>Рядок 1. Комірка 2</th></tr>
<tr><td>Рядок 2. Комірка 1</td><td>Рядок 2. Комірка 2</td></tr>
<tr><td>Рядок 3. Комірка 1</td><td>Рядок 3. Комірка 2</td></tr>
</table>
```

Для зручності ідентифікації комірок в них розміщено текст, що відповідає номеру рядка та номеру комірки в рядку. Відображення такої таблиці у вікні браузера показано на рис. 7.12.

При відсутності параметру *border* у тегах, що визначають таблицю, її межі не відображаються.

Розглянемо призначення найбільш використовуваних параметрів тегів `<table>`, `<caption>`, `<tr>`, `<td>` та `<th>`.

Тег `<caption>` найчастіше використовується тільки з параметрами ***align*** та ***valign***, що визначають відповідно горизонтальне та вертикальне вирівнювання заголовка таблиці.

Можливі значення параметру *align* тегу `<caption>`:

- ***right*** - заголовок вирівнюється по правому краю таблиці;
- ***center*** - заголовок вирівнюється по центру таблиці;
- ***left*** - заголовок вирівнюється по лівому краю таблиці.

Параметр *valign* дозволяє розміщувати заголовок над таблицею (значення ***top***) або під таблицею (значення ***bottom***). При відсутності параметрів *align* та

*align* вирівняний по центру заголовок розміщується над таблицею. Параметри тегу `<table>` та їх призначення наведені в табл. 7.7.

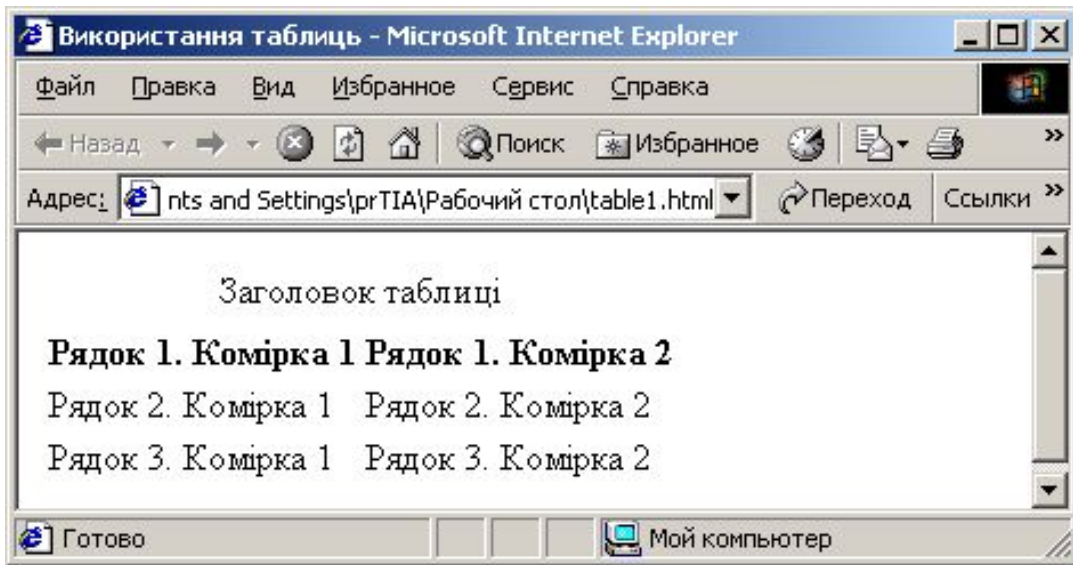


Рис. 7.12 - Використання таблиць

Таблиця 7.7 - Параметри тегу `<table>`

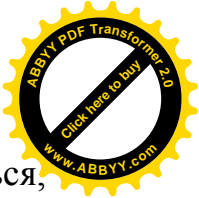
Параметр	Призначення
<i>border</i>	Визначає товщину межі таблиці в пікселях
<i>bordercolor</i>	Визначає колір меж таблиці
<i>cellspacing</i>	Визначає відстань між комітками таблиці в пікселях
<i>cellpadding</i>	Визначає відстань від меж комірок до даних, що знаходяться в цих комітках
<i>width</i>	Визначає ширину таблиці в пікселях або процентах від ширини HTML-документу
<i>height</i>	Визначає висоту таблиці в пікселях
<i>align</i>	Визначає горизонтальне вирівнювання таблиці
<i>bgcolor</i>	Визначає колір фону таблиці
<i>background</i>	Визначає фоновий рисунок таблиці

## 7.12 Фрейми

**Фреймова структура HTML-документу** дозволяє розділити вікно браузера на декілька прямокутних областей (**фреймів**), в кожену із яких можливо завантажити окремий HTML-документ.

Між фреймами можливо організувати **взаємодію**. Сенс **взаємодії між фреймами** полягає в тому, що вибір посилання в одному із фреймів може призвести до завантаження вказаного HTML-документу в інший фрейм.

В основному **фрейми застосовуються** для організації управління завантаженням HTML-документів в певну область вікна браузера при роботі користувача в іншій області, тобто для організації меню.



Хоча на сьогодні **фреймова структура** досить широко застосовуються, але вона **не відповідає сучасним тенденціям** в галузі побудови Web-сайтів.

Особливістю HTML-документу із фреймовою структурою є те, що кількість html-файлів на одиницю більше від кількості фреймів. Причиною цього є необхідність використання окремого html-файлу для визначення фреймової структури.

**Фреймова структура** HTML-документу визначається за допомогою тегу-контейнеру `<frameset>`, що використовується замість тегу `<body>`.

Тег `<frameset>` може мати параметри *rows* та *cols*, за допомогою яких визначається кількість та розміри областей, на які розділяється вікно браузера. За допомогою параметру *rows* вікно браузеру розділяється на горизонтальні області, а за допомогою параметру *cols* – на вертикальні області.

Синтаксис запису тегу:

```
<frameset rows="висота_1, висота_2..." cols="ширина_1, ширина_1...">
```

Кількість заданих "висот" та "ширин" задають кількість горизонтальних та вертикальних областей (**фреймів**), на які буде розділене вікно браузера. Для коректного визначення фреймової структури необхідно, як мінімум, задати хоча б два розміри для одного із параметрів *rows* чи *cols*.

Розміри фреймів можуть задаватись в **пікселях, процентах від розмірів вікна браузера** або **відносних одиницях**.

Наведемо приклади.

Визначення розмірів фреймів в пікселях:

```
<frameset rows="100,500">
```

 - визначає два горизонтальні фрейми висотою 100 та 500 пікселів.

Визначення розмірів фреймів в процентах:

```
<frameset rows="10%,90%">
```

 - визначає два горизонтальні фрейми висотою 10% та 90% від висоти вікна браузера.

Визначення розмірів фреймів в відносних одиницях:

```
<frameset rows="*,2*">
```

 - визначає два горизонтальні фрейми висотою  $\frac{1}{3}$  та  $\frac{2}{3}$  від висоти вікна браузера. В цьому випадку для показу відносних величин використовується символ \*.

В середині контейнеру `<frameset>` можуть розміщуватись тільки теги визначення окремого фрейма `<frame>` або вкладені теги `<frameset>`. Сумарна кількість цих тегів повинна бути рівною кількості фреймів, заданих параметрами *rows* та *cols* контейнеру `<frameset>`.

Параметри тегу `<frame>` представлені в табл. 7.8.

Наведемо приклад html-коду для визначення двох фреймів, що займають відповідно 25% та 75% від ширини вікна браузера.

Відзначимо, що межа між фреймами повинна відображатись у вікні браузера.

Лівий фрейм має ім'я *fmenu*, в ньому завантажений файл *fmenu.html*.

Правий фрейм має ім'я *fwindow*, в ньому завантажений файл *fwindow.html*.

Поля лівого фрейму становлять 5 пікселів, а поля правого фрейму 7 пікселів.

Розміри правого фрейму не можуть бути змінені користувачем.

```
<frameset cols="25%,75%">
    <frame src="fmenu.html" name="fmenu" frameborder="1"
marginwidth="5" marginheight="5">
    <frame src="fwindow.html" name="fwindow" marginwidth="7"
marginheight="7" noresize>
</frameset>
```

Відповідне вікно браузеру показано на рис. 7.13.

Таблиця 7.8 - Параметри тегу <frame>

Параметр	Призначення
<i>src</i>	Визначає URL- адресу документу, завантаженого в фрейм
<i>name</i>	Визначає ім'я фрейму, яке використовується для взаємодії між фреймами.
<i>frameborder</i>	Вказує на наявність ("1"), або відсутність ("0") межі між фреймами
<i>bordercolor</i>	Визначає колір межі (при <i>frameborder</i> ="1") між фреймами
<i>scrolling</i>	Задає наявність смуг прокрутки. "No" – немає, " Yes "– є, "Auto" – з'являються при необхідності
<i>marginwidth</i>	В пікселях визначає розміри горизонтальних відступів (полів) від меж фрейма до об'єктів, що в ньому відображаються
<i>marginheight</i>	В пікселях визначає розміри вертикальних відступів (полів) від меж фрейма до об'єктів, що в ньому відображаються
<i>noresize</i>	Наявність цього параметру забороняє користувачам міняти розміри фреймів.

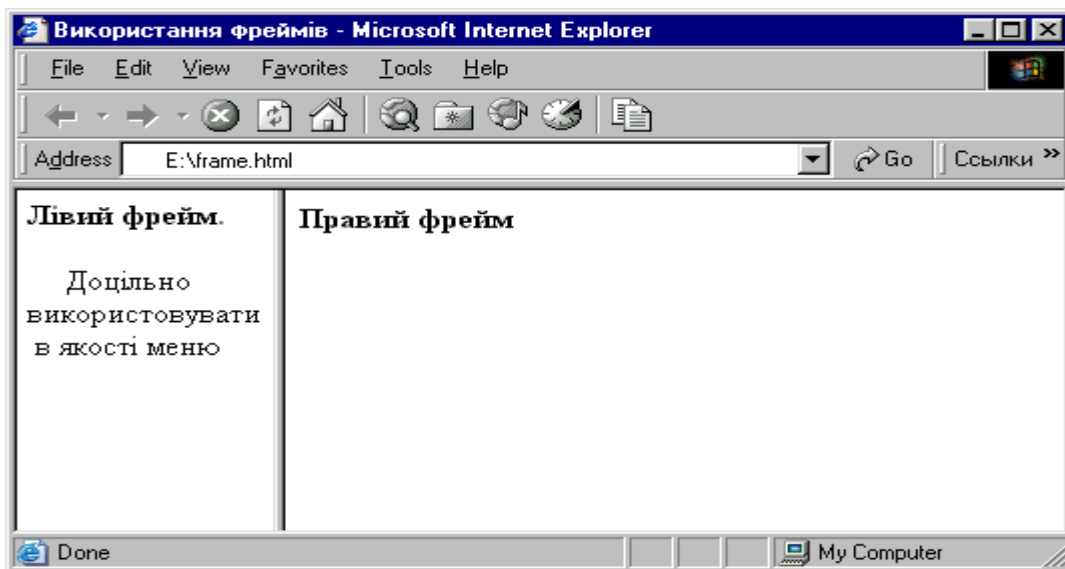
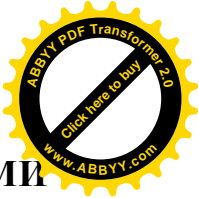


Рис. 7.13 - HTML-документ з фреймовою структурою





## 8 ПРОГРАМНІ ЗАСОБИ РОБОТИ ЗІ СТРУКТУРОВАНИМИ ДОКУМЕНТАМИ

### 8.1 Текстовий редактор WORD

#### 8.1.1 Завантаження редактора Word

Для завантаження редактора Word виконати команди:

**"Пуск" → "Програми" → "MsWord".**

При цьому запуститься на виконання основний файл *WinWord.exe* і буде створений документ з ім'ям *"Документ1"*, який має розширення *.doc*.

#### 8.1.2 Збереження документа

Для збереження документа виконати команди:

**"Файл" → "Сохранить".**

При першому збереженні документа з'являється вікно (рис. 8.1), у якому можна вказати папку, у якій зберігаєте файл, і ім'я файлу. Розширення файлу можна не вказувати, воно автоматично *буде .doc*.

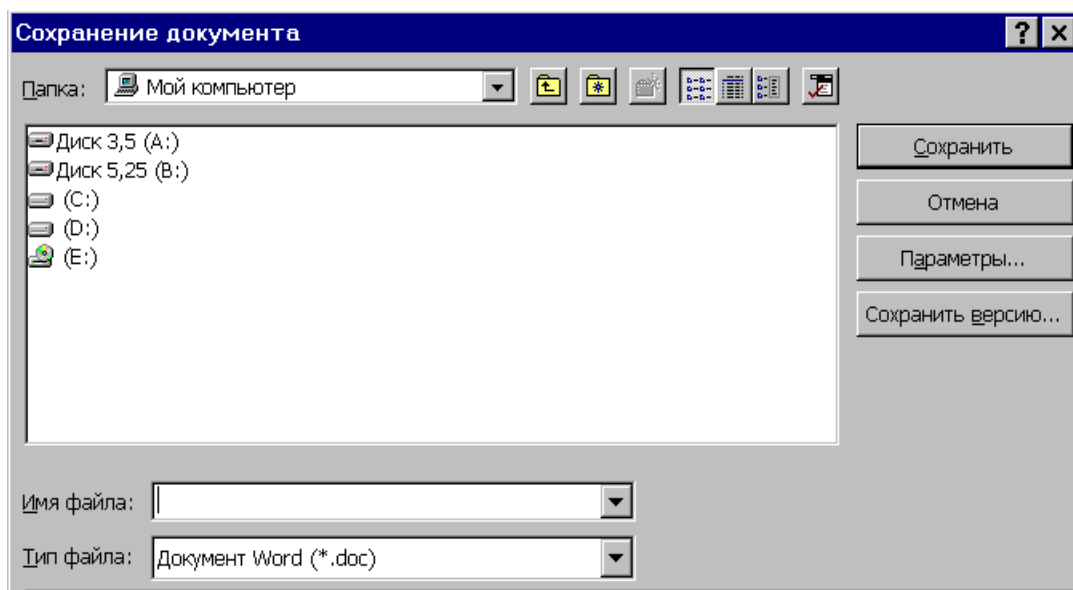


Рис. 8.1 – Вікно збереження документа

При повторному збереженні файлу це вікно не з'явиться, і файл збережеться з колишнім ім'ям на колишньому місці.

#### 8.1.3 Збереження файлу з іншим ім'ям або в іншій папці

Для збереження файлу з іншим ім'ям або в іншій папці виконати команди:

**"Файл" → "Сохранить как..."**

При цьому знову з'являється вікно як при першому збереженні файлу (рис. 8.1). У ньому треба набрати нове ім'я і нову папку і клацнути по кнопці **Сохранить**.



### 8.1.4 Відкриття документа

Для Відкриття документа виконати команди:

**"Файл" → "Открыть".**

З'явиться вікно, у якому потрібно вказати папку, що містить даний документ, і ім'я файлу, а потім клацнути по кнопці **Сохранить**.

### 8.1.5 Настроювання параметрів сторінки

Під настроюванням параметрів сторінки мається на увазі настроювання полів документа, орієнтації папера та інш.

Для настроювання параметрів сторінки виконати команди:

**"Файл" → "Параметры страницы"** (рис. 8.2) → вкладка **"Поля"** (для зміни полів зліва, справа, зверху, знизу) або вкладка **"Размер бумаги"** (для зміни розміру сторінки й *орієнтації* папера на *книжкову* або *альбомну*).

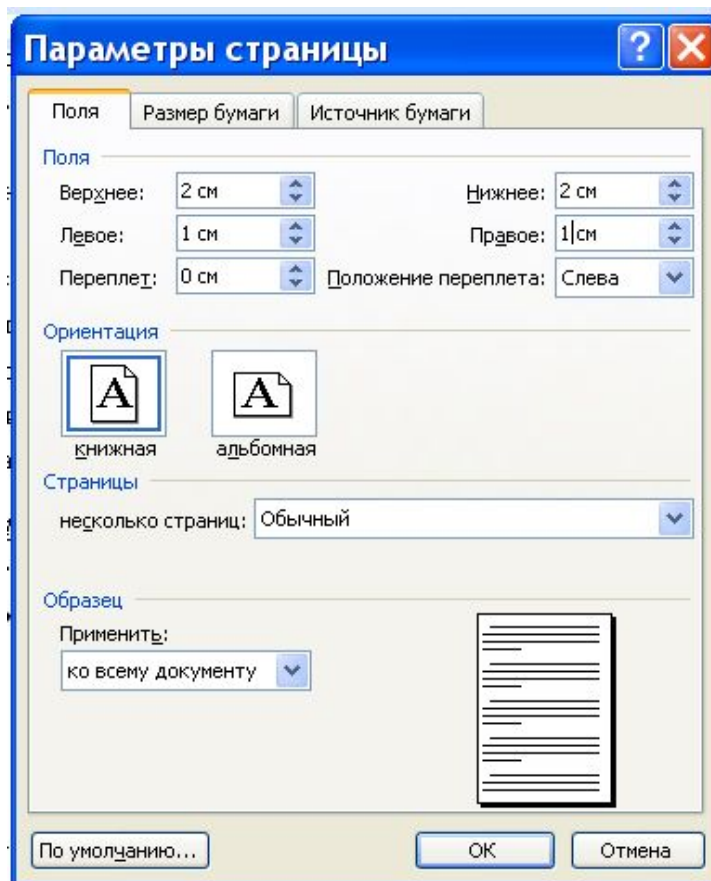


Рис. 8.2 – Вікно для установки параметрів сторінки

### 8.1.6 Друкування файлу

Файл, що знаходиться в редакторі можна надрукувати такими способами:

*1 спосіб:* Виконати команди: **"Файл" → "Печать"** → вказати які сторінки друкувати, скільки екземплярів (Рис. 8.3) і натиснути кнопку **Ок**

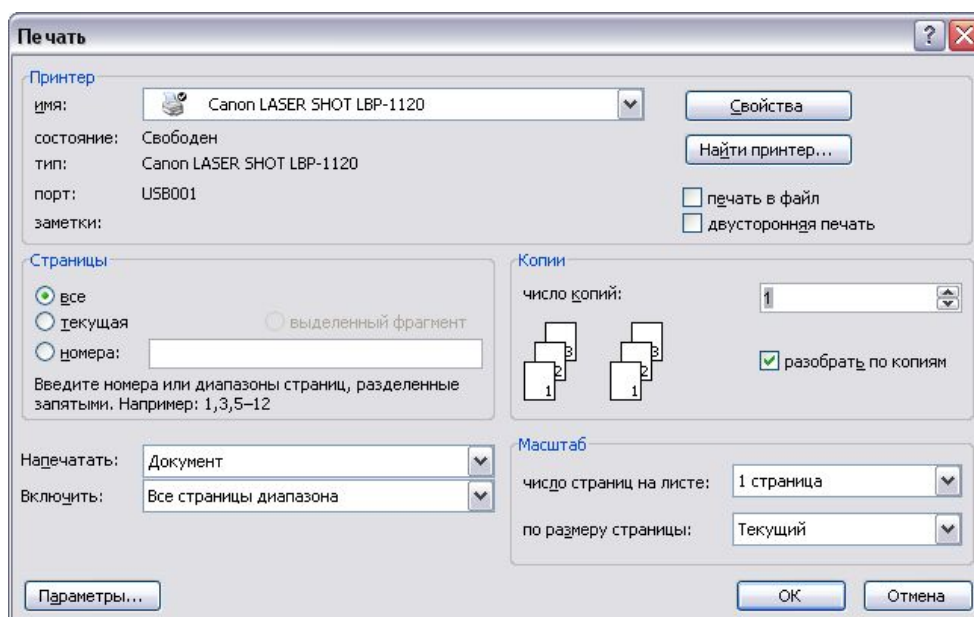


Рис. 8.3 – Вікно друкування документу

2 спосіб: Клацнути по значку  ("Печать") на панелі інструментів (При цьому надрукується весь документ)

### 8.1.7 Режим попереднього перегляду

Попередній перегляд документу можна здійснити такими способами:

1 спосіб: "Файл" → "Предварительный просмотр".

2 спосіб: На панелі інструментів вибрати .

Щоб повернутися з вікна попереднього перегляду треба клацнути по кнопці **Закертыть**.

### 8.1.8 Робота з панелями інструментів

Щоб додати або видалити панель інструментів необхідно:

1 спосіб: Клацнути правою кнопкою миші на панелі інструментів → клацанням на назві активізувати потрібну панель, якщо не було прапорця або видалити з екрана, якщо був прапорець.

2 спосіб: Виконати команди: "Вид" → "Панели инструментов" → вибрати назву потрібної панелі інструментів.

Стандартне положення панелі інструментів – у верхній частині екрана, але вона може знаходитися і будь-якому іншому місці екрана.

Якщо панель інструментів знаходиться в робочій частині, то називається такою, що плаває. Щоб зафіксувати панель, що плаває, необхідно двічі клацнути на її заголовку. Щоб перенести панель інструментів в інше місце, можна схопити її за вільне місце і перетягнути в потрібне місце.

#### 8.1.8.1 Підключення окремих кнопок до панелі інструментів

Для підключення окремих кнопок до панелі інструментів виконати

команди:

"Сервис" → "Настройка..." → вкладка "Команды" → вибрати в лівому вікні потрібну категорію, а праворуч – потрібну кнопку → схопити її і відпустити на панелі інструментів.

### 8.1.8.2 Видалення кнопки з панелі інструментів

Для видалення кнопки з панелі інструментів необхідно при натиснутій клавіші **Alt** схопити кнопку, що видаляється, і кинути її поза панеллю інструментів.

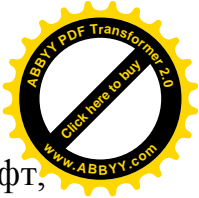
### 8.1.9 Виділення тексту

Смуга виділення – це невидимий стовпчик уздовж лівого краю вікна документа. Коли покажчик миші попадає в смугу виділення, він приймає форму стрілки ⇒, ⇨, ⇧.

Дії, які потрібно виконати для виділення окремих частин тексту, представлені в таблиці 8.1.

Таблиця 8.1 – Дії, потрібні для виділення окремих частин тексту

№ п/п	Що виділити?	Потрібні дії
1.	Виділити слово	Двічі клацнути лівою кнопкою миші всередині слова
2.	Виділити один рядок	Клацнути лівою кнопкою миші у смугі виділення рядка
3.	Виділити абзац	Тричі клацнути лівою кнопкою миші у будь-якому місці абзацу або двічі клацнути лівою кнопкою миші в смугі виділення абзацу
4.	Виділити речення	Клацнути лівою кнопкою миші в будь-якому місці речення при натиснутій клавіші <b>Ctrl</b>
5.	Виділити НЕ прямокутний фрагмент	Клацнути лівою кнопкою миші на початку фрагмента, а потім при натиснутій клавіші <b>Shift</b> клацнути наприкінці фрагмента (або розширювати границі →, ↓, ↑, ←)
6.	Виділити прямокутний фрагмент	Клацнути лівою кнопкою миші на початку фрагмента, а потім при натиснутій клавіші <b>Alt</b> здійснити виділення мишею.
7.	Розширити або зменшити зону виділення	При натиснутій клавіші <b>Shift</b> розширювати або зменшувати фрагмент клавішами ⇩, ⇧, ⇨, ⇩.
8.	Виділити весь документ	Тричі клацнути в смугі виділення або виконати команди: "Правка" → "Выделить все"
9.	Скасувати виділення	Клацнути лівою кнопкою миші в будь-якому не позначеному місці



Виділений фрагмент тексту можна форматувати, тобто змінювати шрифт, розмір, жирність, нахил, видаляти, а також переміщувати і копіювати в будь-яке місце даного документа, іншого документа, в окремий файл або друкувати.

### 8.1.10 Копіювання фрагмента

Для копіювання фрагмента необхідно виконати наступні дії:

1. Виділити фрагмент.
2. Виконати команди: **"Правка"** → **"Копировать"**.
3. Встановити курсор в місце, куди потрібно копіювати фрагмент.
4. Виконати команди: **"Правка"** → **"Вставить"**.

### 8.1.11 Переміщення фрагмента

Для переміщення фрагмента необхідно виконати наступні дії:

1. Виділити фрагмент.
2. Виконати команди: **"Правка"** → **"Вырезать"**.
3. Встановити курсор в місце, куди потрібно переміщати фрагмент.
4. Виконати команди: **"Правка"** → **"Вставить"**.

### 8.1.12 Видалення фрагмента

Для видалення фрагмента необхідно виконати наступні дії:

1. Виділити фрагмент.
2. Клацнути по клавіші  або **"Правка"** → **"Очистить"**.

### 8.1.13 Форматування тексту (символів)

Команди форматування відносяться до виділеного фрагмента, а якщо виділеного фрагмента немає, то вони будуть відноситися для набору наступних символів, поки знову не зміниться формат.

Щоб змінити шрифт (вид, розмір, підкреслення, колір букв і т.д.) можна виконати такі дії:

*1 спосіб:* Виконати команди: **"Формат"** → **"Шрифт"** → змінити потрібні параметри (рис. 8.4) → натиснути кнопку .

*2 спосіб:* Скористатися кнопками на панелі інструментів:

1) установка шрифту

Times New Roman	▼
-----------------	---

2) розмір шрифту можна вибрати в списку, що розкривається, або ввести в цьому полі з клавіатури

10	▼
----	---

3)  **Ж** жирний шрифт,

4)  **К** курсив,

5)  **Ч** підкреслений.

Будь-яку дію можна скасувати, повернувши на один крок назад кнопкою

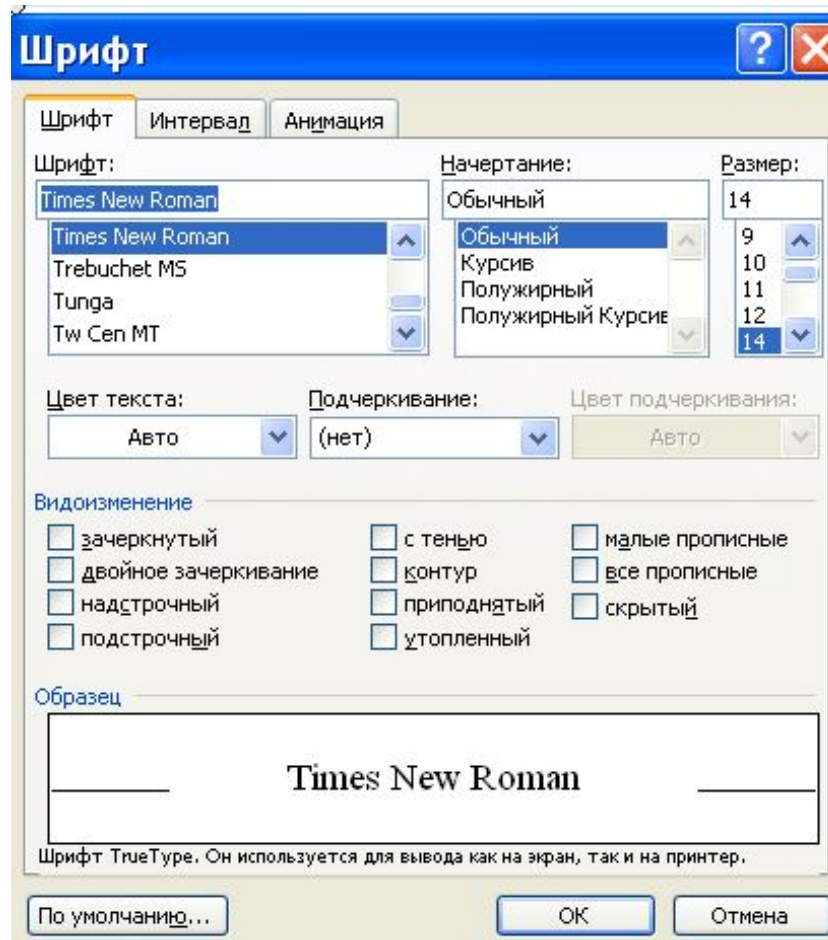


Рис. 8.4 – Вікно форматування символів

#### 8.1.14 Створення нестандартного підкреслення символів

Для створення нестандартного підкреслення виконати наступні дії: виділити символи, які потрібно підкреслити → "**Формат**" → "**Шрифт**" → відкрити список "**Подчеркивание**", що розкривається → вибрати потрібний вид підкреслення → клацнути по кнопці **ОК**.

#### 8.1.15 Вставка спеціальних символів

Для вставка спеціальних символів, тобто символів, яких немає на клавіатурі (*наприклад*, ∞, ≤, λ та інш.) необхідно виконати наступні дії: встановити курсор у місце, куди треба вставити символ → "**Вставка**" → "**Символ**" → вибрати тип шрифту (*наприклад*, Symbol) → виділити символ, що вставляється → клацнути по кнопці **Вставить** → клацнути по кнопці **Закерить**.

#### 8.1.16 Вставка нижніх і верхніх індексів

Щоб набрати фрагмент тексту з нижніми або верхніми індексами можна:



1. Набрати символи один поруч з іншим.
2. Виділити символ, що повинний стати індексом і перетворити його в індекс одним зі способів:

1 спосіб. "**Формат**" → "**Шрифт**" → встановити прапорець біля одного з ефектів: "**верхний индекс**" або "**нижний индекс**".

2 спосіб. На панелі інструментів клацнути по кнопці  $X^2$  для перетворення символу на верхній індекс або по кнопці  $X_2$  для перетворення символу на нижній індекс.

### 8.1.17 Форматування абзаців

Для форматування одного абзацу досить помістити курсор у будь-яке місце цього абзацу і зробити потрібне форматування.

Для форматування декількох абзаців одночасно необхідно спочатку виділити ці абзаці, а потім їх відформатувати.

Для форматування одного або декількох абзаців одночасно необхідно виконати команди:

"**Формат**" → "**Абзац**" → встановити потрібні параметри (рис.8.5) → **OK**.  
В таблиці 8.2 приведені параметри, які можна встановити для абзаців.

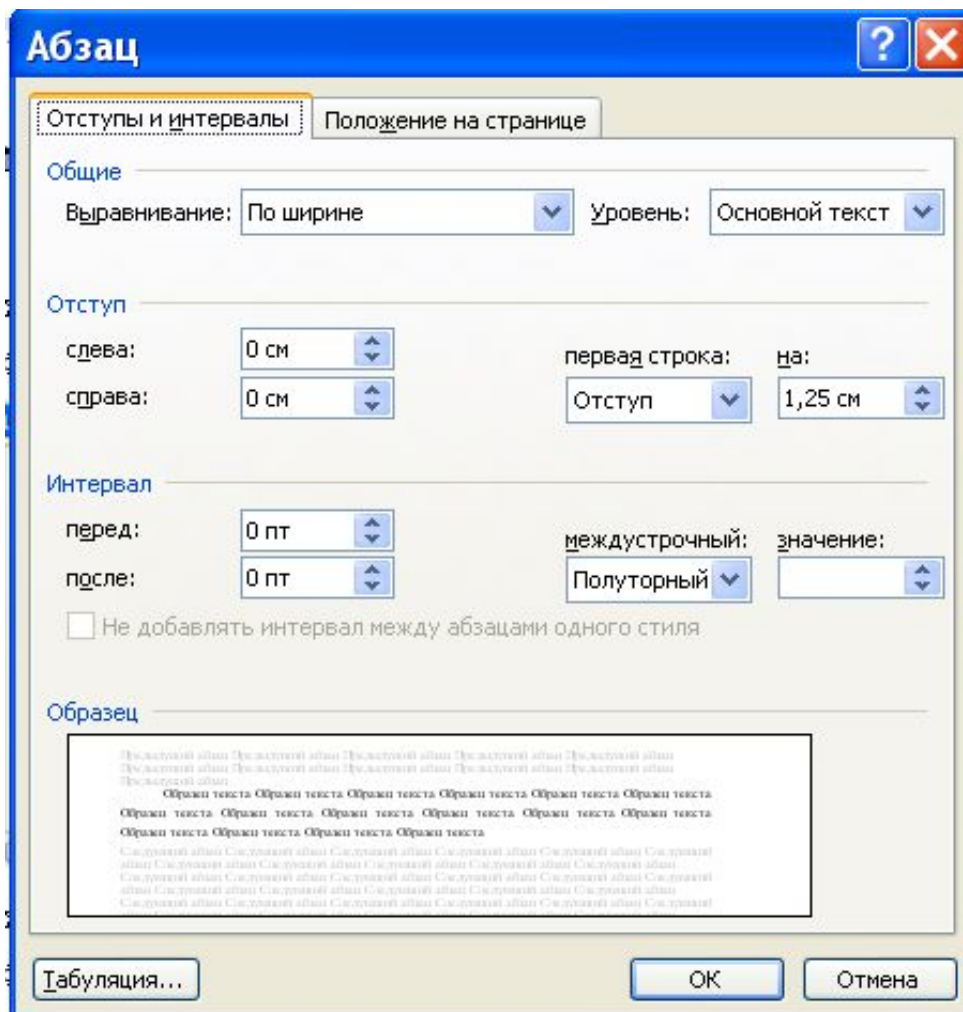


Рис. 8.5 – Вікно форматування абзацу



Таблиця 8.2 – Параметри абзаців

№ п/п	Параметр	Пояснення
1.	Вирівнювання тексту абзацу	По лівому краю, по правому краю, по центру, по ширині
2.	Відступи для всього абзацу	Зліва і справа від полів документа
3.	Перший рядок абзацу	Відступ, виступ від лівої границі абзацу
4.	Інтервал перед абзацом	Додатковий відступ, що вставляється зверху перед абзацом
5.	Інтервал після абзацу	Додатковий відступ, що уставляється внизу після абзацу.
6.	Міжрядковий інтервал	Відстань між рядками тексту в абзаці. ( <i>наприклад</i> , одинарний, подвійний...)

Команди **Вирівнювання** винесені на панель інструментів у виді кнопок:



по лівому краю



по правому краю




по центру



по ширині (по лівому і правому краях одночасно)


### 8.1.18 Створення рамки навколо фрагмента тексту

Для створення рамки навколо фрагмента тексту необхідно виконати такі дії: виділити фрагмент → "**Формат**" → "**Границы и заливка**" → вкладка "**Граница**" → вибрати *Тип* "**Рамка**" → вибрати потрібний тип лінії рамки (суцільна, пунктирна, хвиляста і т.д.) → **ОК**.

Щоб створити просту рамку навколо фрагмента тексту, можна виділити його і клацнути по кнопці на панелі інструментів  (зовнішні границі).

### 8.1.19 Створення газетних стовпчиків

При створенні стовпчиків у частині документа необхідно:

1. Прослідити, щоб виводилися символи, що не друкуються. Якщо їх немає, то клацнути мишею на Панелі інструментів по кнопці  - "**Непечатаемые символы**"
2. Виділити фрагмент, що розбивається на стовпчики і при виділенні фрагмента НЕ слід виділяти маркер кінця тексту.
3. Виконати команди: "**Формат**" → "**Колонки**".
4. У вікні, що показане на рис. 8.6, вибрати кількість стовпчиків, ширину і проміжок між колонками.
5. Якщо потрібно, щоб між колонками була лінія, то активізувати режим

√ "Разделитель".  
 6. Натиснути кнопку **ОК**.

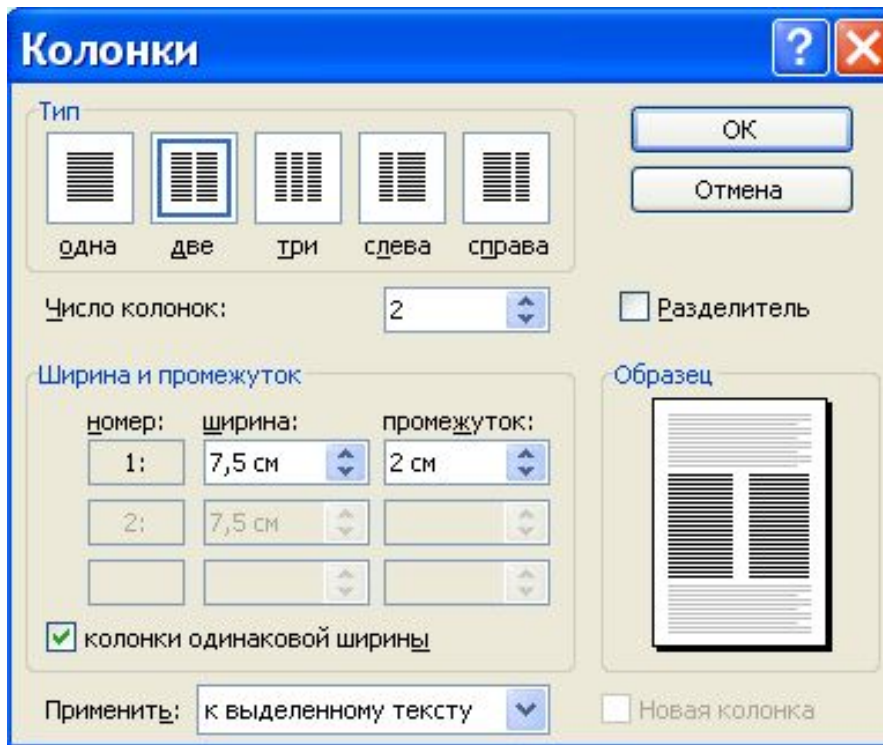


Рис. 8.6 – Вікно для установки газетних стовпчиків

**Зауваження!!!** Щоб змінити параметри встановлених раніше стовпчиків, необхідно помістити курсор усередині тексту стовпчиків (але НЕ виділяти текст стовпчиків) і виконати команди: "**Формат**" → "**Колонки**" → змінити потрібні параметри → **ОК**.

### 8.1.20 Створення списків

Список – це абзаци, на початку яких ставиться символ або число.

Види списків:

Маркіровані списки:

- Вставляється на початку кожного абзацу спеціальний символ (маркер).
- Маркер редагувати і змінювати не можна.
- У кожному абзаці у всіх рядках здійснюється відступ.

Нумеровані списки:

1. Ставиться номер на початку абзацу, а не маркер.
2. Номера виділяти і видаляти не можна.
3. Усі рядки в абзаці мають відступ як у першому рядку абзацу.
4. Якщо рядок додається або видаляється, нумерація обновляється.

Багаторівневі списки.

- 1. ....
- 1.1.....
- 1.1.1 ...

Щоб створити список можна виконати команди: "**Формат**" → "**Список**" → вибрати одну з вкладок "**Маркированный**" (рис. 8.7), "**Нумерованный**" (рис. 8.8) або "**Многоуровневый**" (рис. 8.9) → вибрати вид списку → **OK** → почати писати текст списку → після натискання по клавіші **Enter** список продовжиться.

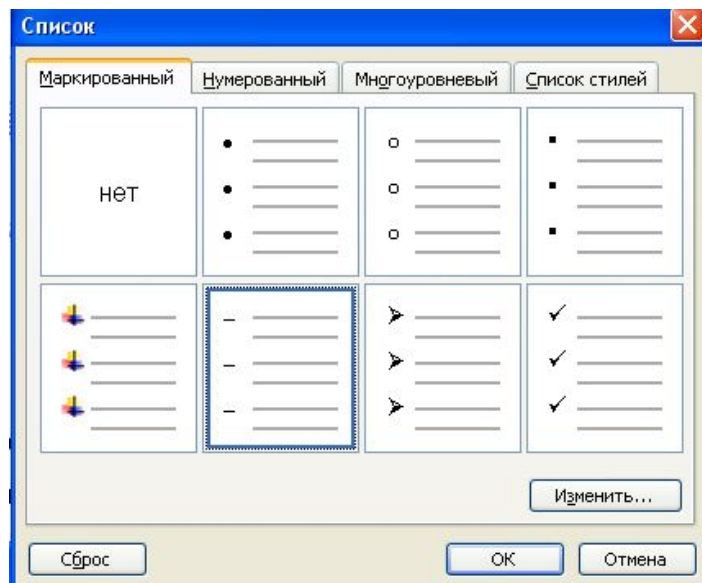


Рис. 8.7 – Вікно вибору маркірованого списку

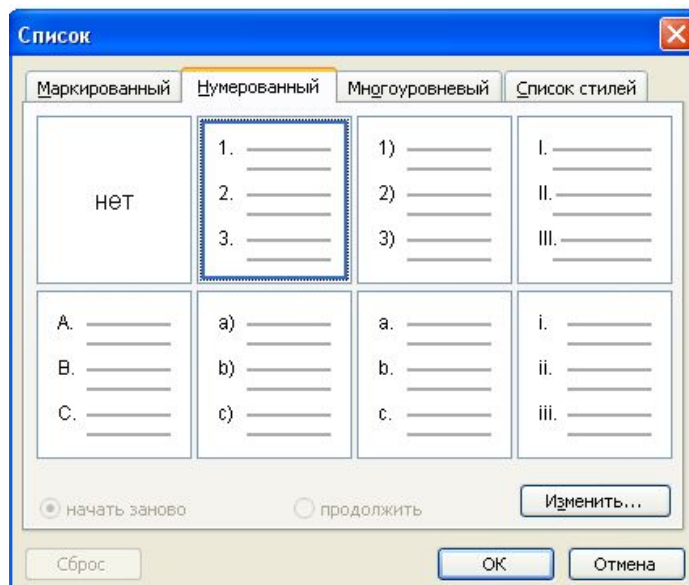


Рис. 8.8 – Вікно вибору нумерованого списку

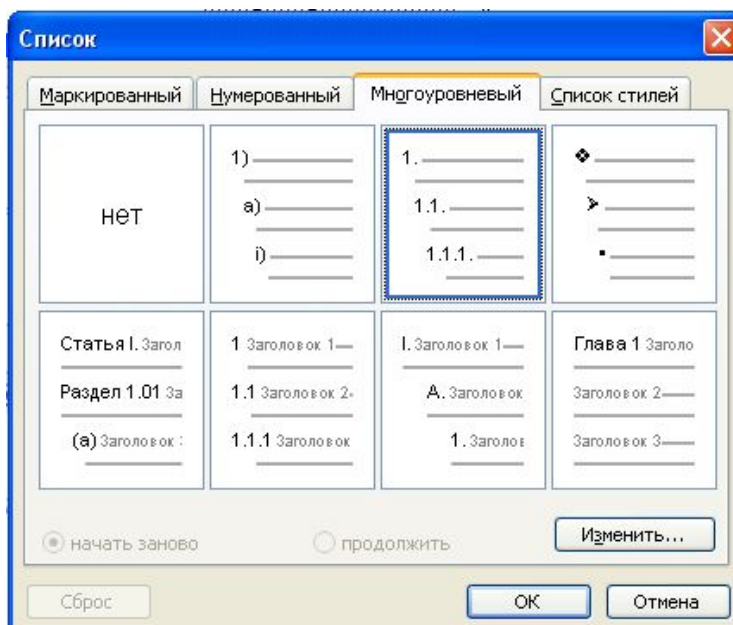


Рис. 8.9 – Вікно вибору багаторівневого списку

Щоб припинити список, тобто перестати нумерувати або маркірувати абзаци, необхідно двічі підряд натиснути клавішу **Enter**.

Щоб видалити список, тобто скасувати маркірування або нумерацію абзаців, необхідно виділити ці абзаци → "**Формат**" → "**Список**" → вибрати вид списку "**НЕТ**" → **OK**.

Щоб перетворити в список звичайні абзаци можна: виділити ці абзаци → "**Формат**" → "**Список**" → вибрати потрібну вкладку і вид списку → **OK**

Щоб змінити вид маркірування або нумерації, треба: виділити абзаци списку → "**Формат**" → "**Список**" → вибрати потрібну вкладку → вибрати потрібний вид або той, що найбільше влаштовує і клацнути по кнопці **Изменить**, внести потрібні зміни й клацнути по кнопці **OK**.

### 8.1.21 Створення багаторівневого списку

Щоб створити багаторівневий список можна виконати такі дії:

1. Всі абзаци, що будуть входити в список зробити абзацами без відступів.
2. Виділити фрагмент, для якого встановлюється список.
3. Виконати команди: "**Формат**" → "**Список**" → вкладка "**Многоуровневый**".
4. У вікні, що з'явилося, вибрати вид списку. Якщо в пропонованому переліку немає потрібного, то вибрати той, що найбільше влаштовує і скористатися кнопкою **Изменить**, а потім настроїти кожен рівень за своїм розсудом.

5. При створенні багаторівневого списку в такий спосіб всі абзаци виділеного фрагмента будуть пронумеровані як перший рівень (*наприклад*, 1., 2., 3., 4. і т.д.)
6. Для створення другого рівня (*наприклад*, для створення нумерації 1.1., 1.2., 1.3. і т.д.) треба виділити абзаци, що відносяться до другого рівня і натиснути клавішу **ТАВ** або виділити його і клацнути на ньому правою кнопкою миші, а потім у меню об'єкта вибрати **"Увеличить отступ"**.
7. Для створення наступних рівнів (*наприклад*, для створення нумерації 1.1.1., 1.1.1.1., 1.1.1.1.1. і т.д.) виконати дії, аналогічні діям попереднього пункту.
8. Для перетворення абзацив якогось рівня на абзаци більш високого рівня (*наприклад*, для перетворення нумерації з 1.1.1 на 1.2.) необхідно виділити ці абзаци і клацнути на них правою кнопкою миші, а потім у меню об'єкта вибрати **"Уменьшит отступ"**.

### 8.1.22 Нумерація сторінок

Для здійснення нумерації сторінок необхідно виконати команди:

**"Вставка"** → **"Номера страниц"** → указати положення і вирівнювання номера сторінки, а також чи нумерувати першу сторінку (рис. 8.10) → натиснути кнопку **ОК**.

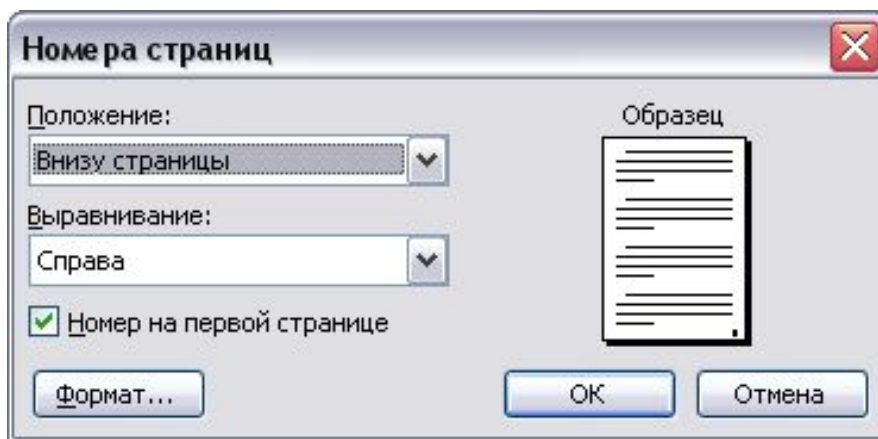


Рис. 8.10 – Вікно для визначення номерів сторінок

### 8.1.23 Видалення номерів сторінок

Для видалення номерів сторінок необхідно виконати такі дії:

1. Двічі клацнути на номері будь-якої сторінки. При цьому з'явиться панель інструментів **"Колонтитулы"**
2. В меню Word виконати команди: **"Правка"** → **"Выделить все"** → **"Правка"** → **"Очистить"**
3. Закрити панель інструментів **"Колонтитулы"**.

### 8.1.24 Установка переносу слів

Для включення або відключення автоматичного переносу слів необхідно виконати такі команди: **"Сервис"** → **"Язык"** → **"Расстановка переносов..."** (рис. 8.11). При цьому включення режиму **"Автоматическая расстановка переносов"** створює перенос слів у всьому документі незалежно від того, коли він був включений.

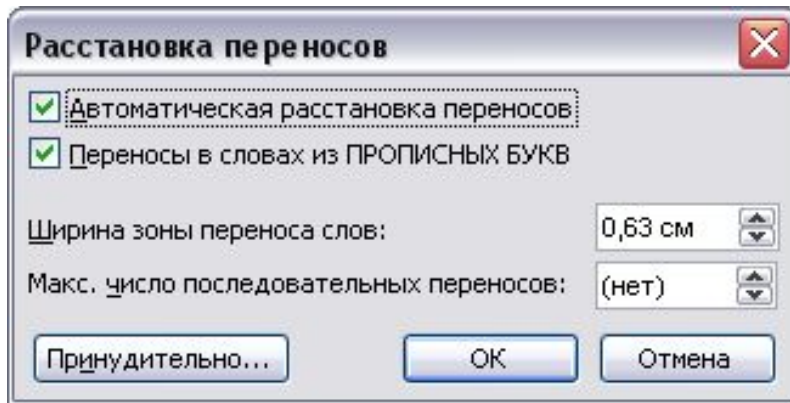


Рис. 8.11 – Вікно установки переносу слів

### 8.1.25 Робота з таблицями

#### 8.1.25.1 Створення таблиці

Для створення таблиці необхідно виконати такі дії: встановити курсор у місце, куди потрібно вставити таблицю → **"Таблица"** → **"Добавь таблицу"** → в вікні діалогу вказати

Число стовпців:	2	◆
Число рядків:	2	◆
Ширина стовпця:	АВТО	◆

і натиснути кнопку **Ок**.

#### 8.1.25.2 Розміри комірок

Змінити ширину стовпця, рядка або окремої комірки можна:

- 1 спосіб. Схопити за лінію сітки стовпця, рядки або комірки → пересувати її в потрібному напрямку.
- 2 спосіб. Виділити стовець, рядок або комірку → **"Таблица"** → **"Высота и ширина ячейки"** → вказати ширину стовпця або висоту рядка.



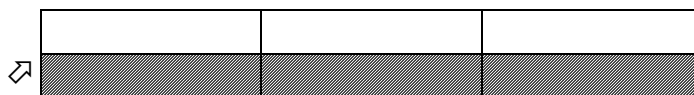
### 8.1.25.3 Виділення рядків, стовпців і комірок

1. Виділення комірки:



Клацнути мишею по комірці, коли миша усередині комірки має вигляд стрілки ↗, тобто коли миша знаходиться в лівій границі комірки.

2. Виділення рядка:

Клацнути мишею в смузї виділення напроти виділюваного рядка

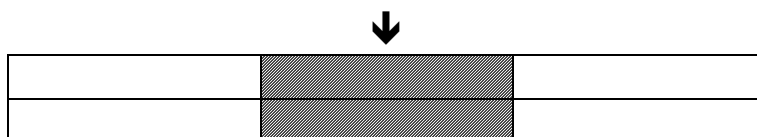


3. Виділення декількох рядків:



Виділити поруч розташовані рядки можна, клацнувши в смузї виділення по першому виділюваному рядку, а потім при натиснутій **Shift** розширити виділення клавішами  і .

4. Виділення стовпця:





Повільно підвести курсор до верхньої границі стовпця, поки курсор миші не прийме вид маленької чорної стрілки в границі і клацнути мишею.



5. Виділення декількох стовпців:

Виділити поруч розташовані стовпці можна, виконавши виділення одного стовпця і при натиснутій **Shift** розширити виділення клавішами  і .

6. Виділення декількох комірок:

Клацнути мишею в лівій границі кутові комірки, виділивши її. Натиснути **Shift** і не відпускаючи, розширити виділення клавішами , ,  і .

7. Виділення таблиці:

Установити курсор в одній з комірок таблиці → "**Таблиця**" → "**Выделить таблицу**".

### 8.1.25.4 Зміна орієнтації тексту в комірці

Щоб змінити орієнтацію тексту в комірці, наприклад, зверху вниз або знизу нагору, необхідно в меню об'єкта цього комірки (праве клацання миші) вибрати "**Направление текста**", і у вікні, що з'явиться (рис. 8.12) вибрати потрібну орієнтацію і клацнути по клавіші **OK**.

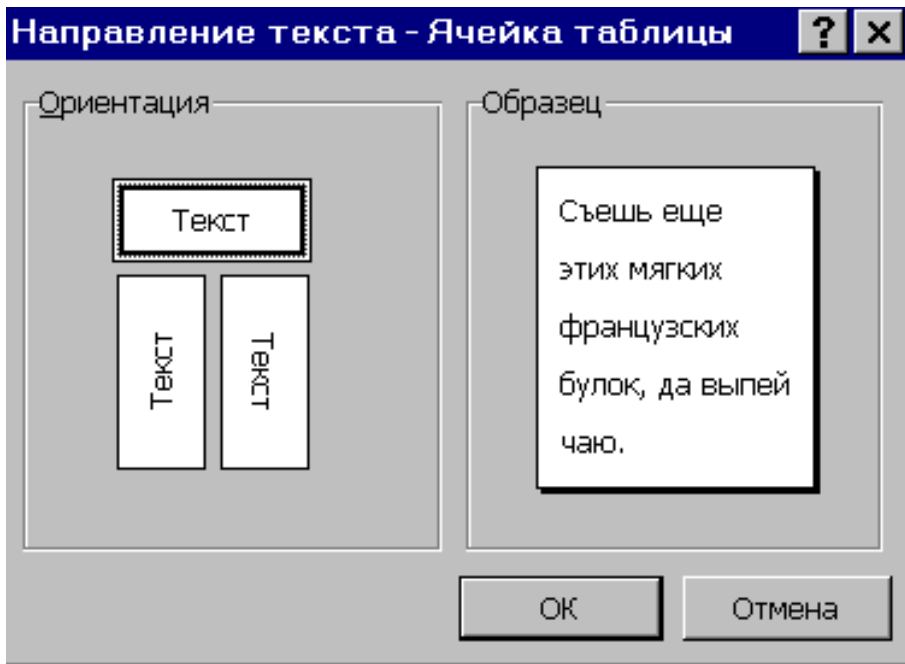


Рис. 8.12 – Вікно зміни орієнтації тексту в комірці таблиці

#### 8.1.25.5 Заливання комірок кольором

Для заливання комірок кольором необхідно виконати такі дії: виділити комірки → **"Формат"** → **"Границы и заливка"** → вкладка **"Заливка"** (рис. 8.13) → вибрати колір заливання → натиснути кнопку **ОК**.

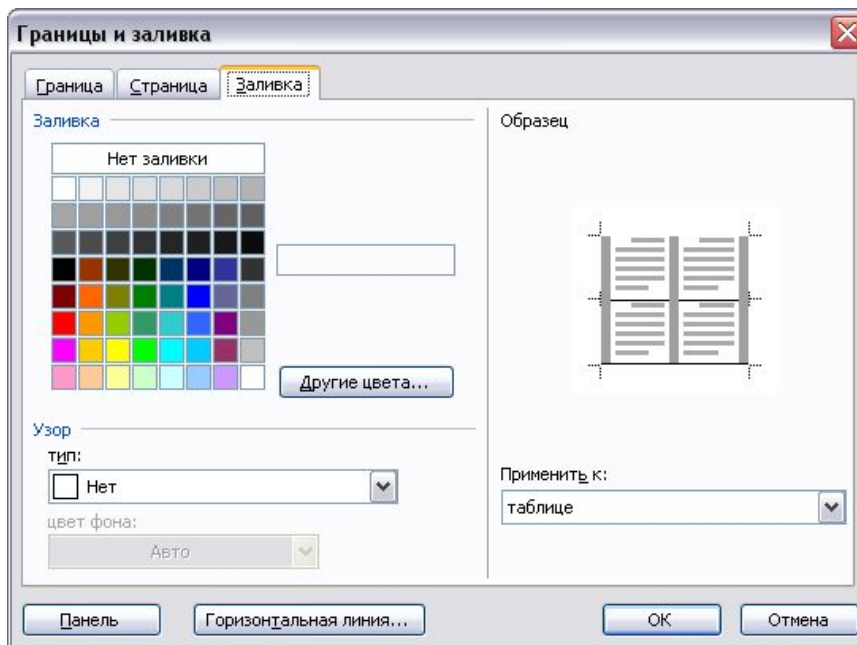


Рис. 8.13 - Вікно для визначення кольору заливання комірки таблиці

#### 8.1.25.6 Зміна виду границь комірок

Для зміна виду границь комірок необхідно виконати такі дії: виділити

комірки → **"Формат"** → **"Границы и заливка"** → вкладка **"Граница"** (рис. 8.14) → вибрати колір, тип, товщину і місцезнаходження границі → натиснути кнопку **OK**.

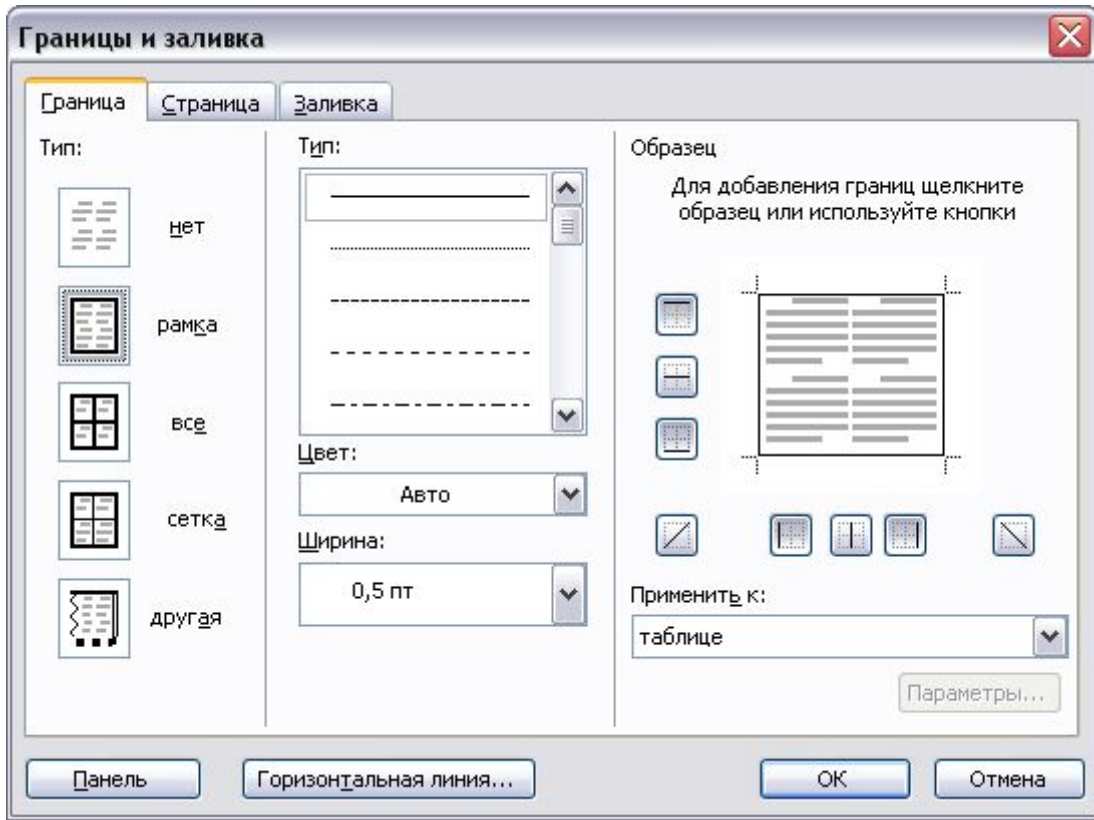


Рис. 8.14 – Вікно для визначення границь комірок таблиці

Крім того для зміни місцезнаходження границь, можна скористатися кнопкою на панелі інструментів



### 8.1.25.7 Об'єднання комірок таблиці

Для об'єднання комірок таблиці необхідно: виділити комірки → **"Таблица"** → **"Объединить ячейки"**.



### 8.1.25.8 Розбивка комірок таблиці

Для розбивка комірок таблиці необхідно: виділити комірку → **"Таблица"** → **"Разбить ячейку"** → указати на скількох рядків і стовпців розбити комірку.



### 8.1.25.9 Видалення рядків, стовпців, комірок

Для видалення рядків, стовпців, комірок необхідно: виділити рядки, стовпці або комірки → **"Таблиця"** → один з пунктів: **"Удалить строки"**, **"Удалить столбцы"** або **"Удалить ячейки"**.

### 8.1.25.10 Вставка рядка в будь-якому місці таблиці

Для вставки рядка в будь-якому місці таблиці необхідно: виділити рядок, перед яким потрібно вставити новий рядок → **"Таблиця"** → **"Вставить строки"**.

### 8.1.25.11 Додавання рядка в кінець таблиці

Для додавання рядка в кінець таблиці можна скористатися одним із способів:

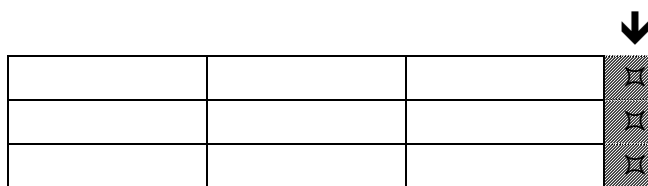
- 1 спосіб. Помістити курсор в останню комірку останнього рядка таблиці і натиснути
- 2 спосіб. Помістити курсор за останню комірку останнього рядка і натиснути клавішу .

### 8.1.25.12 Додавання стовпця в таблицю

Для додавання стовпця в таблицю необхідно: виділити стовець, перед яким потрібно вставити новий → **"Таблиця"** → **"Вставить столбцы"**.

### 8.1.25.13 Додавання стовпця в правий край таблиці

1. Для зручності зробити видимими маркери кінця рядків, клацнувши мишею по кнопці **"Непечатаемые символы"**, на стандартній панелі інструментів.
2. Виділити маркери кінця рядків, як стовець.





3. Виконати команди: **"Таблиця"** → **"Вставить столбцы"**.
4. Зробити невидимими маркери кінця рядків, зробивши неактивною кнопку **"Непечатаемые символы"**, на стандартній панелі інструментів.

### 8.1.25.14 Переміщення і копіювання рядків, стовпців і комірок


Щоб використовувати метод перетаскування необхідно виконати команди: **"Сервис"** → **"Параметры"** → вкладка **"Правка"** → установити прапорець

<input checked="" type="checkbox"/>	Використовувати перетаскування тексту при виправленні
-------------------------------------	---

## – Переміщення рядків або стовпців

Виділити рядки або стовпці → схопити їх за ліву верхню комірку у той момент, коли курсор стане  → перетягнути в потрібне місце.

### 1. Копіювання рядків або стовпців

Виділити рядки або стовпці → схопити їх у той момент, коли курсор стане  → при натиснутої Ctrl перетягнути в потрібне місце.

### 2. Переміщення і копіювання комірок

Виділити комірки.

Для переміщення необхідно схопити виділену частину і перетягнути в потрібне місце. При цьому на колишнім місці комірки стануть порожніми, а їхній вміст вставиться на нове місце, видаливши з них колишню інформацію.

Для копіювання необхідно перетаскувати виділену частину при натиснутій клавіші **Ctrl**.

Щоб при копіюванні з однієї комірки в іншу, в другій комірці текст не знищувався, а до нього додавався текст, що копіюється, необхідно виділити в першій комірці тільки текст (замість всієї комірки) і перетягнути його в нову комірку.

## 8.1.25.15 Сортування даних у таблиці

Сортування здійснюється у вертикальному напрямку.

1. Виділити область таблиці, що сортується. Для сортування всієї таблиці досить, щоб курсор знаходився усередині таблиці. Але якщо заголовки стовпців таблиці містять об'єднані комірки, то потрібно виділити таблицю без рядка заголовків.
2. Виконати команди: **"Таблиця"** → **"Сортировка"**.
3. У вікні, що показане на рис. 8.15

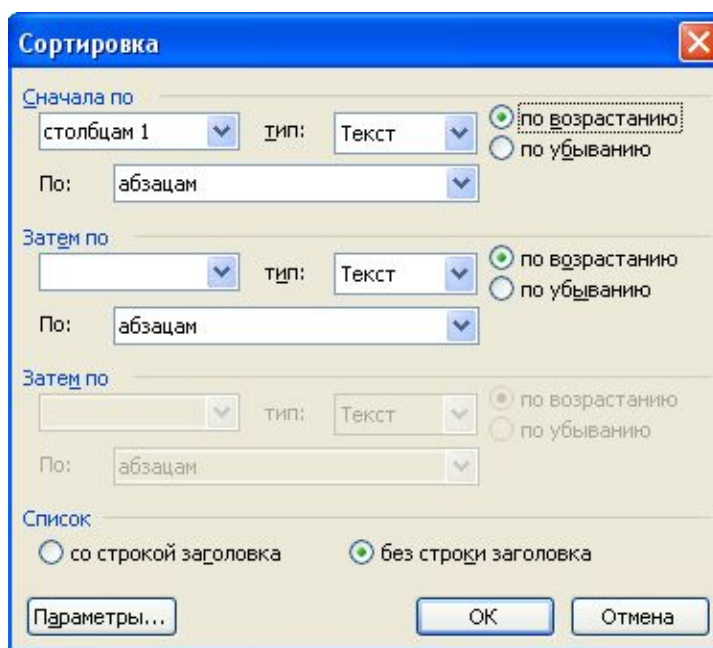


Рис. 8.15 – Вікно сортування даних

- активізувати перемикач "со строкой заголовка" або "без строки заголовка" в залежності від того, як була виділена область для сортування
- у полі "Сортировать" вказати стовпець, що буде служити критерієм упорядкування.
- у полі "Тип" вказати тип інформації в стовпці-критерії упорядкування.
- активізувати перемикач "по возрастанию" або "по убыванию"

4. Натиснути .

Якщо критерій упорядкування складний, то необхідно для наступного стовпця, що входить у критерій заповнити запити "Затем" аналогічно першому стовпцеві критерія.

**Зауваження!** При сортуванні тільки одного або декількох стовпців, необхідно: виділити ці стовпці → "Таблица" → "Сортировка" → кнопка  → активізувати режим Параметри сортировки  
 Только столбцы

В іншому випадку, якщо навіть виділити один або кілька стовпців, сортуватися будуть усі стовпці таблиці таблиці.

### 8.1.26 Вставка об'єкта WordArt

Об'єкти WordArt дозволяють вставляти текст, що нестандартно отформатовано (*наприклад*, об'ємний, перекошений, вигнутий і т.п.).

Для вставки об'єкта WordArt необхідно виконати команди: "Вставка" → "Рисунок" → "Объект WordArt" (рис. 8.16) → Із запропонованого переліку стилів написів вибрати потрібний, двічі клацнувши по ньому мишею → у вікні, що відкрилося, потрібно написати текст, вибрати шрифт, розмір шрифту і клацнути по кнопці .

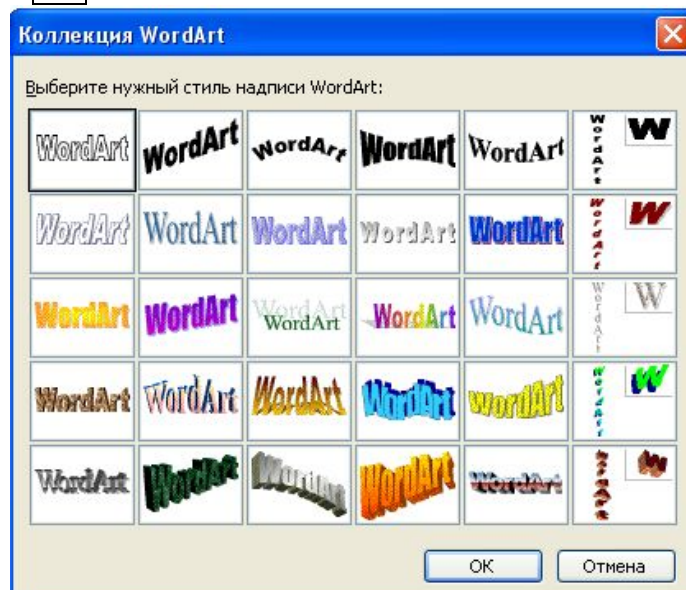
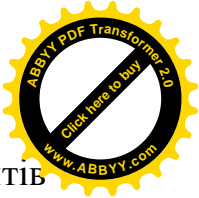


Рис. 8.16 – Вікно вставки об'єкту WordArt





Для зміни об'єкта WordArt можна використовувати панель інструментів WordArt або в меню правого клацання цього об'єкта вибрати пункт "**Формат об'єкта WordArt**".

### 8.1.27 Вставка готових малюнків

1. Для вставки стандартного малюнка необхідно виконати такі дії:  
встановити курсор у місце, куди треба вставити малюнок → "**Вставка**" → "**Рисунок**" → "**Картинки...**" → у переліку стандартних малюнків, що з'явився, виділити потрібний → клацнути по кнопці **Вставить**.  
Обраний малюнок буде вставлений у поточне положення курсору.
2. Для вставки малюнка з файлу необхідно виконати такі дії:  
встановити курсор у місце, куди треба вставити малюнок → "**Вставка**" → "**Рисунок**" → "**Из файла...**" → указати потрібний файл → клацнути по кнопці **Добавить**.

### 8.1.28 Зміна розмірів графічних об'єктів

Виділити графічний об'єкт, клацнувши по ньому один раз. Довкола нього з'явиться рамка з квадратиками:

Необхідно схопити за один з них курсором (↔) і переміщати в потрібному напрямку, змінюючи границі.

### 8.1.29 Переміщення графічних об'єктів

Виділити об'єкт, схопити його мишею, коли покажчик прийме вид сніжинки (✦), і перетягнути його в потрібне місце.

Можна рухати об'єкт униз, нагору, вправо, вліво клавішами керування курсором (↓, ↑, →, ←).




### 8.1.30 Зміна параметрів графічних об'єктів

Щоб змінити вид, розмір, колір лінії, обрізку, обтікання, положення необхідно виконати дії: виділити фігуру → "**Формат**" → останній пункт ("**Об'єкт**" або "**Рисунок**") → вибрати потрібну вкладку → змінити потрібний параметр і натиснути **ОК**.

#### ▪ 8.1.31 Обрізка малюнка

Якщо потрібно залишити тільки частину малюнка (прямокутну), то малюнок потрібно обрізати. Для цього необхідно підключити панель інструментів "**Настройка изображения**", потім виділити малюнок і на

підключеній панелі клацнути по кнопці  , а потім схопити цим значком один із граничних квадратиків і перемістити в потрібному напрямку, обрізуючи малюнок.

### 8.1.32 Обтікання малюнка текстом

При вставці малюнка текст буде розташований вище або нижче малюнка. Щоб розташувати текст біля малюнка або прямо по малюнку, потрібно змінити обтікання малюнка текстом.

Для цього виконати дії: виділити малюнок → **"Формат"** → **"Рисунок"** → вкладка **"Положение"** → вибрати один з видів обтікання (рис. 8.17 - "в тексте", "вокруг рамки", "по контуру", "за текстом", "перед текстом")

Якщо потрібно встановити вид обтікання "сверху и снизу" или "сквозное", необхідно виконати дії: виділити малюнок → **"Формат"** → **"Рисунок"** → вкладка **"Положение"** → клацнути по кнопці **Дополнительно** → вкладка **"Обтекание текстом"** → вибрати один з видів обтікання (рис. 8.18)

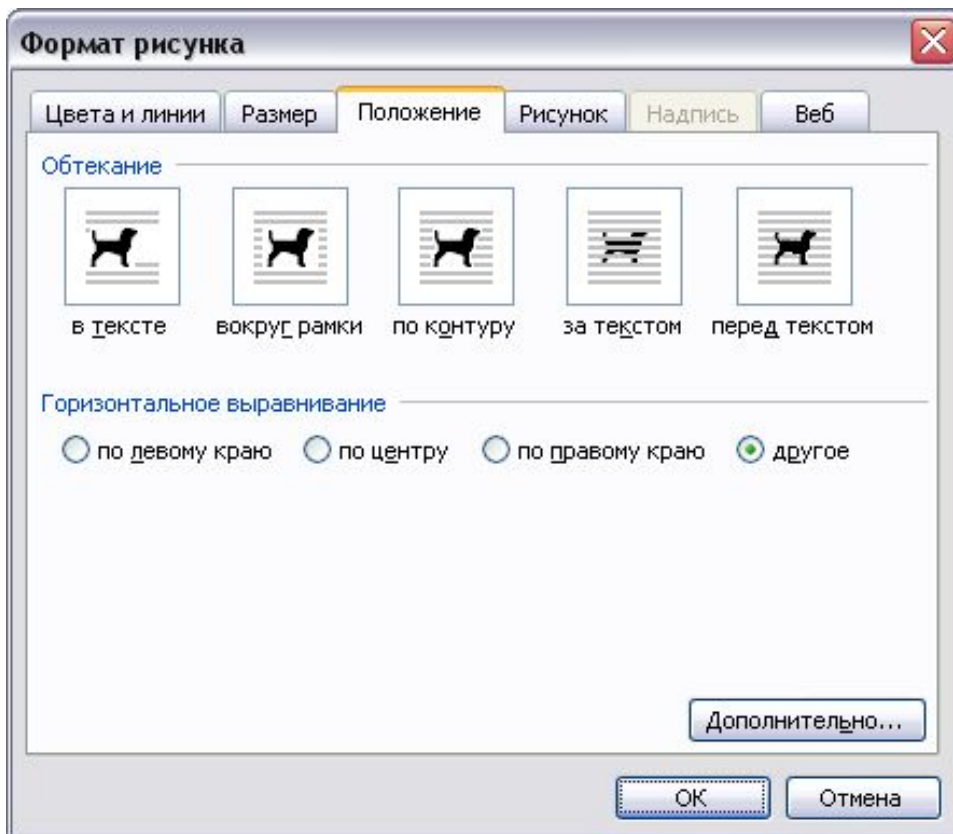


Рис. 8.17 – Види обтікання малюнка текстом

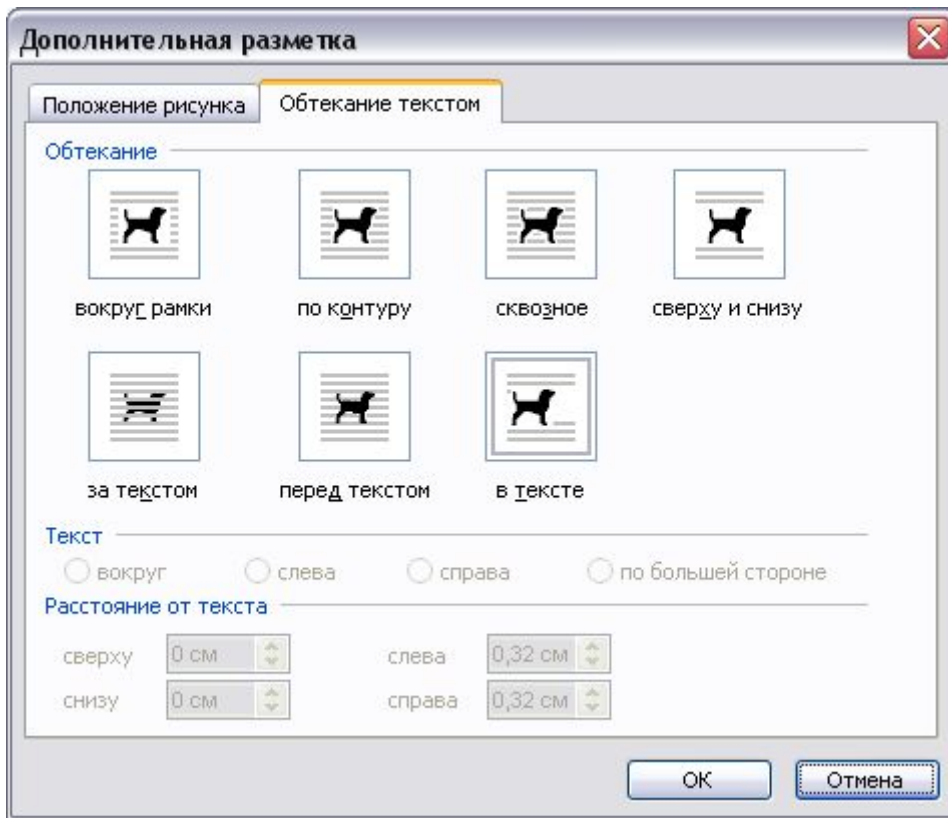


Рис. 8.18 – Додаткові види обтікання малюнка текстом

Для зміни виду обтікання малюнка текстом можна використати кнопку "Меню Обтекание текстом" на панелі "Настройка изображения" (рис. 8.19).

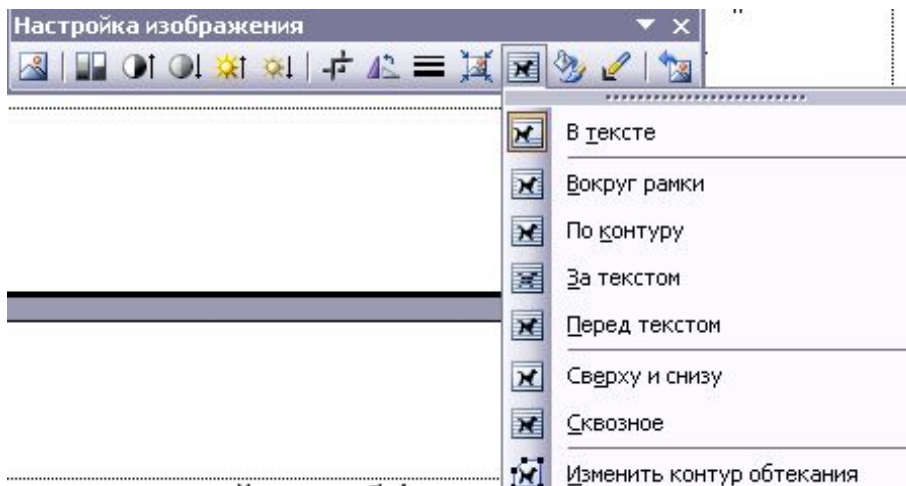


Рис. 8.19 – Меню "Обтекание текстом" на панелі "Настройка изображения"

### 8.1.33 Малювання об'єктів

Підключити панель інструментів "Рисование" (якщо вона не підключена). Вибрати *Тип лінії* або *Фігуру* на панелі інструментів і потім малювати її в потрібному місці, натиснувши кнопку миші на початку об'єкта і, розтягнувши об'єкт до потрібного розміру, відпустити кнопку миші.

Маються наступні кнопки малювання (рис. 8.20):

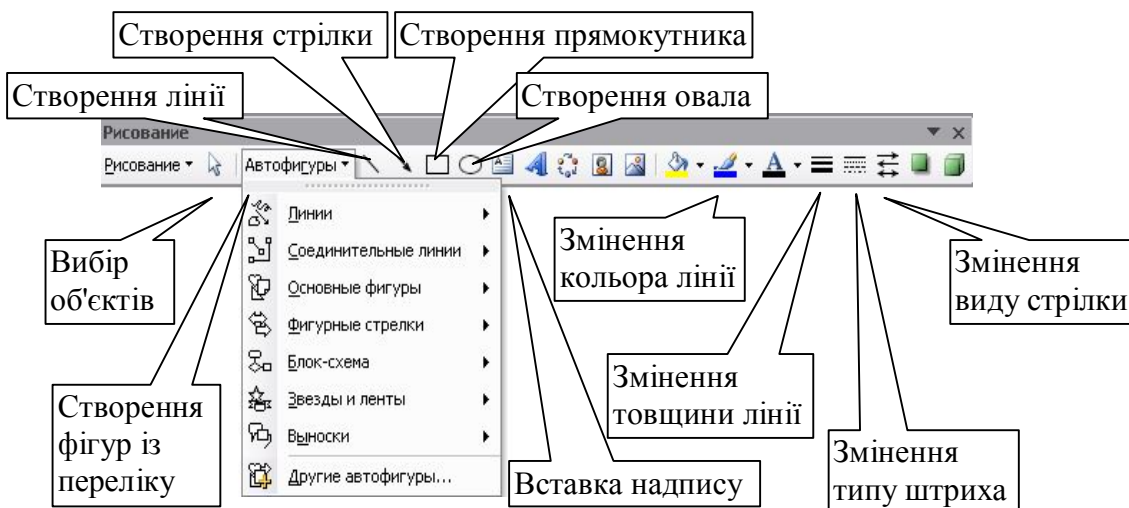



Рис. 8.20 – Кнопки панелі інструментів "Рисование"

### 8.1.34 Групування фігур

Щоб всі окремі фігури сприймалися, як єдиний малюнок, його елементи можна виділити і згрупувати.

Для виділення графічних елементів необхідно виконати такі дії:

1 спосіб. На панелі інструментів **"Рисование"** клацнути по кнопці  , підвести курсор миші в лівий верхній кут зони виділення об'єктів, натиснути кнопку миші і, не відпускаючи її, захопити в прямокутник усі потрібні об'єкти, відпустити кнопку миші. При цьому всі об'єкти усередині прямокутника виявляться виділеними.

2 спосіб. Клацнути по першому потрібному графічному елементу, а потім клацати по інших мишею при натиснутій клавіші **Shift**

Для групування виділених графічних елементів в один об'єкт необхідно виконати такі дії:


1 спосіб. Клацнути по кнопці **Рисование** на панелі інструментів **"Рисование"** і вибрати пункт **"Группировка"**.

2 спосіб. Праве клацання на одній з виділених фігур → **"Группировка"** → **"Группировать"**.

Щоб змінити один з об'єктів об'єднаного малюнка, малюнок варто розгрупувати в такий спосіб: кнопка **Рисование** на панелі інструментів **"Рисование"** → **"Разгруппировать"** → клацнути мишею на вільному місці → клацнути мишею на змінюваному об'єкті → змінити об'єкт.

### 8.1.35 Напис усередині фігури

Щоб зробити напис усередині фігури необхідно виконати такі дії:

1. Виділити фігуру.
2. Клацнути мишею по кнопці  на панелі **"Рисование"**, а потім усередині фігури або клацнути правою кнопкою миші на фігурі і вибрати пункт **"Добавить текст"**
3. Набрати напис.
4. Клацнути мишею поза фігурою.

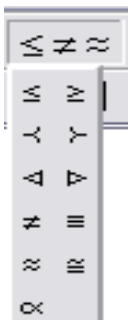
### 8.1.36 Побудова формул

Щоб побудувати формулу необхідно виконати такі дії:

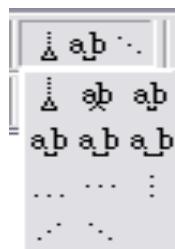
1. Помістити курсор у місце вставки формули;
2. Виконати команди: **"Вставка"** → **"Объект"**;
3. Вибрати тип об'єкта **"Microsoft Equation 3.0"**;
4. З'явиться вікно майстра формул з панеллю інструментів. Верхній ряд панелі **"Формула"** містить символи, яких немає на клавіатурі. Нижній ряд містить структури формул.



де



- "Символы отношений",



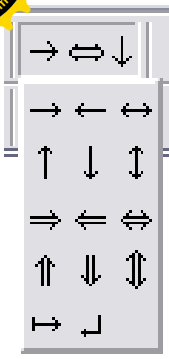
- "Пробелы и многоточия",



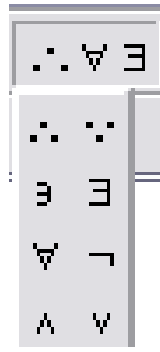
- "Надстрочные знаки",



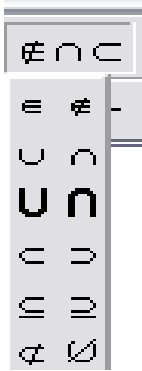
- "Операторы",



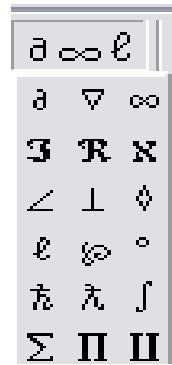
- "Стрелки",



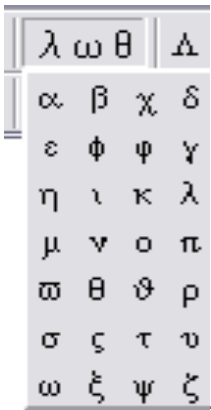
- "Логические символы",



- "Символы теории множеств",



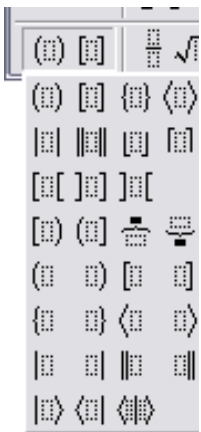
- "Разные символы",



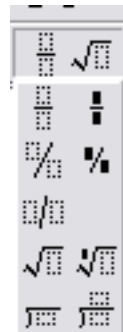
- "Греч. буквы(строчн.)",



- "Греческие буквы(прописн.)",

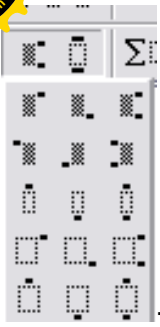


- "Шаблоны скобок",

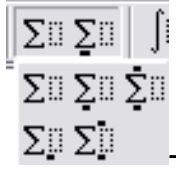


- "Шаблоны дробей и радикалов",

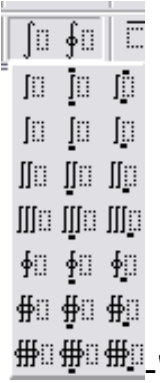




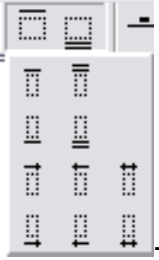
"Шаблоны верхних и нижних индексов",



"Шаблоны сумм"



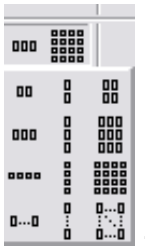
"Шаблоны интегралов",



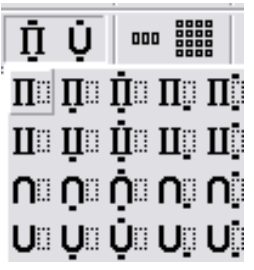
"Шаблоны надчеркивания и подчеркивания"



"Шаблоны стрелок с текстом",



"Шаблоны матриц",



"Шаблоны произведений и символов теории множеств".

- 5. Набрати формулу;
- 6. Клацнути в будь-якому місці за межами формули.

### 8.1.37 Редагування формул

Клацнути двічі по формулі, викликається майстер формул і поміщає туди дану формулу. Тепер формулу можна відредагувати, установлюючи курсор у потрібну частину структури. Після коректування клацнути мишею в будь-якому місці поза формулою, щоб повернутися в редактор Word.

Щоб збільшити розмір формули, клацнути мишею по формулі і схопити за кожною з квадратиків на її границі і розтягти неї в потрібному напрямку.

## 8.2 Електронні таблиці EXCEL

Microsoft Excel з'явився в 1985р. і одержав визнання як одна з наймогутніших і популярніших електронних таблиць.

### 8.2.1 Завантаження електронних таблиць EXCEL

Для завантаження електронних таблиць EXCEL виконати команди:

**"Пуск" → "Програми" → "Microsoft Excel".**

При цьому запуститься на виконання основний файл *Excel.exe* і буде створена книга з ім'ям *"Книга1"*, яка має розширення *.xls*.

### 8.2.2 Вікно робочої книги

Вікно робочої книги має вигляд, показаний на рис. 8.21.

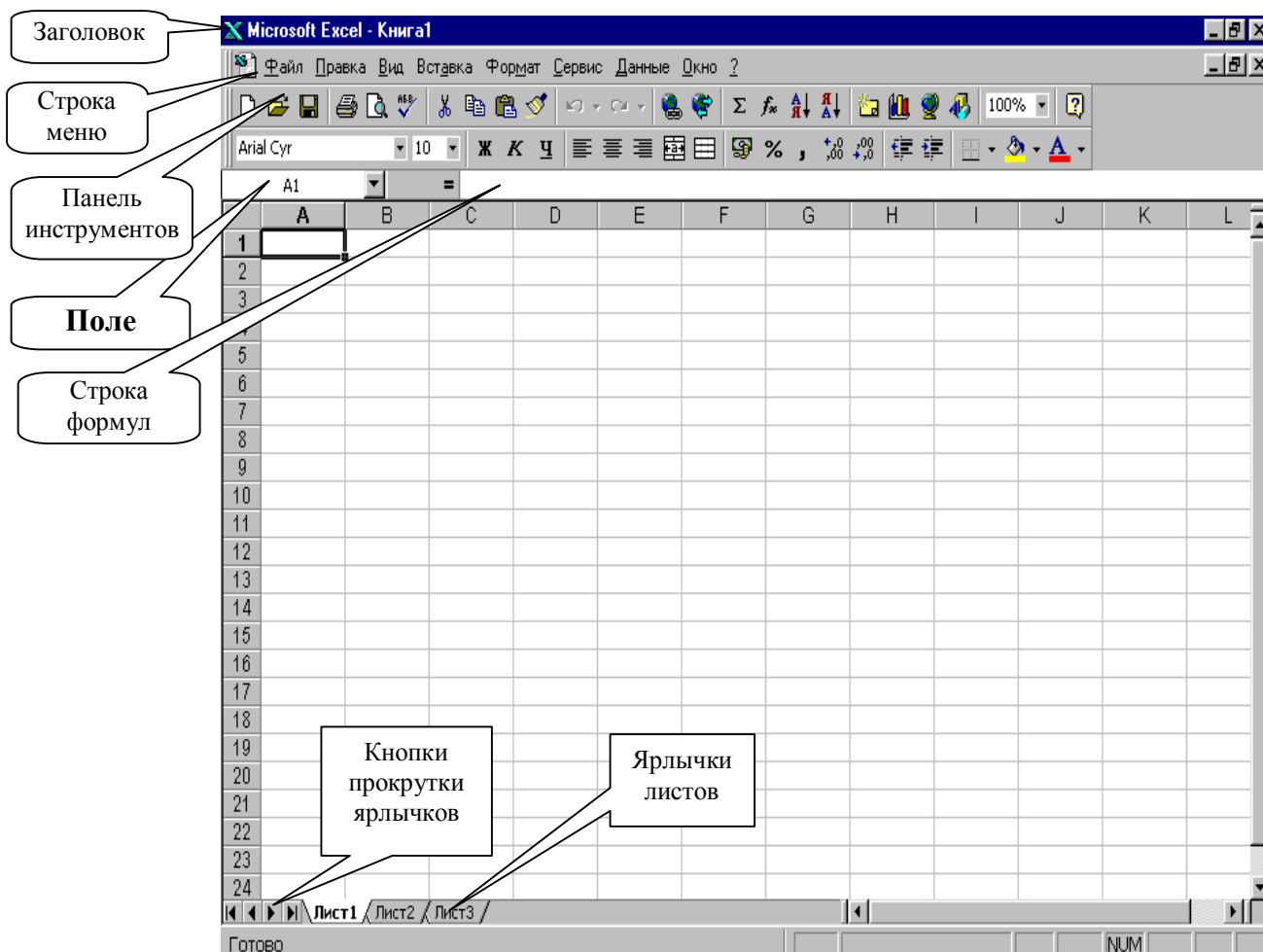
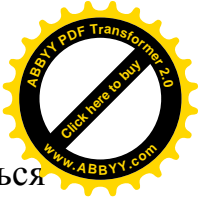


Рис. 8.21 – Вікно робочої книги

Кожна книга Excel містить декілька листів (Лист1, Лист2 і т.д.).

Робчі листи розбиті на рядки і стовпці. Рядки нумеруються від 1 до 16384, стовпці - від A до IV (від A до Z, AA до AZ, BA до BZ...) - всього 256 стовпців.



Перетинаючись, рядки і стовпці утворюють комірки, в яких зберігаються дані. Кожна комірка має однозначні координати, звані адресою.

Адреса комірки утворюється з назви стовпця і номера рядка, на перетині яких знаходиться осередок: наприклад, **A1; C6; B12**. На листі більше за 4 млн. комірок.

При використуванні діапазону комірок їх адреси записуються через двокрапку (:).

У будь-який момент часу активної може бути тільки одна робоча книга, один лист і одна комірка.

Ім'я активної книги розташоване в заголовку вікна.

Ярлик активного листу виділений білим кольором.

Активна (або поточна) комірка обрамлена чорною рамкою, а адреса активної комірки виводиться в полі імені.

Щоб зробити активною іншу книгу, необхідно в меню **"Окно"** вибрати ім'я потрібної книги в списку відкритих книг, що з'явився

Щоб активізувати інший лист, необхідно клацнути лівою кнопкою миші по ярлику потрібного листу, що знаходиться внизу екрану.

Щоб активізувати іншу комірку, необхідно клацнути лівою кнопкою по цій комірці.

### 8.2.3 Перейменування листа

Для перейменування листа виконати команди:

Праве клацання миші по ярличку листа → **"Переименовать"** → ввести нове ім'я і натиснути **ENTER**.

### 8.2.4 Вставка нового листа

Для перейменування листа виконати команди:

**"Вставка"** → **"Лист"**

При цьому новий лист вставляється перед активним листом книги.

### 8.2.5 Видалення листа

Для видалення листа виконати команди:

1 спосіб: Праве клацання по ярличку листа → **"Удалить"**

2 спосіб: **"Правка"** → **"Удалить лист"**

### 8.2.6 Переміщення листа

Для переміщення листа необхідно схопити мишею ярлик листу і перетягнути його в потрібному напрямі.

### 8.2.7 Переміщення по листах книги

Для переміщення по листах книги треба клацати по кнопках (рис. 8.22):

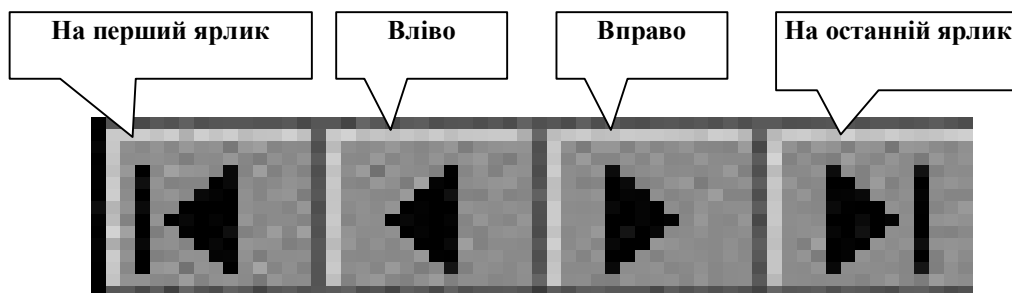


Рис. 8.22 – Кнопки прокрутки ярликів

### 8.2.8 Переміщення по листу книги

По листу книги можна переміщуватися за допомогою клавіатури, як показано в таблиці 8.3:

Таблиця 8.3 – Переміщення по листу книги

Кнопки	Результат
← → ↑ ↓	На одну комірку вліво, вправо, вгору, вниз
ENTER	На одну комірку вниз
TAB	На одну комірку вправо
SHIFT + ENTER	На одну комірку вгору
SHIFT + TAB	На одну комірку вліво
HOME	У першу комірку рядка
CTRL + HOME	У комірку <i>A1</i>
PgUp, PgDn	На екранну сторінку вгору, вниз
Alt+PgUp (Alt+PgDn)	На ширину екрану вліво (вправо)
CTRL+Backspace	До активної комірки

### 8.2.9 Рядок формул

Рядок формул служить для відображення вмісту активної комірки, а також для введення і редагування в ній даних.

Для підключення або відключення рядку формул необхідно виконати команди:

**"Вид" → "Строка формул"**

При установці курсору в рядку формул в нім з'являються три кнопки:


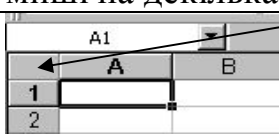
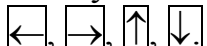
Кнопка введення (аналогічна кнопці **ENTER**)

Кнопка відміни (аналогічна кнопці **ESC**)

Кнопка виклику майстра функцій. При натисненні з'являється вікно майстра функцій, яке дозволяє вставити в активну комірку будь-яку вбудовану функцію листа.

### 8.2.10 Виділення комірок

Таблиця 8.4 – Дії, потрібні для виділення комірок

№ п/п	Що виділити?	Потрібні дії
1.	Виділити одну комірку	Клацнути лівою кнопкою миші. Комірка виділиться чорною рамкою.
2.	Виділити діапазон комірок	<i>1 спосіб:</i> Встановити у верхній лівій комірці діапазону покажчик миші і коли він прийме вигляд  , натиснути ліву кнопку миші і протягнути вправо і вниз до останньої комірки, відпустити кнопку миші. <i>2 спосіб:</i> Виділити першу комірку. При натиснутій клавіші <b>SHIFT</b> клацнути по останній комірці.
3.	Виділити несуміжні діапазони комірок	Виділити перший діапазон, а всі подальші діапазони виділяти при натиснутій клавіші <b>CTRL</b> .
4.	Виділити стовпець	Клацнути мишею по заголовку стовпця в сірому полі. При виділенні декількох стовпців - протягнути курсор миші на декілька стовпців.
5.	Виділити рядок	Клацнути мишею по номеру рядка в сірому полі. При виділенні декількох рядків - протягнути курсор миші на декілька рядків.
6.	Виділити весь лист	 Клацнути по порожній кнопці, розташованій на перетині номерів рядків і стовпців у верхньому лівому кутку листа книги
7.	Розширення виділення	При натиснутій клавіші <b>CTRL</b> використовувати клавіші  .

В таблиці 8.4 наведені дії, які потрібно виконати щоб виділити окрему зону листа

**Зауваження!** У виділеному діапазоні всі комірки, окрім активної, закрашуються чорним кольором, але активна комірка завжди залишається білого кольору.

## 8.2.11 Форматування комірок

Щоб змінити формат даних в комірках, необхідно виділити комірки, що формуються, і в меню "**Формат**" вибрати пункт "**Ячейки**" або в меню об'єкту вибрати пункт "**Формат ячеек...**". З'явиться вікно з 6 вкладками: "**ЧИСЛО**", "**ВЫРАВНИВАНИЕ**", "**ШРИФТ**", "**ГРАНИЦА**", "**ВИД**", "**ЗАЩИТА**". Необхідно вибрати потрібну вкладку і здійснити необхідне форматування.

### 8.2.11.1 Вкладка "ЧИСЛО"

У вкладці "**ЧИСЛО**" можна вказати потрібний формат даних (рис. 8.23)

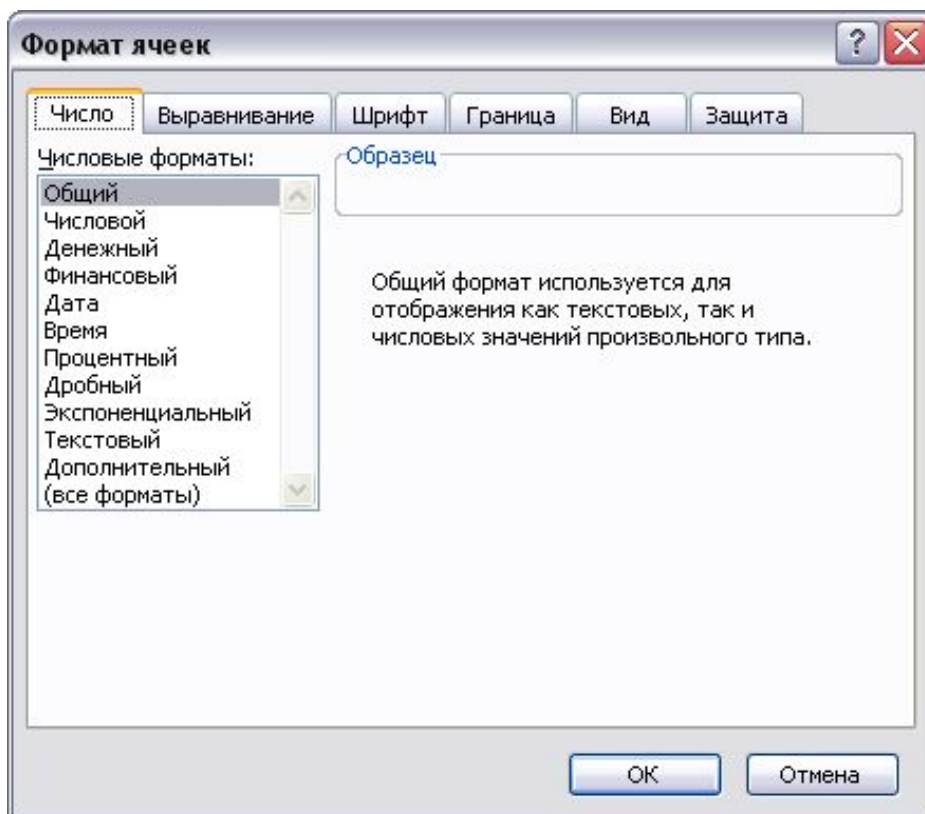


Рис. 8.23 – Вкладка "ЧИСЛО"

### 8.2.11.2 Вкладка "ВЫРАВНИВАНИЕ"

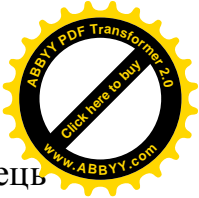
Вкладка "**ВЫРАВНИВАНИЕ**" дозволяє:

1. Змінити вирівнювання тексту в комірках по горизонталі і по вертикалі
2. Змінити орієнтацію тексту в комірках

Текст	Текст
-------	-------

3. Переносити текст в осередках на наступний рядок цієї ж комірки, якщо активізувати прапорець  "**переносить по словам**" або





розміщувати текст завжди в 1 рядок, якщо зняти прапорець "переносить по словам"

Ф.И.О. співробітника	Ф.И.О. співробітника
-------------------------	----------------------

4. Об'єднати виділені комірки, якщо активізувати прапорець  "объединение ячеек" або роз'єднати комірки, якщо прибрати прапорець "объединение ячеек"

Об'єднати виділені комірки можна також клацнувши на панелі інструментів по кнопці




5. Автоматично підбирати ширину стовпця, якщо активізувати прапорець "автоподбор ширины"

Вкладка "**ВЫРАВНИВАНИЕ**" показана на рис. 8.24.

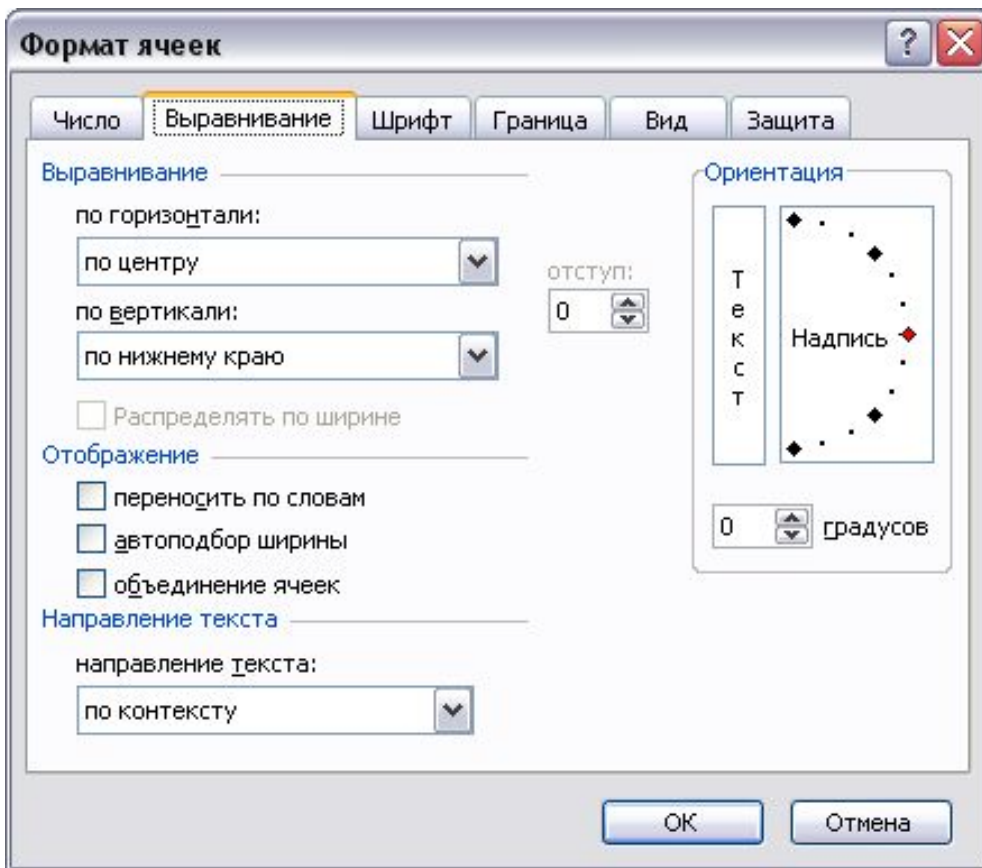


Рис. 8.24 – Вкладка "ВЫРАВНИВАНИЕ"

### 8.2.11.3 Вкладка "ШРИФТ"

Вкладка "ШРИФТ" дозволяє змінювати: Шрифт, Зображення, Розмір, Підкреслення, Колір, Ефекти (рис. 8.25).

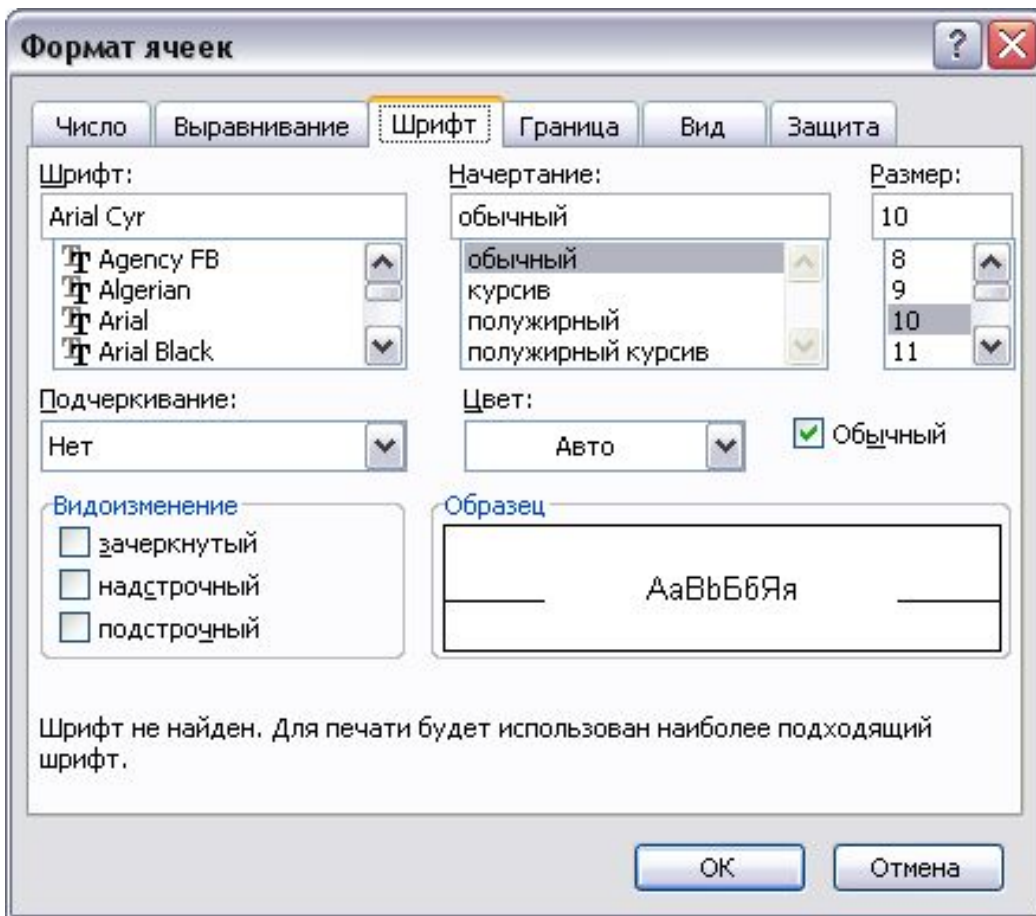



Рис. 8.25 – Вкладка "ШРИФТ"

Для зміни шрифту можна використовувати відповідні кнопки на панелі інструментів "Форматирование".

### 8.2.11.4 Вкладка "ГРАНИЦА"

У вкладці "ГРАНИЦА" можна вибрати тип і колір ліній, а також вид рамки (зовнішня, внутрішня, немає – рис. 8.26)

Якщо потрібні стандартні границі, то можна скористатися кнопкою  на панелі інструментів.

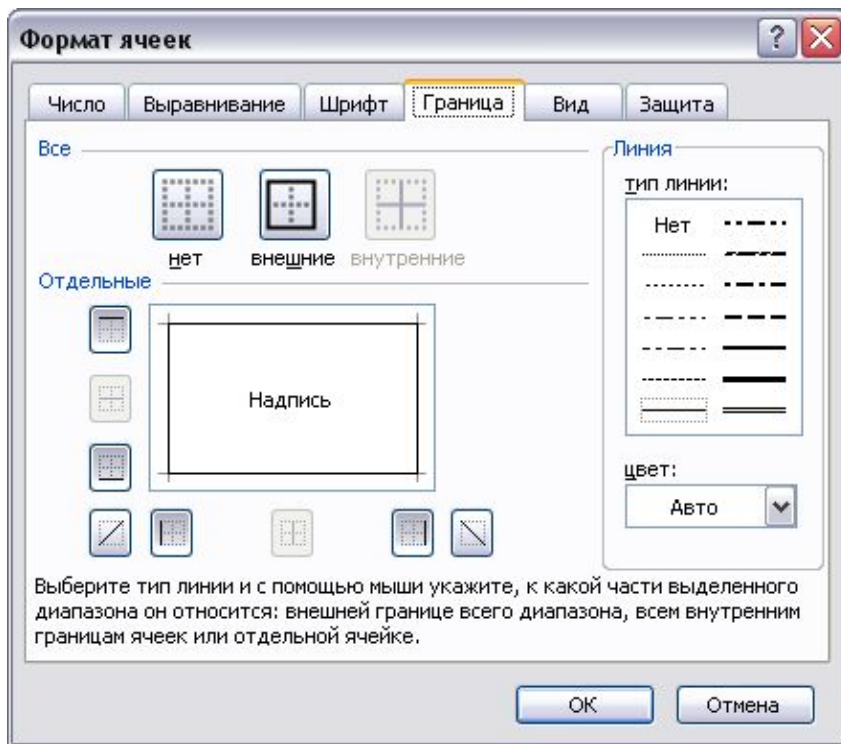



Рис. 8.26 – Вкладка "ГРАНИЦА"

### 8.2.11.5 Вкладка "ВИД"

Вкладка **"ВИД"** служит для оформления фона комірок кольором і узором (рис. 8.26). Для заливки комірок кольором і узором можна також скористатися кнопкою  на панелі інструментів.

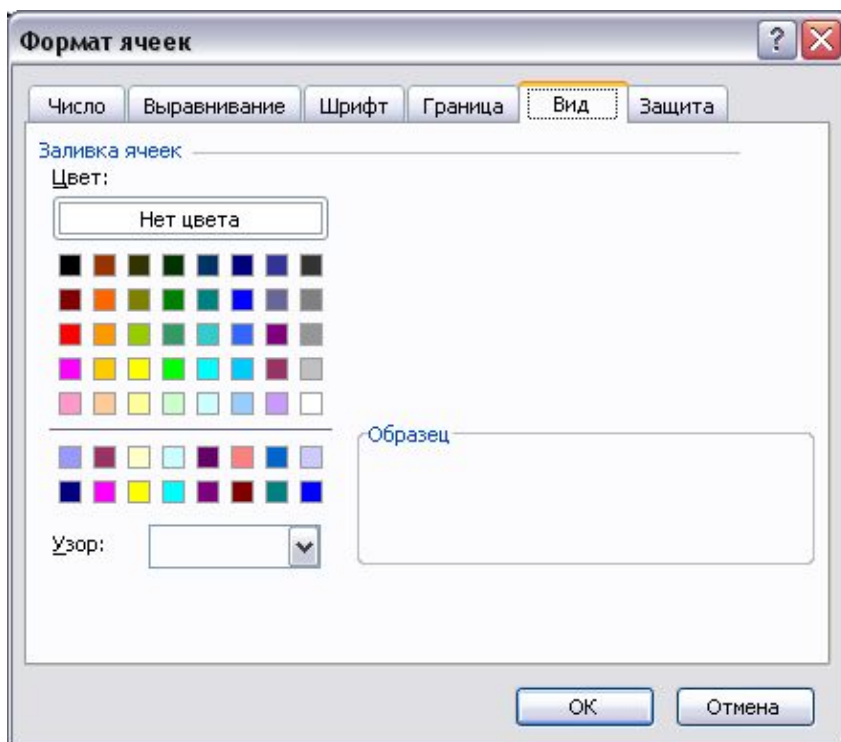
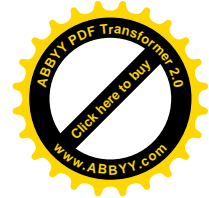


Рис. 8.27 – Вкладка "ВИД"



### 8.2.12 Зміна ширини стовпців

- 1 спосіб. Підвести курсор до правої межі заголовка стовпця у сірому полі до появи двунаправленої стрілки ( $\leftrightarrow$ ), натиснути клавішу миші і розтягнути стовпець.
- 2 спосіб. **"Формат"** → **"Столбец"** → вибрати потрібне (ширина, автопідбір ширини, приховати, показати, стандартна ширина).
- 3 спосіб. Двічі клацнути по правій межі заголовка стовпця – розширяється по щонайдовшому.

### 8.2.13 Зміна висоти рядків

- 1 спосіб. Підвести курсор до нижньої межі номера рядка у сірому полі до появи двунаправленої стрілки ( $\updownarrow$ ), натиснути клавішу миші і розтягнути рядок.
- 2 спосіб. **"Формат"** → **"Строка"** → вибрати потрібне (висота, автоподбор висоти, приховати, показати).
- 3 спосіб. Двічі клацнути по нижній межі номера рядка – розширяється по щонайширшому.

### 8.2.14 Вставка стовпців (рядків)

Виділити заголовок стовпця (номер рядка) в сірому полі, перед яким потрібно вставити новий, а потім **"Вставка"** → **"Столбец"** (**"Строку"**).

**Зауваження!** Вставляється стільки стовпців (рядків), скільки було виділено.

### 8.2.15 Видалення рядків, стовпців

Для видалення рядків або стовпців необхідно виконати дії:  
Виділити → **"Правка"** → **"Удалить"**.

**Зауваження!** Не можна видаляти несуміжні області за одну операцію.

### 8.2.16 Очищення комірок

Для очищення комірок необхідно виконати дії:

- 1 спосіб: Виділити → **"Правка"** → **"Очистить"**
- 2 спосіб: Виділити → кнопка **Del**.

### 8.2.17 Копіювання даних за допомогою Буфера обміну

1. Виділити область комірок, яку треба скопіювати → **"Правка"** → **"Копировать"** (навколо виділеної області з'явиться рухома рамка).
2. Помістити курсор у вільну комірку будь-якого листа (вона вказуватиме з якого місця починати поміщати копію) → **"Правка"** → **"Вставить"**.



## 8.2.18 Введення даних

Вводяться 2 типи даних: константи і формули.

Константи:

- 1) числові (вирівняні по правому краю комірки)
- 2) текстові значення (вирівняні по лівому краю комірки)
- 3) значення дати і часу (вирівняні по правому краю комірки)
- 4) логічні значення
- 5) помилкові значення.

### 8.2.18.1 Введення чисел

Для введення чисел використовують цифри від 0 до 9 і спеціальні символи:

"-" (мінус) використовується для вказівки від'ємного числа.

"E" використовується при введенні чисел в експоненціальному представленні: (наприклад,  $1,3E6 = 1,3 \cdot 10^6 = 1\ 300\ 000$ )

"," – ціла частина від дробової відділяється комою (наприклад, 13,25).

### 8.2.18.2 Введення довгого тексту

При введенні довгого тексту в 1 комірку, він може відображатися не повністю, якщо сусідні комірки не порожні (рис. 8.28).

	A	B	C	D	E
1	Наименование товара				
2					

	A	B	C	D	E
1	Наименование	Стоимость	1 шт.		
2					

Рис. 8.28 – Приклад відображення довгого тексту

Якщо в комірці текст не поміщається повністю, то можна використовувати декілька варіантів:

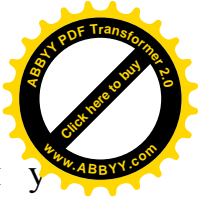
*1 спосіб.* Дозволити перенесення слів в цій комірці ("**Формат**" → "**Ячейки**" → вкладка "**ВЫРАВНИВАНИЕ**" → включити прапорець  "**переносить по словам**"). Тоді комірка збільшується автоматично по висоті на стільки, щоб помістився текст.

*2 спосіб.* Розширити межі стовпця, щоб текст помістився.

### 8.2.18.3 Введення дати і часу

При введенні дати\_як роздільник використовується крапка.

При введенні часу\_як роздільник використовується двокрапка.



Основна одиниця часу в Excel - доба. Кожному дню ставиться у відповідність послідовне число від 1 до 2 958 465. Значення 1 привласнене базовій даті 01.01.1900, 2 958 465 – 31.12.9999 року. При введенні дати Excel зберігає її у вигляді десяткового значення, яке рівне кількості днів між базовою і заданою датою. Наприклад, 01.01.96 р. - 35065. Тому з даними типу Дата і Час можна проводити ті ж операції, що і з числами.

Щоб привласнити комірці якийсь формат дати або змінити формат вже введеної дати необхідно виконати команди:

**"Формат"** → **"Ячейки"** → Вкладка **"ЧИСЛО"** → у списку **"Числові формати"** вибрати категорію **"Дата"** → в списку **"Тип"** вибрати потрібний формат:

*Д.ММ.ГГ* - 1.01.99  
*Д.МММ.ГГ* - 1 янв 99  
*Д.МММ* - 1 янв  
*МММ.ГГ* - янв 99

### 8.2.19 Редагування даних

Щоб почати редагувати вміст комірки можна:

*1 спосіб.* Виділити комірку і клацнути в рядку формул.

*2 спосіб.* Двічі клацнути по комірці до появи курсору.

**Зауваження!** Якщо другий спосіб не виходить виконати: **"Сервіс"** → **"Параметри"** → вкладка **"Правка"** → встановити прапорець  **"Правка прямо в ячейке"**

*3 спосіб.* Виділити комірку і натиснути кнопку **F2**.

Після редагування даних натиснути кнопку **ENTER**.

### 8.2.20 Сортування даних

Якщо дані таблиці необхідно упорядкувати по якій-небудь ознаці, то цю таблицю можна відсортувати. Для цього необхідно:

1. Виділити таблицю, яку треба відсортувати. (Якщо в таблиці немає об'єднаних комірок, то можна виділити лише одну комірку таблиці або виділити всю таблицю разом з рядком заголовку, а якщо в рядку заголовку таблиці є об'єднані комірки, то необхідно виділити таблицю без рядку заголовку, інакше сортування буде неможливе.)
2. Виконати команду **"Данные"** → **"Сортировка"**
3. З'явиться вікно, показане на рис. 8.29.



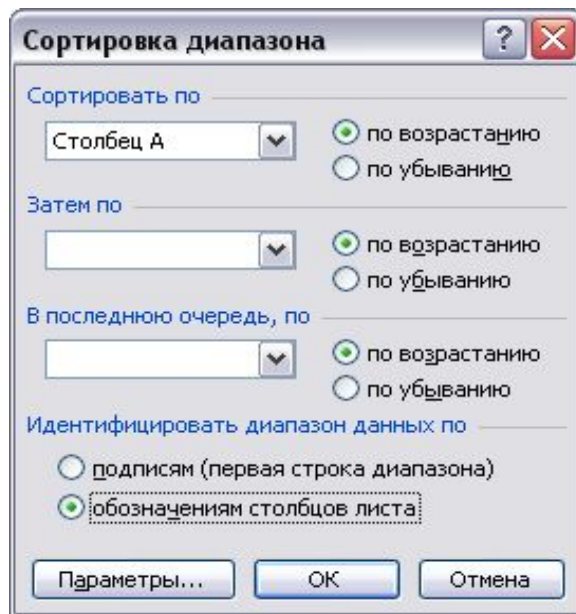


Рис. 8.29 – Вікно сортування списку

4. Вибрати в групі перемикачів "Идентифицировать диапазон данных по" перемикач "подписям" (якщо таблиця виділена з рядком заголовку) або перемикач "обозначениям столбцов листа" (якщо таблиця виділена без рядку заголовку).
5. У списку, що розкривається, "Сортировать по" вибрати стовпець, по якому здійснюється сортування
6. Активізувати перемикач "по возрастанию" або "по убыванию"
7. Підтвердити **ОК**.

Можна здійснювати складне сортування, тобто по декількох стовпцях. Для цього слід заповнити запити "Затем по" і "В последнюю очередь по".

**Зауваження!** Якщо виділити тільки один, два і т.д., але не всі стовпці таблиці, то відсортуються рядки тільки виділених стовпців і дані можуть перемішатися.

### 8.2.21 Введення формул

Всі формули в EXCEL обов'язково починаються із знаку =.  
Для введення формули необхідно:

1. встановити курсор в потрібну комірку,
2. набрати з клавіатура знак =,
3. набрати формулу,
4. натиснути **ENTER**.

У формулах можуть використовуватися:

- 1) знаки арифметичних операцій (+, -, \*, /, ^);



- 2) стандартні функції (*sin, cos, СУММ...*);
- 3) константи;
- 4) посилання на комірки;
- 5) круглі дужки для зміни порядку дій.

EXCEL відобразить результат обчислень у виділеній комірці, а саму формулу – в рядку формул.

Одне з основних достоїнств EXCEL – використання посилань (тобто адрес комірок) у формулі.

Значення формули залежить від вмісту комірок, на які вказують посилання.

Щоб створити у формулі посилання на осередок, можна:

- 1 *спосіб*. Використовувати мишу: клацнути по комірці, посилання на яку вставляється у формулу. Навколо комірки з'являється рухома рамка і посилання на комірку з'являється у формулі. Після закінчення введення -  або  в рядку формул.
- 2 *спосіб*. В процесі введення формули ввести посилання на комірки з клавіатури латинськими символами.

### 8.2.22 Відображення формул в комірках листа

Для відображення формул в комірках листа необхідно виконати команди:

"Сервис" → "Параметры" → вкладка "Вид" → в групі "Параметры окна" встановити прапорець  "формулы"

### 8.2.23 Відносні, абсолютні і змішані посилання

По написанню абсолютне посилання відрізняється від відносного тим, що перед номером рядка і номером стовпця комірки ставиться знак \$.

- =A4 - відносне посилання
- =\$A\$4 - абсолютне посилання
- =\$A4 і
- =A\$4 - змішані посилання

#### 8.2.23.1 Зміна типу посилання:

Для зміни типу посилання в формулі необхідно виконати дії:

- 1 *спосіб*. Виділити в формулі посилання на комірку і натиснути . При цьому відносне посилання зміниться на абсолютне. Повторні натиснення  - зміна типу посилання від абсолютного - до змішаного, а потім – до відносного.
- 2 *спосіб*. Вручну дописати або прибрати знак \$.



### 8.2.23.2 Різниця абсолютного і відносного посилань

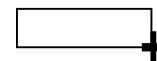
Різниця абсолютного і відносного посилань виявляється тільки при копіюванні формули, що містить посилання:

1. При копіюванні абсолютного посилання в новій формулі її координати не змінюються, тобто не настроюються.
2. При копіюванні відносного посилання **вниз** або **вгору** в новій формулі автоматично настроюється координата рядка, змінюючись на стільки, на скільки рядків змістилася формула щодо колишнього розміщення.
3. При копіюванні відносного посилання **управо** або **вліво** в новій формулі автоматично настроюється координата стовпця, змінюючись на стільки, на скільки стовпців змістилася формула щодо колишнього розміщення.

### 8.2.24 Копіювання формули

Щоб скопіювати (розтягнути) формулу з даної комірки в сусідні комірки, потрібно виділити комірку з формулою, схопити її за маркер **заповнення** комірки, розташований в нижньому правому кутку комірки, коли покажчик миші прийме вигляд +, не відпускаючи протягнути в сусідні комірки.

Приклад.

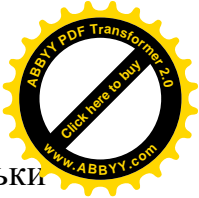
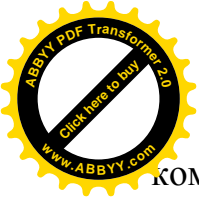


Відома кількість відпрацьованих годин і вартість 1 години роботи для декількох робітників. Знайти заробіток кожного робітника і його частку в загальному заробітку (рис. 8.30).

	A	B	C	D	E
1	П.І.Б.	Вартість 1ч. роботи	Кількість годин	Сума до виплати	Частка від загальної суми
2	Іванов	5	10	=B2*C2	=D2/\$D\$5
3	Петров	7	4	=B3*C3	=D3/\$D\$5
4	Сидоров	6	5	=B4*C4	=D4/\$D\$5
5	Разом:			=СУММ(D2:D4)	

Рис. 8.30 – Фрагмент таблиці з даними і розрахунками в Excel

Щоб визначити заробіток Іванова в комірку **D2** необхідно записати формулу: **=B2\*C2**, а потім скопіювати (розтягнути) цю формулу з комірки **D2** в



комірки **D3:D4**, щоб визначити заробіток решти робітників. Оскільки посилання було відносне, то формули в нижніх комірках настрояться.

Для знаходження загального заробітку в комірці **D5** необхідно записати формулу: **=СУММ(D2:D4)**, а для знаходження частки Іванова в загальному заробітку в комірці **E2** – формулу: **=D2/D5**.

Щоб взяти частку решти робітників в загальному заробітку, необхідно скопіювати (розтягнути) формулу з комірки **E2** в комірки **E3:E4**, але попередньо відносне посилання на комірці **D5** треба замінити абсолютним посиланням (захистити \$), оскільки це посилання у формулі не потрібно настроювати при копіюванні формули, а потім вже виділити комірці **E2** з формулою **=D2/\$D\$5** і розтягнути формулу вниз за маркер заповнення.

### 8.2.25 Введення даних за допомогою списків

Іноді в рядку або в стовпці потрібно помістити дані якого-небудь списку (наприклад, *Пн, Вт, Ср...* або *Янв, Фев, ...*), тоді ці дані можна ввести таким чином:

1. Ввести в комірці перше значення із списку,
2. Схопити це дане за маркер заповнення і протягнути в сусідні комірки.

### 8.2.26 Введення даних за допомогою рядів даних

Іноді в рядку або в стовпці потрібно помістити числові дані, що змінюються з певним кроком, тоді ці дані можна ввести таким чином:

1. Ввести два перші значення ряду в сусідні комірки,
2. Виділити ці 2 сусідні комірки, схопити їх за маркер заповнення другої комірки і протягнути в сусідні комірки.

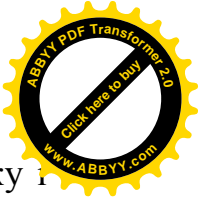
Якщо в рядку або в стовпці потрібно помістити числові дані, що змінюються з кроком, рівним 1, тоді ці дані можна ввести таким чином:

1. Ввести в комірці перше значення,
2. Схопити це дане за маркер заповнення і при натиснутій клавіші **Ctrl** протягнути в сусідні комірки.

### 8.2.27 Посилання на комірки іншого листа тієї ж книги

Посилання на комірки іншого листа складаються з назви листа і адреси комірки, розділені знаком оклику (!) і створюються 2 способами:

*1 спосіб.* З клавіатури ввести знак =, клацнути мишею на ярличку листа, посилання на комірці якого створюється, а потім клацнути на потрібній комірці і натиснути **ENTER** (або ввести будь-який арифметичний знак: +, -, \*, /, ^ якщо необхідно продовжити набір формули).



2 спосіб. З клавіатури ввести знак =, а потім назва листа, знак оклику і адресу комірки.

*Наприклад*, щоб в комірці *A1* листа з ім'ям "*Лист2*" послатися на комірку *A6* з листа з ім'ям "*Лист1*", в ній необхідно отримати формулу: *=Лист1!A6*.

1 спосіб. Виділити комірку *A1* на листі з ім'ям "*Лист2*", ввести знак =, клацнути мишею на ярличку листа з ім'ям "*Лист1*", а потім на комірці *A6* і натиснути **ENTER**.

2 спосіб. У комірці *A1* на листі з ім'ям "*Лист2*" з клавіатури набрати формулу *=Лист1!A6*.

### 8.2.28 Копіювання таблиці на інший лист за допомогою посилань на комірки

Якщо з однією і тією ж таблицею на різних листах потрібно здійснити різні операції, то зручно копіювати початкову таблицю на інші листи за допомогою посилань на комірки. Тоді будь-яка зміна даних в початковій таблиці викличе зміну даних на інших листах.

Щоб створити копію таблиці на іншому листі за допомогою посилань на комірки, необхідно виконати такі дії:

1. Встановити курсор в комірку листа, де треба отримати копію (*наприклад, A1*),
2. Ввести знак =,
3. Клацнути мишею по ярличку листа з початковою таблицею (*наприклад, Лист1*), а потім по комірці, в якій знаходиться верхній лівий кут початкової таблиці (*наприклад, A1*)
4. Перевірити правильність посилання. Повинно бути посилання тільки на одну комірку, а не на діапазон (*наприклад, =Лист1!A1*) і натиснути **ENTER**.
5. Одержану формулу схопити за маркер заповнення і протягнути вправо на стільки стовпців, скільки було в початковій таблиці.
6. Одержаний рядок з формулами схопити за маркер заповнення останньої комірки і протягнути вниз на стільки рядків, скільки було в початковій таблиці.

### 8.2.29 Копіювання таблиці за допомогою команди "Спеціальная вставка"

Іноді необхідно скопіювати значення без формул, за допомогою яких ці значення набулі. А іноді потрібно скопіювати тільки формат. В цьому випадку можна використовувати команду "*Спеціальная вставка*"

Для копіювання таблиці за допомогою команди "*Спеціальная вставка*" необхідно:

1. Виділити таблицю, яку треба скопіювати, і скопіювати її в Буфер обміну ("**Правка**" → "**Копировать**")
2. Встановити курсор в місце для копіювання → "**Правка**" → "**Специальная вставка**"
3. У вікні (рис. 8.31), що з'явиться, встановити перемикач "**значения**" або "**формат**" і натиснути **ОК**.

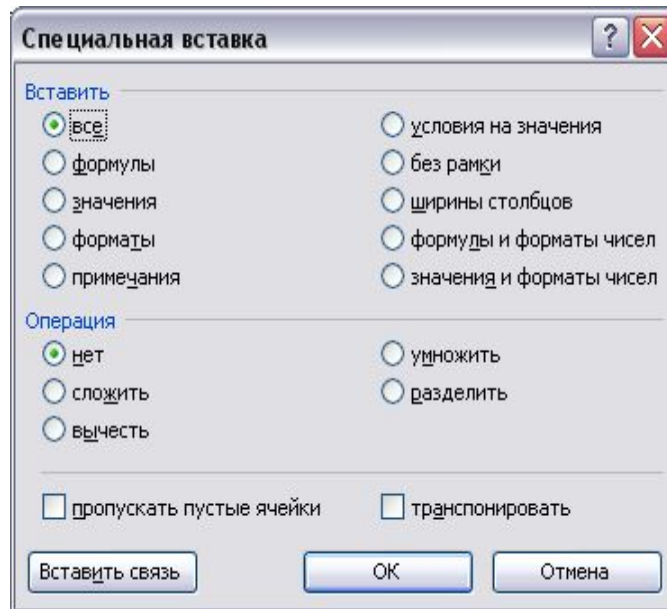


Рис. 8.31 – Вікно команди "Специальная вставка"

При цьому скопіюються тільки значення без формул, за допомогою яких вони отримані, або тільки формат.

### 8.2.30 Використовування Майстра функцій

Щоб використовувати функцію, потрібно ввести її як частину формули в комірку. Значення, які використовуються для обчислення функцій, називають аргументом, а значення, які функція обчислює - результатом.

**Майстер функцій** дозволяє автоматизувати процес використання вбудованих функцій при введенні формул (функцій більше ніж 400).

Для створення формули з використанням **Майстра функцій** необхідно:

1. Встановити курсор в комірку, де потрібно отримати результат,
2. Викликати Майстер функцій: "**Вставка**" → "**Мастер функций**"

або клацнути по кнопці 

3. З'явиться діалогове вікно, в якому спочатку необхідно мишею вибрати Категорію функцій (Повний алфавітний перелік, Математичні, Статистичні, Дата і час...), а потім вибрати потрібну функцію обраної категорії (рис. 8.32)
4. Натиснути **ОК**.



5. З'явиться діалогове вікно для заповнення аргументів функції (рис. 8.33). Аргументи функції краще вводити за допомогою миші. При цьому діалогове вікно можна переміщати або згортати за допомогою маленьких червоних стрілок. Якщо як аргумент вручну вводити діапазон комірок, то адреси 1-ї і останньої комірок діапазону потрібно записати через двокрапку (:). Якщо необхідно використовувати абсолютне посилання на ці комірки, то необхідно натиснути **F4**.
6. Натиснути **OK**.

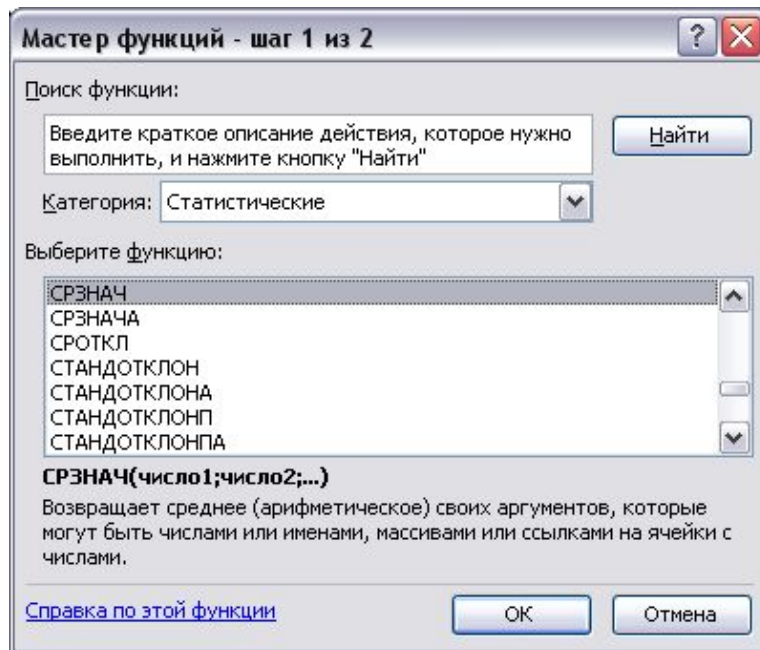


Рис. 8.32 – Вікно Майстра функцій

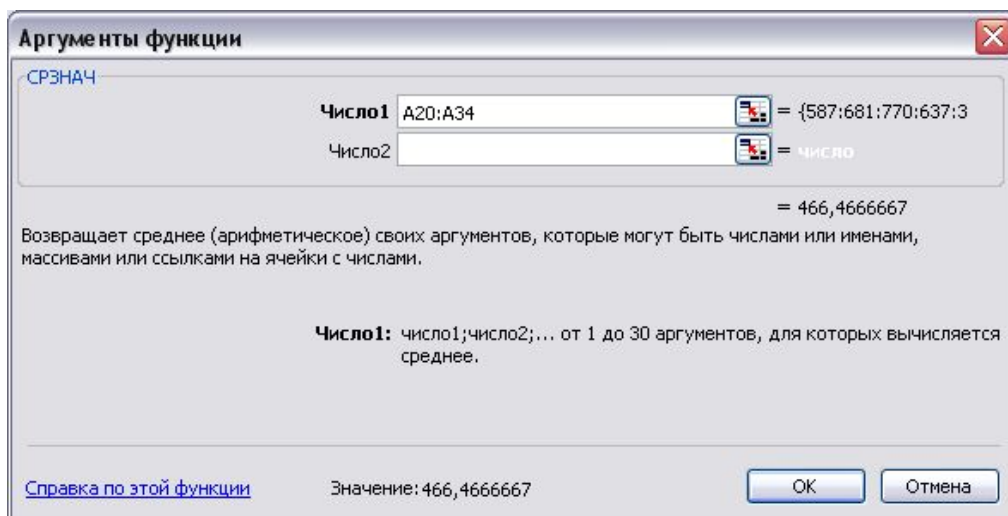


Рис. 8.33 – Вікно вводу аргументів функції



### 8.2.31 Автопідсумовування

Для використання авто підсумовування необхідно:

1. Встановити курсор в комірку, де потрібно отримати суму.
2. Клацнути по кнопці  $\Sigma$  (автосума) на панелі інструментів.
3. Якщо потрібно змінити мишею діапазон підсумовування і натиснути **ENTER**.

### 8.2.32 Математичні функції

**Зауваження!** Аргументи, які вказані в квадратних дужках [...], являються необов'язковими і їх можна не вказувати.

**СУММ("Числа1" [; "Числа2";... ])** – знаходить суму аргументів;

**ПРОИЗВЕД("Числа1" [; "Числа2";... ])** – знаходить добуток аргументів;

**ПИ()** – повертає значення  $\pi \approx 3,14159\dots$ ;

**LOG10("Число")** – знаходить десятковий логарифм аргументу;

**LOG("Число"; "Основа")** – знаходить логарифм заданого числа по заданій основі;

**LN("Число")** – знаходить натуральний логарифм аргументу;

**EXP("Число")** – зводить число  $e \approx 2,718\dots$  у ступінь, рівний аргументу;

**ABS("Число")** – знаходить модуль аргументу;

**Приклад** Обчислити  $y = e^{\left|\frac{1}{x^2}\right|} + \sin^2 x$ .

Якщо значення  $X$  розмістити в комірці **A2**, то для обчислення  $Y$  в комірці **B2** можна помістити формулу (рис. 8.34).

B2		fx =EXP(ABS(1/A2^2))+SIN(A2)^2			
	A	B	C	D	E
1	x	y			
2	-2	2,110847227			

Рис. 8.34 – Приклад обчислення математичної формули

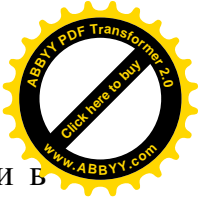
**СУММКВ("Числа1" [; "Числа2"...])** - знаходить квадрати аргументів в і повертає суму квадратів

**СУММЕСЛИ("Интервал1"; "Критерий" [; "Интервал2"])** – знаходить суму значень за умови виконання деякої умови.

Де

**"Интервал1"** – це діапазон комірок, які аналізуються,

**"Критерий"** – це критерій у формі числа, виразу або тексту, який визначає, які комірки додаються до суми.



**"Интервал2"** – це фактичний діапазон для підсумовування. Комірки в **"Интервал2"** підсумовуються, тільки якщо відповідні їм комірки в аргументі **"Интервал1"** задовольняють критерію. Якщо **"Интервал2"** відсутній, то підсумовуються комірки в аргументі **"Интервал1"**.

**СУММПРОИЗВ("Массив1"; "Массив2" [; "Массив2"...])** - перемножує відповідні елементи з кожного масиву і підсумовує одержані добутки

*Приклад.* Маються наступні дані (рис. 8.35):

	A	B	C	D	E	F
1	Найменування товару	Вартість 1 шт.	Кількість за рік			
2			1 квартал	2 квартал	3 квартал	4 квартал
3	Вилка	5,2	100	150	120	98
4	Ніж	4,5	79	65	85	73
5	Ложка	5	150	145	96	112

Рис. 8.35 – Таблиця з похідними даними

1) Визначити вартість всіх виробів за кожен квартал.

Для цього в комірку **A6** запишемо текст *"Вартість всіх виробів за квартал"*, а в комірку **C6** помістимо формулу: **=СУММПРОИЗВ(\$B\$3:\$B\$5;C3:C5)**, а потім скопіюємо цю формулу в комірки **D6:F6**.

2) Підрахувати кількість виробів, проданих в I кварталі з вартістю не меншою 5.

Для цього в комірку **A7** запишемо текст *"Кількість виробів, проданих в I кварталі з вартістю не меншою 5"*, а в комірку **C7** помістимо формулу: **=СУММЕСЛИ(B3:B5;">=5";C3:C5)**; результат – 250.

3) Підрахувати кількість виробів, проданих в I кварталі з кількістю проданого меншою 120.

Для цього в комірку **A8** запишемо текст *"Кількість виробів, проданих в I кварталі з кількістю проданого меншою 120"*, а в комірку **C8** помістимо формулу: **=СУММЕСЛИ(C3:C5;"<120")**; результат – 179.

### 8.2.33 Статистичні функції

**СРЗНАЧ("Числа1" [; "Числа2"...])** – знаходить середнє арифметичне значення своїх аргументів;

**МИН("Числа1" [; "Числа2"...])** – знаходить мінімальне значення серед аргументів;

**МАКС("Числа1" [; "Числа2"...])** – знаходить максимальне значення серед аргументів;



**СЧЕТ**("Числа1" [, "Числа2"...]) - підраховує кількість чисел в списку аргументів;

**СЧЕТЕСЛИ**("Диапазон"; "Критерий") - підраховує кількість комірок, що задовольняють критерію.

Приклад. Розглянемо дані таблиці на рис. 8.35

4) Знайти кількість найменувань товару з кількістю проданого в I кварталі більшою за 80.

Для цього в комірку **A9** запишемо текст "Кількість найменувань товару з кількістю проданого в I кварталі більшою за 80", а в комірку **C9** помістимо формулу:

=СЧЕТЕСЛИ(C3:C5;">80"); результат – 2.

5) Знайти мінімальну кількість проданого товару за кожний квартал.

Для цього в комірку **A10** запишемо текст "Мінімальна кількість за квартал", а в комірку **C10** помістимо формулу: =МИН(C3:C5); а потім скопіюємо цю формулу в комірки **D10:F10**.

### 8.2.34 Логічні функції

**ЕСЛИ**("Лог\_выражение";"Значение\_если\_истина";"Значение\_если\_ложь") – повертає одне з значень в залежності від результату виконання логічного виразу.

Три функції **И**, **ИЛИ**, **НЕ** дозволяють створювати складні логічні вирази. Вони працюють в поєднанні з простими операторами порівняння

=, <, >, <=, >=, <>

Ці функції можуть мати до 30 логічних аргументів.

**И**("Логическое\_значение1"; "Логическое\_значение2" [, ...]) – повертає значення "ИСТИНА" якщо всі аргументи приймають значення "ИСТИНА", і "ЛОЖЬ" якщо хоча б один з аргументів приймає значення "ЛОЖЬ".

**ИЛИ**("Логическое\_значение1"; "Логическое\_значение2" [, ...]) – повертає значення "ИСТИНА" якщо хоча б один з аргументів приймає значення "ИСТИНА", і "ЛОЖЬ" якщо всі аргументи приймають значення "ЛОЖЬ".

**НЕ**("Логическое\_значение") – повертає значення "ИСТИНА" якщо аргумент приймає значення "ЛОЖЬ" і навпаки.

Приклад1.

Похідні дані: **A2** – кількість пропусків, **B2** – середній бал за сесію.

Вивести текст про те, чи буде нарахована стипендія, якщо умовою для нарахування стипендії є середній бал не менший за 4 при кількості пропусків меншою за 36. Для цього створимо формулу:

=ЕСЛИ(И(A2<36;B2>4);"начислить стипендию";"не начислят стипендию")

Приклад2.

Перевіряється вміст комірки **A10** і в залежності від результату перевірки видати повідомлення:



- >100 - добре
- від 80 до 100 - нормально
- <80 – погано

Для цього створимо формулу:

$=\text{ЕСЛИ}(A10>100;"\text{добре}";\text{ЕСЛИ}(A10<80;"\text{погано}";"\text{нормально}"))$

Щоб ввести цю формулу за допомогою **Майстра функцій** необхідно:

1. Помістити курсор в комірку, відведена для результату;
2. Викликати Майстер функцій:
3. В категорії "Логические" вибрати функцію **ЕСЛИ**;
4. У вікні 1-го аргументу занести вираз  $A10>100$ ;
5. У вікні 2-го аргументу занести текст "добре";
6. Клацнути мишею у вікні 3-го аргументу;
7. Клацнути мишею по назві функції **ЕСЛИ** лівіше трьох кнопок рядка формул;
8. Відкриється нове вікно для вкладеної функції **ЕСЛИ** ;
9. У вікні 1-го аргументу занести вираз  $A10<80$ ;
10. У вікні 2-го аргументу занести текст "погано";
11. У вікні 3-го аргументу занести текст "добре";
12. Підтвердити .

## 8.2.35 Діаграми


### 8.2.35.1 Основні елементи діаграм

1. Вісь X – вісь категорій, на якій указуються категорії, до яких відносяться точки осі;
2. Вісь Y – вісь значень;
3. Ряди даних;
4. Назва діаграми;
5. Назва осі X (осі категорій);
6. Назва осі Y (осі значень);
7. Легенда, що містить зображення ряду значень і ім'я цього ряду;
8. Маркери даних, які служать для того, щоб відрізнити один ряд від іншого;
9. Лінії сітки;
10. Зарубки на осі X, які відділяють різні категорії;

### 8.2.35.2 Побудова діаграм

Діаграма може знаходитися на окремому листі або на листі з табличними даними і тоді можна змінювати розмір, межі, переміщати як будь-який графічний об'єкт.

Побудова діаграми здійснюється за декілька кроків.

1. Виділити дані, які потрібно відобразити в графічному вигляді. При цьому краще виділяти з рядком заголовку або разом зі стовпцем пояснень.
2. Викликати майстер діаграм, клацнувши по значку  на панелі інструментів або виконати команди: "**Вставка**" → "**Діаграма**".
3. З'явиться вікно першого кроку Майстра діаграм (рис. 8.36), в якому треба вибрати "**Тип**" діаграми, її "**Вид**" і клацнути по кнопці **Далее**

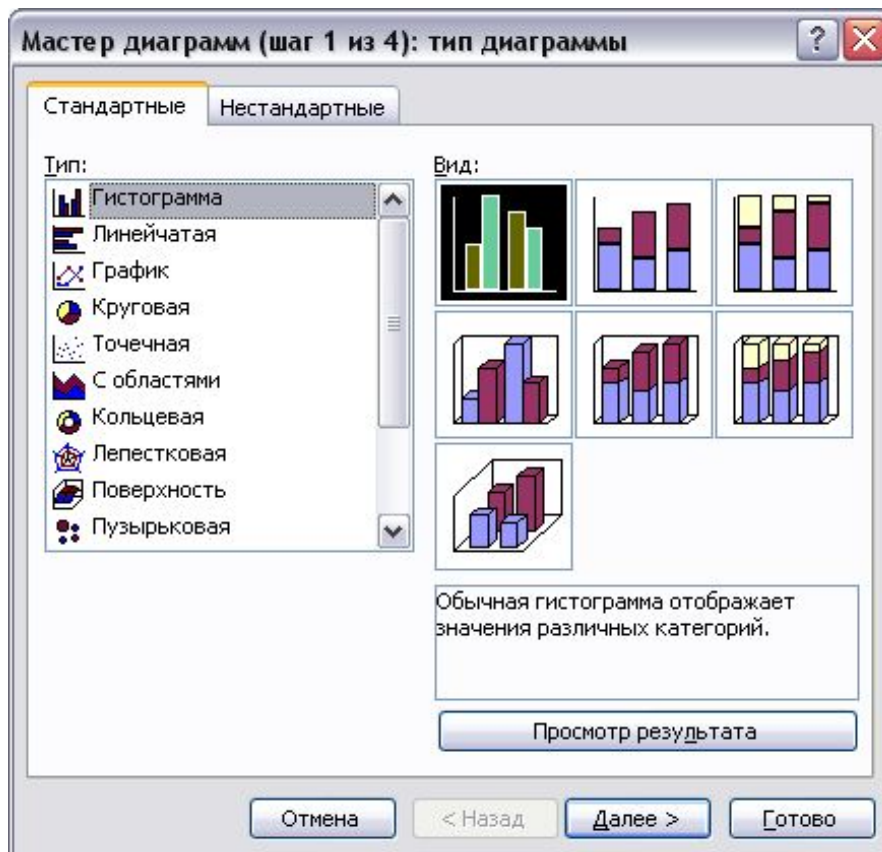


Рис. 8.36 – Вікно 1-го кроку Майстра діаграм

4. З'явиться вікно другого кроку Майстра діаграм (рис. 8.37), в якому на вкладці "**Диапазон данных**" якщо правильно виділений діапазон, то вибрати як розташовані ряди: "**в строках**" або "**в столбцах**", а потім перейти на вкладку "**Ряд**" (рис. 8.38), де можна змінити "**Имя**" і "**Значения**" для кожного ряду, а також "**Подписи оси X**" і клацнути по кнопці **Далее**. На цій вкладці можна також додавати або видаляти ряди.



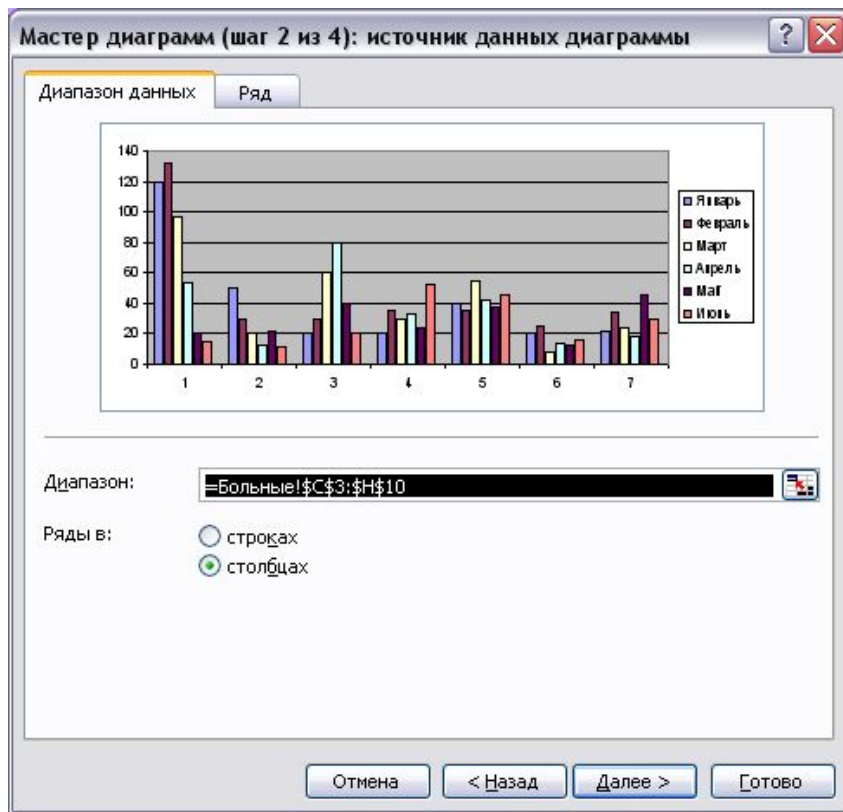


Рис. 8.37 – Вікно 2-го кроку Майстра діаграм. Вкладка "Диапазон данных"

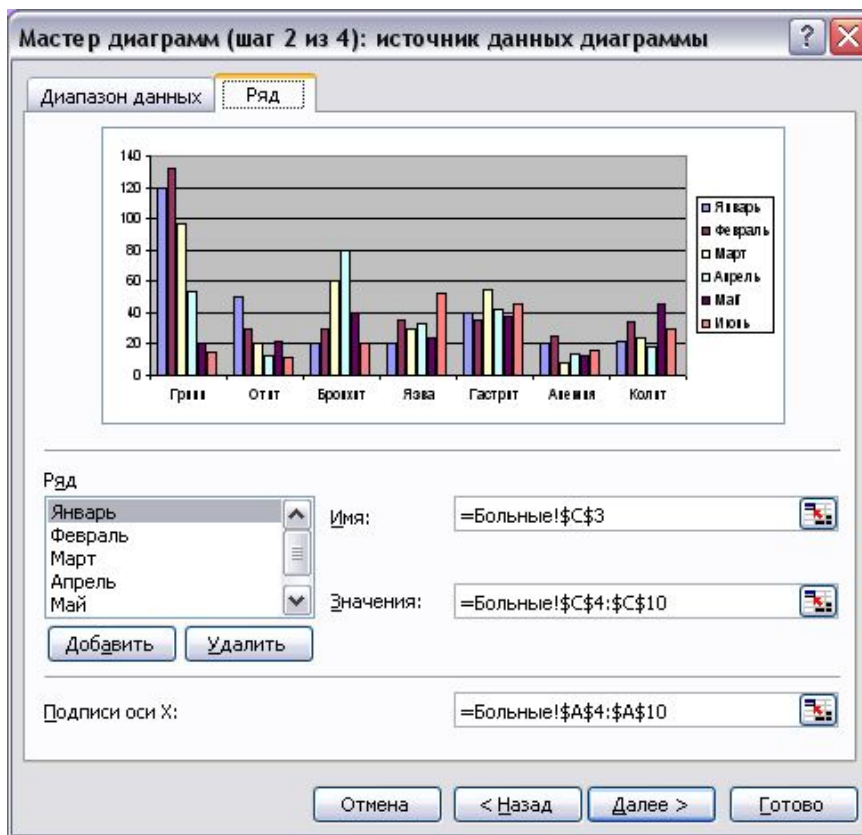


Рис. 8.38 – Вікно 2-го кроку Майстра діаграм. Вкладка "Ряд"

5. З'явиться вікно третього кроку Майстра діаграм (рис. 8.39), в якому по черзі відкривати вкладки "Заголовки", "Оси", "Линии сетки", "Легенда", "Подписи данных", "Таблица данных" (рис. 8.39 – 8.44), заповнити потрібні поля і клацнути по кнопці **Далее**

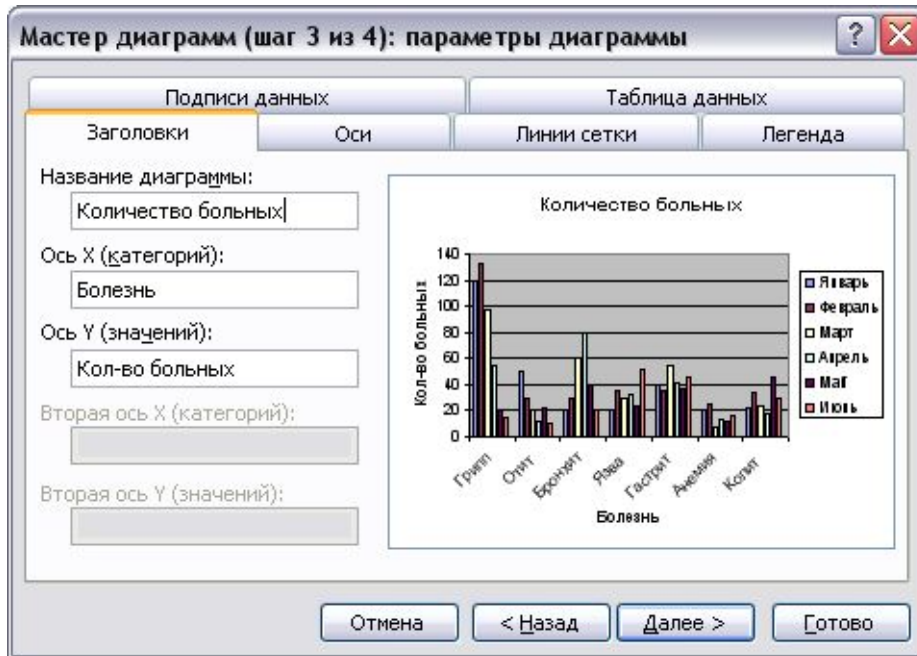


Рис. 8.39 – Вікно 3-го кроку Майстра діаграм. Вкладка "Заголовки"

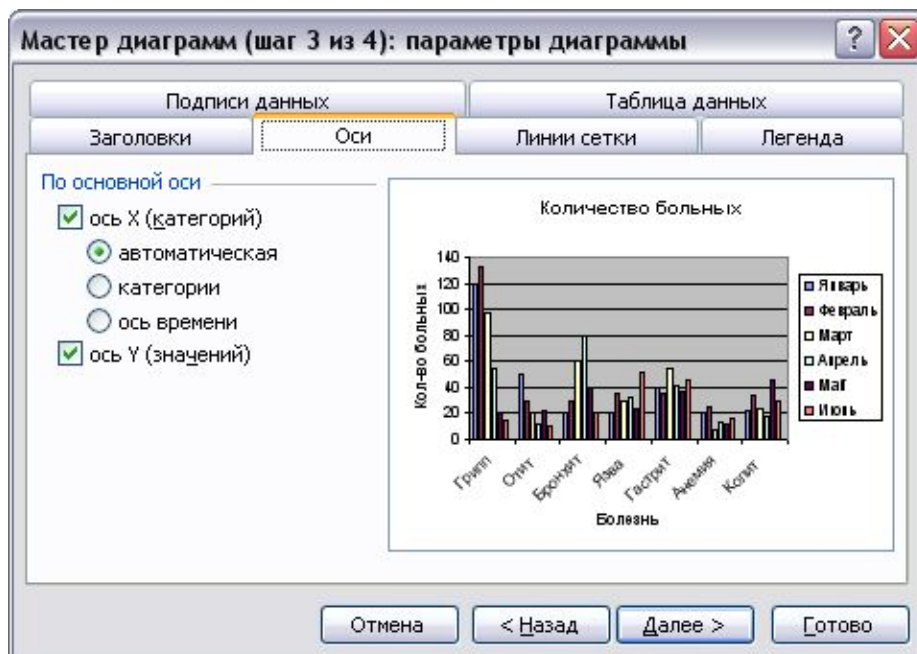


Рис. 8.40 – Вікно 3-го кроку Майстра діаграм. Вкладка "Оси"

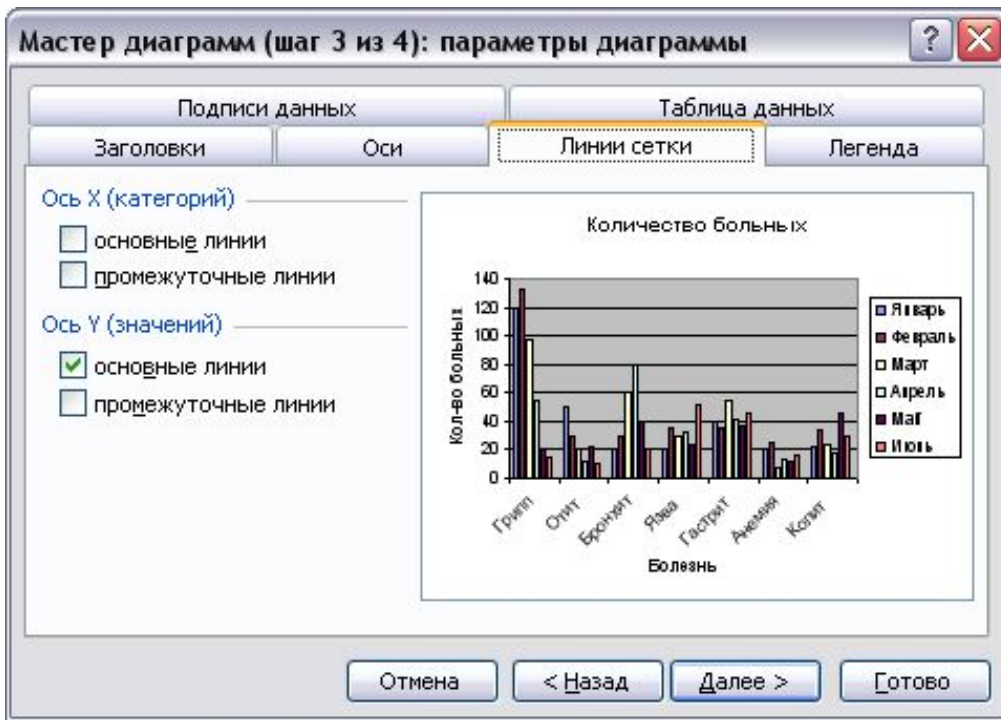


Рис. 8.41 – Вікно 3-го кроку Майстра діаграм. Вкладка "Линии сетки"

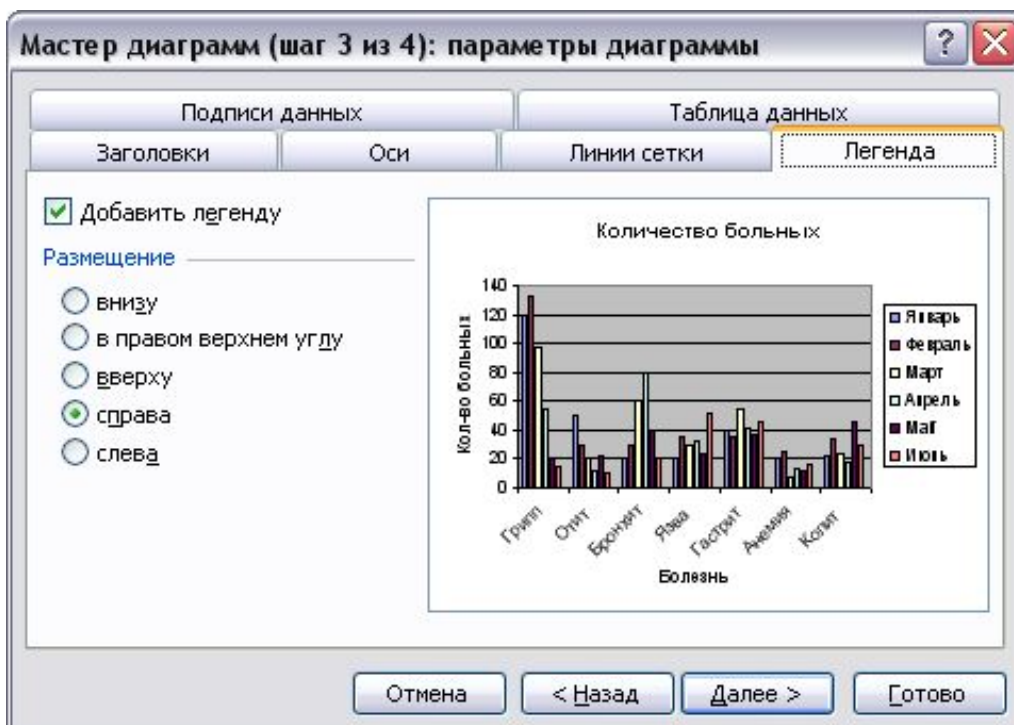


Рис. 8.42 – Вікно 3-го кроку Майстра діаграм. Вкладка "Легенда"

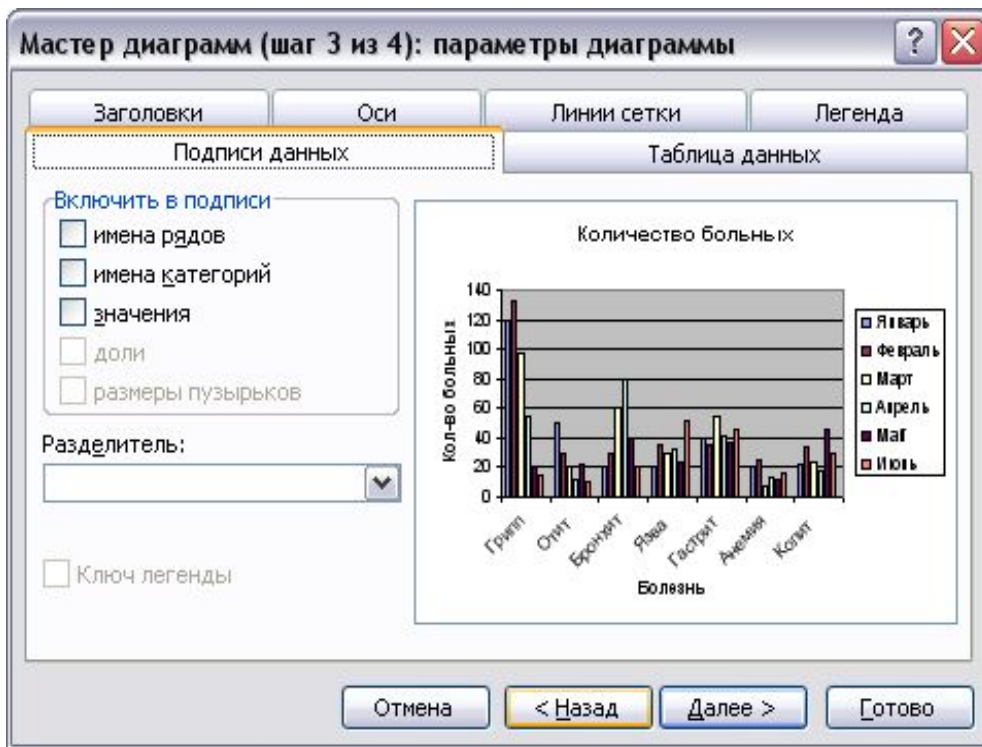


Рис. 8.43 – Вікно 3-го кроку Майстра діаграм. Вкладка "Подписи данных"

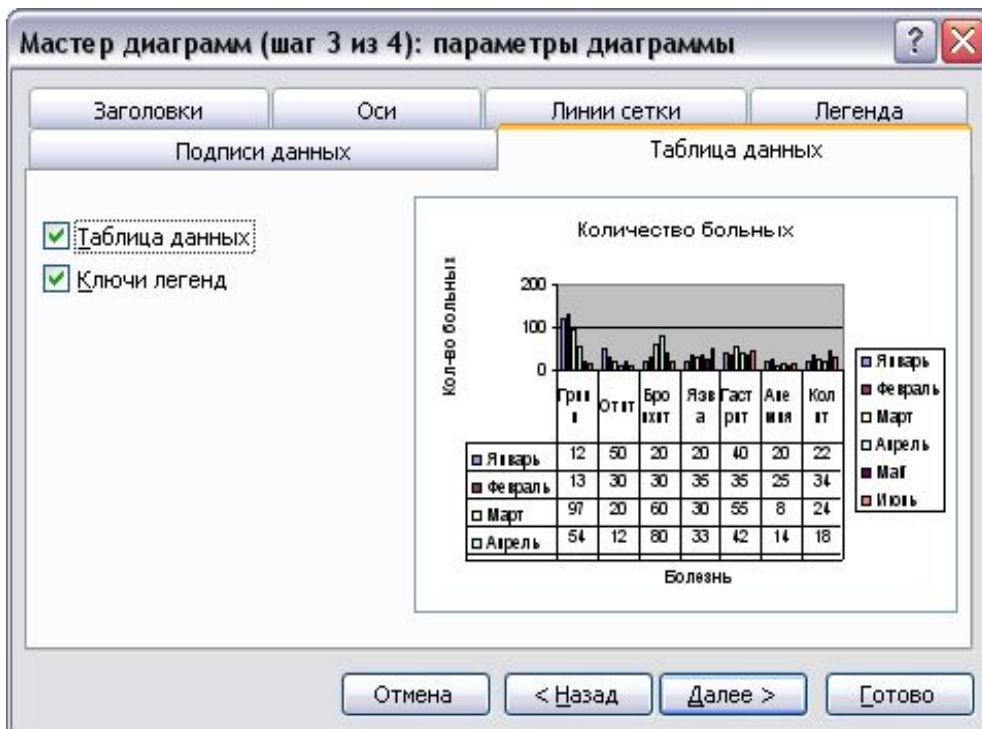


Рис. 8.44 – Вікно 3-го кроку Майстра діаграм. Вкладка "Таблица данных"

- З'явиться вікно четвертого кроку Майстра діаграм (рис. 8.45), в якому необхідно вказати, на якому листі розмістити діаграму і клацнути по кнопці **Готово**

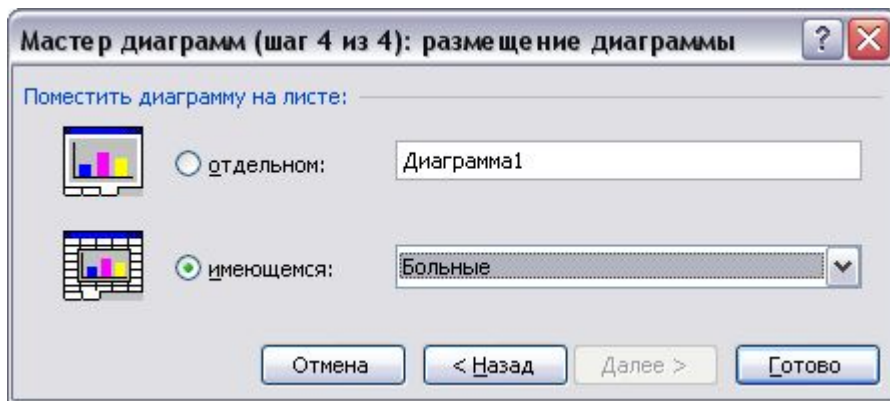


Рис. 8.45 – Вікно 4-го кроку Майстра діаграм.

### Зауваження.

Гістограма і лінійчата діаграми застосовуються для порівняння дискретних значень з декількох рядів даних.

Графік використовується для відображення динаміки змін значень ряду через рівні проміжки часу.

Кругова діаграма може відображати тільки один ряд даних, який в цілому можна вважати за 100%.

Кільцева діаграма може відображати декілька рядів даних, кожен з яких можна вважати за 100% і являються однотипними.

Точкова діаграма використовується для визначення типу залежності між двома рядами даних. Обидві осі – це осі значень.

Змішані діаграми використовують або два типи діаграм, або допоміжну вісь X, або допоміжну вісь Y. Використовуються коли деякі ряди даних змінюються в межах одного діапазону значень, а дані в інших рядах – в істотно відмінному діапазоні.

### 8.2.35.3 *Коректування елементів діаграм*

Для здійснення коректування діаграми в цілому необхідно виконати дії:

1. Виділити діаграму клацанням миші. З'явиться рамка для зміни розмірів, а також пункт меню "**Діаграма**".
2. Вибрати пункт "**Діаграма**" або клацнути мишею на вільному місці вікна діаграми і вибрати один з пунктів:
  - "**Тип діаграмми**" (при цьому з'явиться вікно 1-го кроку Майстра діаграм, де можна змінити тип і вид діаграми)
  - "**Исходные данные**" (при цьому з'явиться вікно 2-го кроку Майстра діаграм, де можна змінити діапазон даних, розташування рядів, імена рядів, значення рядів, підпис осі X, додати або видалити ряд)



- **"Параметры диаграммы"** (при цьому з'явиться вікно 3-го кроку Майстра діаграм, де можна додати або змінити назви діаграми і осей, додати або видалити легенду, підписати дані та інш.)
- **"Размещение"** (при цьому з'явиться вікно 4-го кроку Майстра діаграм, де можна змінити розміщення діаграми).

Для здійснення коректування окремого елемента діаграми необхідно виконати дії: клацнути правою кнопкою миші на окремому елементі діаграми і вибрати пункт **"Формат..."**.

### 8.2.36 Додавання лінії тренда

Лінія тренда – лінія регресії, яка апроксимує точки даних.

Щоб побудувати лінію тренда, бажано щоб тип діаграми був **"Точкова"**, а її вигляд – окремі крапки.

Для додавання лінії тренда необхідно спочатку виділити одним клацанням діаграму і одним клацанням дані, а потім правою кнопкою миші клацнути на одному з маркерів даних і в меню правого клацання виділених крапок вибрати пункт **"Добавить линию тренда"**. У вікні, що з'явиться (рис. 8.46), вибрати **Тип** лінії тренда (лінійна, логарифмічна, поліноміальна, степенна, експоненціальна) і клацнути по кнопці **ОК**.

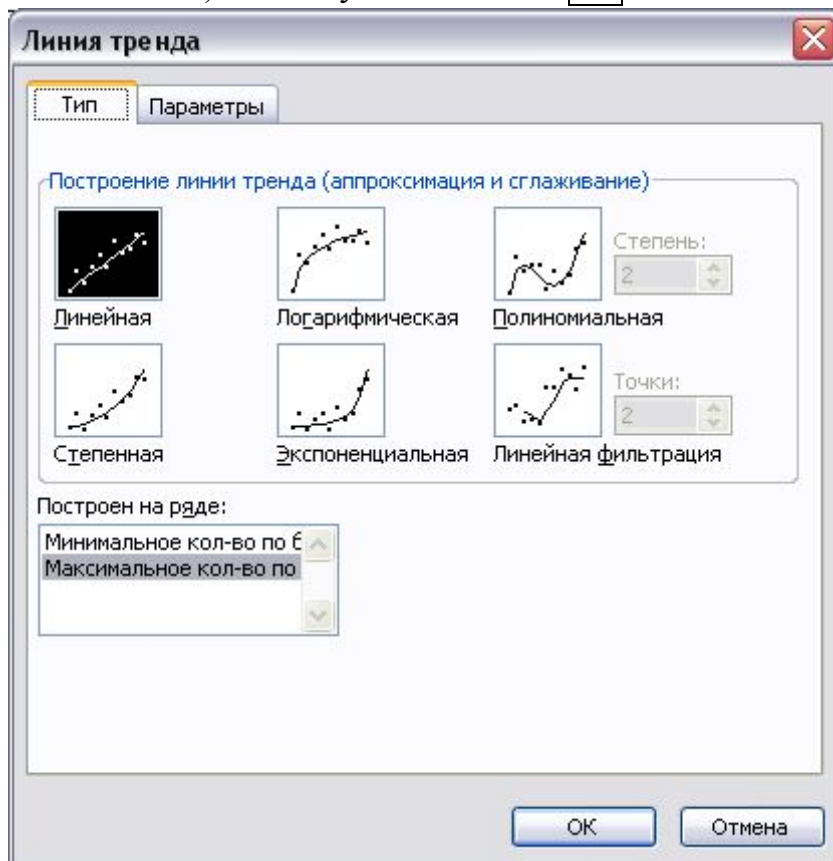
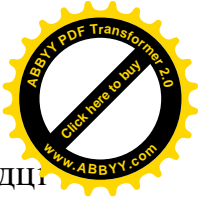


Рис. 8.46 – Вікно побудови лінії тренда. Вкладка "Тип"





Щоб правильно підібрати тип лінії тренда, необхідно у вкладці "**Параметри**" (рис. 8.47) для лінії тренда встановити два прапорці:

✓	"Показывать уравнение на диаграмме"
✓	"Поместить на диаграмму величину достоверности аппроксимации (R <sup>2</sup> )"

Величина **R<sup>2</sup>** показуватиме, наскільки близько дана лінія тренда описує початкові дані. Чим ближче **R<sup>2</sup>** до 1, тим точніше лінія тренда (**R<sup>2</sup>** приймає значення в діапазоні від 0 до 1).

Достовірною вважається лінія тренда, для якої **R<sup>2</sup> ≥ 0,75**.

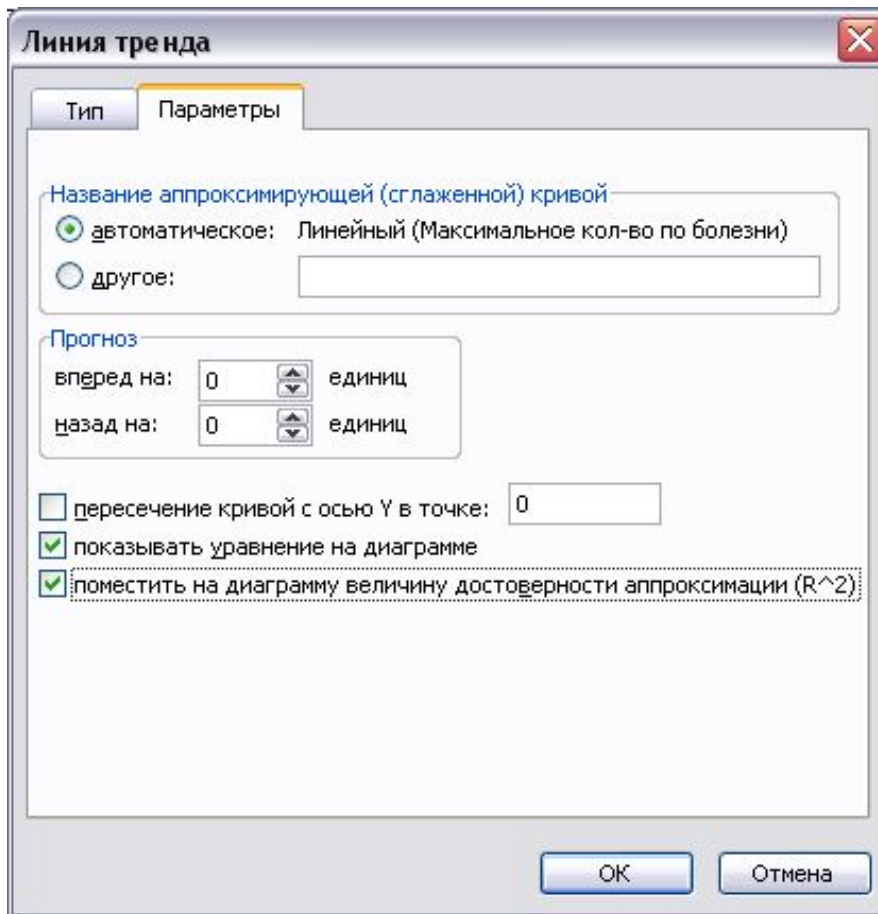


Рис. 8.47 – Вікно побудови лінії тренда. Вкладка "Параметры"

### 8.2.37 Прогноз даних за допомогою лінії тренда

Для прогнозування даних за допомогою лінії тренда необхідно виділити клацанням миші лінію тренда і в меню "**Формат**" вибрати "**Линии тренда**" або клацнути на лінії тренда правою кнопкою миші і в меню об'єкту вибрати "**Формат линии тренда**". У вікні, що з'явилося, вибрати вкладку "**Параметры**" і вказати в ній, на скільки періодів вперед або назад потрібен прогноз і натиснути кнопку **ОК**.



**Зауваження.** Якщо як лінія тренда вибрати поліноміальну залежність порядку більшого 2, то прогноз, швидше за все, буде недостовірним.

### 8.2.38 Формули масивів

Формулу масиву можна використовувати для створення формул, які повертають деяку безліч результатів.

Щоб ввести формулу масиву необхідно дотримуватися правил для створення формул масиву:

1. Спочатку потрібно виділити діапазон комірок, який служитиме для розміщення результату.
2. Ввести формулу.
3. Клацнути мишею в рядку формул.
4. Натиснути комбінацію клавіш **Ctrl+Shift+ENTER** для фіксації введення формули масиву. (Необхідно утримуючи натиснутими **Ctrl** і **Shift** натиснути **ENTER**).

Якщо дії виконані правильно, Excel бере формулу у фігурні дужки, вказуючи тим самим, що вона є формулою масиву. Але не можна вводити фігурні дужки самим, оскільки Excel розцінить це як введення текстового значення.

Якщо з якихось причин формула не розповсюдилася на всі виділені комірки, то, не знімаючи виділення з цих комірок, потрібно натиснути **F2**, а потім натиснути **Ctrl+Shift+ENTER** для фіксації введення формули масиву.

Не можна: редагувати, очищати або переміщати окремі комірки в діапазоні масиву, а також вставляти або видаляти комірки. Комірки в діапазоні масиву розглядаються як єдине ціле, і редагувати їх потрібно все відразу.

#### Приклад 1.

Знайти суму матриць  $A$  і  $B$ , які знаходяться в діапазонах  $A1:C3$  (матриця  $A$ ) і  $E1:G3$  (матриця  $B$ ).

Для знаходження суми матриць необхідно виконати такі дії:

1. Виділити область під результат, наприклад:  $A5:C7$
2. Ввести знак =
3. Виділити область з першою матрицею  $A$ :  $A1:C3$
4. Ввести знак +
5. Виділити область з другою матрицею  $B$ :  $E1:G3$
6. Клацнути мишею в рядку формул і натиснути **Ctrl+Shift+ENTER**

В результаті в комірках  $A5:C7$  з'явиться формула  $\{=A1:C3 + E1:G3\}$

#### Приклад 2.

Помножити елементи матриці  $A$ , яка знаходиться в комірках  $A1:C3$  на коефіцієнт 1,5



Для знаходження результату необхідно виконати такі дії:

1. Виділити область під результат, наприклад  $A9:C11$
2. Ввести знак =
3. Виділити область з матрицею  $A$ :  $A1:C3$
4. Ввести знак \* і 1,5
5. Натиснути **Ctrl+Shift+ENTER**

В результаті в комітках  $A9:C11$  з'явиться формула  $\{=A1:C3*1,5\}$

### 8.2.39 Функції роботи з матрицями

**МУМНОЖ**("масив1";"масив2" [...]) - перемноження матриць;

**МОБР**("масив") - знаходження зворотної матриці;

**МОПРЕД**("масив") - обчислення визначника матриці;

**ТРАНСП**("масив") - транспонування матриці,

тут функції **МУМНОЖ**, **МОБР**, **ТРАНСП** можна використовувати тільки в формулах масивів.

Розглянемо етапи виконання перемноження матриць за допомогою функції **МУМНОЖ**:

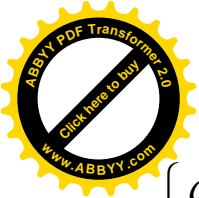
1. Виділити діапазон комірок для матриці-результату. При цьому число рядків результуючої матриці дорівнює числу рядків першої матриці, а число стовпців – числу стовпців другої матриці.
2. Викликати майстер функцій:  $f_x$  → категорія "*Математические*" → функція **МУМНОЖ**
3. У вікні діалогу в полі першого аргументу ввести інтервал комірок, в яких розташована перша матриця
4. У полі другого аргументу ввести інтервал комірок, в яких розташована друга матриця.
5. Клацнути в рядку формул і натиснути комбінацію **Ctrl+Shift+ENTER**.

У виділеній області з'явиться результат добутку двох матриць. При цьому треба пам'ятати, що перемноження матриць не підкоряється переміщувальному закону множення ( $A*B \neq B*A$ ).

При використанні функції **ТРАНСП** слід пам'ятати, що замість матриць можна обробляти таблиці с заголовками, окремі рядки заголовків або окремі стовпці. І тоді при виділенні області для результату число рядків дорівнює числу стовпців початкової матриці, а число стовпців = числу рядків початкової матриці.

### 8.2.40 Розв'язування системи лінійних алгебраїчних рівнянь (СЛАУ)

Розв'язати систему:



$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

або в матричному вигляді  $A \cdot X = B$ ,

де,

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix}.$$

### 8.2.40.1 Метод зворотної матриці

Для рішення СЛАУ методом зворотної матриці необхідно розглянути матричне рівняння:

$$A \cdot X = B.$$

Далі необхідно помножити обидві частини матричного рівняння зліва на зворотну матрицю:

$$A^{-1} \cdot A \cdot X = A^{-1} \cdot B.$$

Тепер можна замінити добуток зворотної матриці на пряму одиничною матрицею:

$$E \cdot X = A^{-1} \cdot B, \quad \text{де } E \text{ – одинична матриця.}$$

Тоді рішенням буде матриця:

$$X = A^{-1} \cdot B$$

### 8.2.40.2 Метод Крамера

Для рішення СЛАУ методом Крамера необхідно розглянути матричне рівняння:

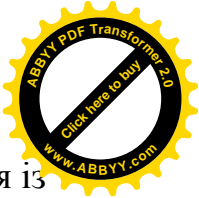
$$A \cdot X = B.$$

Невідомі можна знайти як відношення допоміжних визначників матриці коефіцієнтів до основного визначника:

$$x_i = \frac{\Delta_i}{\Delta}, \quad (i=1, 2, \dots, n),$$

де  $n$  – кількість рівнянь в СЛАУ,

$\Delta$  - основний визначник матриці коефіцієнтів,



$\Delta_i$  – допоміжні визначники, тобто визначники матриць, які створюються із матриці коефіцієнтів шляхом заміни  $i$ -го стовпця на стовпець вільних членів.

Приклад розв’язання СЛАУ за допомогою EXCEL:

$$\begin{cases} x_2 - 13x_3 + 4x_4 = -5 \\ x_1 - 2x_3 + 3x_4 = -4 \\ 3x_1 + 21x_2 - 5x_4 = 2 \\ 4x_1 + 3x_2 - 5x_3 = 3 \end{cases}$$

Спочатку необхідно ввести коефіцієнти при невідомих (матриця  $A$ ) і стовпець вільних членів (вектор  $B$ ) в комірки робочого листу Excel, розташовуючи матрицю  $A$  в комірках  $B1:E4$ , а вектор  $B$  – в комірках  $H1:H4$ .

Перед матрицею і вектором необхідно розташувати підписи: у комірці  $A3$  записати текст "A=", а в комірці  $G3$  – текст "B=" (рис. 8.48).

	A	B	C	D	E	F	G	H
1		0	1	-13	4			-5
2		1	0	-2	3			-4
3	A=	3	21	0	-5		B=	2
4		4	3	-5	0			3
5								
6								
7								
8	Аобр=						X=	
9								

Рис. 8.48 – Фрагмент листа Excel для розв’язання СЛАУ

Розв’язання СЛАУ методом зворотної матриці.

Для розв’язання СЛАУ методом зворотної матриці необхідно:

1. В комірці  $A8$  розташувати підпис "Аобр=".
2. Потім виділити комірки  $B6:E9$  для зберігання зворотної матриці.
3. Викликати майстер функцій ( $f_x$ ) і в категорії "Математические" необхідно вибрати функцію **МОБР**.
4. З’явиться вікно для заповнення аргументів функції (рис. 8.48), де як "массив" необхідно вказати діапазон, в якому знаходиться початкова матриця  $A$ , тобто  $B1:E4$ . Клацнути мишею в рядку формул і натиснути **Ctrl+Shift+ENTER**.

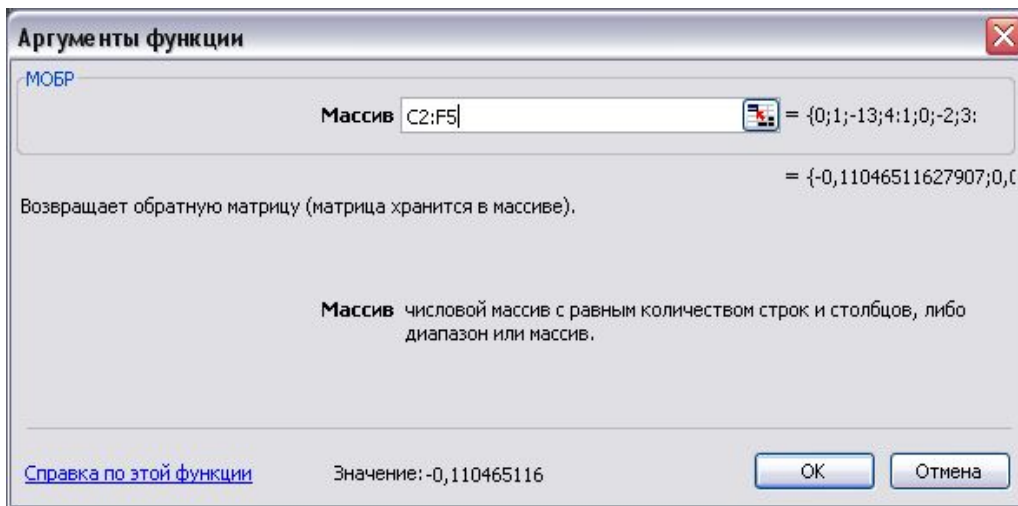


Рис. 8.49 – Вікно функції МОБР

5. Тепер треба помножити одержану зворотну матрицю на вектор  $B$ . Для цього в комірці  $G8$  треба розташувати підпис "X="
6. А потім необхідно виділити область  $H6:H9$  для зберігання результуючого вектора  $X$
7. Викликати майстер функцій ( $f_x$ ).
8. В категорії "*Математические*" вибрати функцію МУМНОЖ.
9. З'явиться вікно для заповнення аргументів функції (рис. 8.50), де як "массив1" вказати діапазон, в якому знаходиться зворотна матриця, тобто  $B6:E9$ , а як "массив2" вказати діапазон, в якому знаходиться вектор вільних членів  $B$ , тобто  $H1:H4$
10. Клацнути мишею в рядку формул і натиснути **Ctrl+Shift+ENTER**.
11. Зробити перевірку знайденого рішення, тобто помножити матрицю коефіцієнтів на стовпець невідомих за допомогою функції МУМНОЖ. В результаті повинен вийти вектор  $B$ .

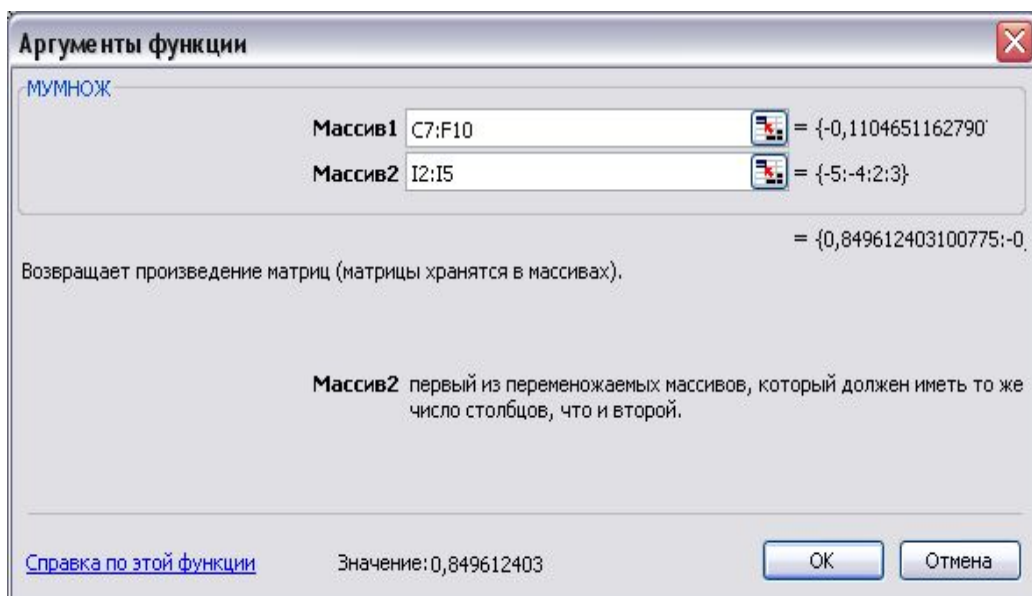


Рис. 8.50 – Вікно функції МУМНОЖ





### Розв'язання СЛАУ методом Крамера.

Для розв'язання СЛАУ методом Крамера необхідно:

1. Скопіювати початкову матрицю 4 рази (стільки раз скільки рівнянь в СЛАУ)
2. Перша копія може знаходитись в комірках  $B6:E9$ , друга – в комірках  $B11:E14$ , третя – в комірках  $B16:E19$ , а четверта – в комірках  $B21:E24$ .
3. В комірки  $A7$ ,  $A12$ ,  $A17$ ,  $A22$  треба помістити тексти "A1=", "A2=", "A3=" і "A4=" відповідно.
4. В першій копії замінити перший стовпець вектором  $B$ , в 2-ій, 3-ій і 4-ій копіях – відповідно 2-ий, 3-ій і 4-ий стовпець замінити вектором  $B$ .
5. Обчислити основний і допоміжні визначники (визначники матриць  $A$ ,  $A1$ ,  $A2$ ,  $A3$ ,  $A4$ ), скориставшись 5 разів функцією **МОПРЕД**.
6. Результати обчислень пояснити підписами "Опр=", "Опр1=", "Опр2=", "Опр3=", "Опр4=".
7. Для знаходження коренів СЛАУ треба розділити кожний допоміжний визначник на основний, пояснивши результати обчислень текстом "X=" і розмістивши їх один над одним. Вийде вектор  $X$ .
8. Зробити перевірку знайденого рішення, тобто помножити матрицю коефіцієнтів на стовпець невідомих за допомогою функції **МУМНОЖ**. В результаті повинен вийти вектор  $B$ .

### 8.2.41 Макроси

Макрос – це набір інструкцій, які вказує послідовність дій, які EXCEL повинен виконати замість користувача.

Макроси використовуються для автоматизації трудомістких і задач, що часто повторюються. В EXCEL макрос можна створювати, записуючи команди меню, натиснення клавіш і інші дії.

Після запису послідовності дій можна запускати макрос всякий раз, коли потрібно виконати цю задачу знову.

Процес запису макросу складається з 3 кроків:

1. **"Сервіс" → "Макрос" → "Начать запись"**.

З'явиться вікно (рис. 8.51), в якому потрібно:

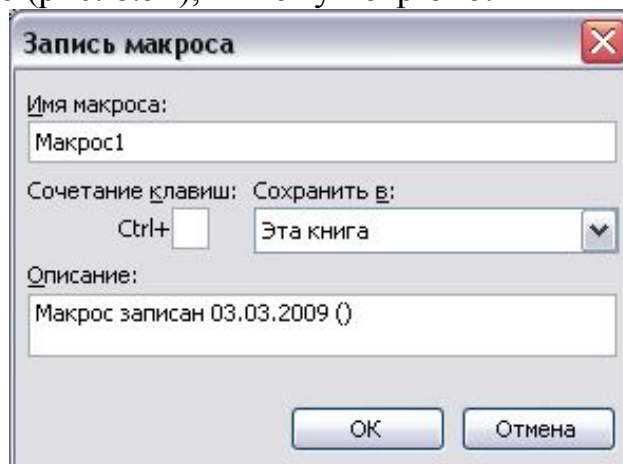


Рис. 8.51 – Вікно запису макроса

а) заповнити поле "*Имя макроса*", ввівши будь-яке ім'я, але без пропусків, крапок, ком та інших спеціальних символів.

- б) заповнити поле "*Сохранить в*", вибравши один з 3 можливих варіантів:
- "*Эта книга*" (тоді макрос буде доступний при відкритті цієї книги);
  - "*Личная книга макросов*" (тоді макрос буде доступний з будь-якої книги EXCEL);
  - "*Новая книга*" (тоді макрос буде доступний відразу після створення нової книги EXCEL).


в) якщо потрібно призначити клавішу швидкого виклику макросу, ввівши в полі "*Сочетание клавиш*" будь-яку латинську букву.

Після натиснення **ОК** на екрані з'явиться панель інструментів з кнопкою "*Остановить запись*" (рис. 8.52).



Рис. 8.52 – Панель інструментів для зупинення запису макроса

2. Виконати всі дії, які EXCEL повинен буде згодом відтворити.
3. Зупинити запис макросу одним з 2-х способів:

- Натиснути кнопку "*Остановить запись*" 
- "*Сервис*" → "*Макрос*" → "*Остановить запись*"

**Зауваження!!!** Якщо не зупинити запис макроса, EXCEL записуватиме дії нескінченно.

Для виклику макросу необхідно: "*Сервис*" → "*Макрос*" → "*Макросы*" → вибрати ім'я потрібного макросу → клацнути по кнопці **Выполнить**.

**Зауваження:** Якщо макрос створюється для форматування комірок, то

- 1) перед записом макросу необхідно виділити комірку, на прикладі форматування якої буде записуватись макрос;
- 2) під час запису макросу саме з цією коміркою виконати необхідні дії;
- 3) перед викликом макросу для такого ж форматування інших комірок необхідно виділити область застосування макросу.

### 8.2.42 Рішення задач оптимізації

**Постановка задачі:** Потрібно знайти такі значення невідомих, які задовольняють ряду обмежень, і при яких функція мети досягає оптимального значення (мінімального або максимального).

Розв'язати цю задачу в EXCEL можна за допомогою **Вирішального блоку**. Для виклику **Вирішального блоку** необхідно виконати команди:



### "Сервис" → "Поиск решения"

Якщо немає такого пункту в меню "Сервис", то спочатку необхідно виконати команди: "Сервис" → "Надстройки" → включити прапорець "Поиск решения"

#### Приклад.

Фабрика випускає два типи фарб: для внутрішніх (фарба І) і зовнішніх (фарба Е) робіт. Для виробництва фарб використовуються два початкові продукти А і В. Витрата початкових продуктів на тону фарби наведені в табл. 8.5.

Таблиця 8.5 – Норма витрат продуктів на тону фарби

Початковий продукт	Витрата продуктів на тону фарби, т		Максимально можливий запас, т
	Фарба Е	Фарба І	
А	1	2	6
В	2	1	8

Добовий попит на фарбу І ніколи не перевищує попиту на фарбу Е більше ніж на 1т. Попит на фарбу І ніколи не перевищує 2т на добу. Оптові ціни однієї тони фарб рівні: 3000 крб. для фарби Е і 2000 крб. для фарби І.

Яку кількість фарби кожного виду повинна виробляти фабрика, щоб дохід від реалізації продукції був максимальним?

Тобто необхідно спланувати об'єм виробництва фарб так, щоб максимізувати прибуток.

Для вирішення цієї задачі необхідно побудувати математичну модель. Для цього потрібно відповісти на 3 запитання:

#### 1. Які змінні моделі?

$X_1$  - добовий об'єм виробництва фарби Е.

$X_2$  - добовий об'єм виробництва фарби І

#### 2. У чому полягає мета?

$Z = 3000 \cdot X_1 + 2000 \cdot X_2 \rightarrow \max$  – тобто сумарний добовий прибуток від виробництва фарби І і фарби Е повинен бути максимальним.

#### 3. Яким обмеженням повинні задовольняти невідомі?

$X_1, X_2 \geq 0$  – об'єм виробництва фарб не може бути від'ємним

$\left\{ \begin{array}{l} 1 \cdot X_1 + 2 \cdot X_2 \leq 6 \end{array} \right.$  – витрата продукту А не повинен бути більше запасу

$\left\{ \begin{array}{l} 2 \cdot X_1 + 1 \cdot X_2 \leq 8 \end{array} \right.$  – витрата продукту В не повинен бути більше запасу

$\left\{ \begin{array}{l} X_2 - X_1 \leq 1 \end{array} \right.$  – обмеження на величину попиту на фарби

$X_2 \leq 2$

Для деяких задач повинно додаватися обмеження на те, щоб невідомі були цілими.

Таким чином, математична модель даної задачі має наступний вигляд:

$$Z = 3000 \cdot X_1 + 2000 \cdot X_2 \rightarrow \max$$



при наступних обмеженнях:

$$\begin{cases} 1 \cdot X_1 + 2 \cdot X_2 \leq 6, \\ 2 \cdot X_1 + 1 \cdot X_2 \leq 8, \\ X_2 - X_1 \leq 1, \\ X_2 \leq 2 \\ X_1, X_2 \geq 0 \end{cases}$$

Для розв'язання цієї задачі необхідно розмістити дані на листі Excel таким чином (рис. 8.53):

	A	B	C	D
1	Переменные	X1	X2	
2		0	0	
3				
4		Краска E	Краска I	Запас продукта
5	Продукт A	1	2	6
6	Продукт B	2	1	8
7	Оптовая цена	3000	2000	
8				
9	Функция цели	Z=	=B7*B2+C7*C2	
10				
11	Ограничение1	=B5*\$B\$2+C5*\$C\$2	≤	6
12	Ограничение2	=B6*\$B\$2+C6*\$C\$2	≤	8
13	Ограничение3	=C2-B2	≤	1
14	Ограничение4	=C2	≤	2

Рис. 8.53 – Розміщення даних на листі Excel

1. Спочатку необхідно визначитися, в яких комірках поміщати значення невідомих величин. Наприклад це будуть комірки B2:C2. Спочатку треба помістити в них нулі, а в комірках B1:C1 помістити підписи цих величин.
2. В комірки A4:D7 помістити таблицю з даними задачі, доповнивши її оптовими цінами на 1 тону кожного виду фарби.
3. Потім в комірку C9 необхідно помістимо формулу, за якою обчислюється цільова функція Z. При цьому слід пам'ятати, що X1 знаходиться в комірці B2, а X2 – в комірці C2. Таким чином в комірці C9 формула може бути



$$\begin{aligned} &=3000*B2+2000*C2 \text{ або} \\ &=B7*B2+C7*C2 \text{ або} \\ &=\text{СУММПРОИЗВ}(B2:C2;B7:C7) \end{aligned}$$

4. В комірки  $A11:A14$  необхідно ввести назви обмежень: "Ограничение1", "Ограничение2" і т.д.
5. В комірки  $B11:B14$  необхідно ввести ліві частини обмежень, а саме в комірку  $B11$ :  
 $=1*B2+2*C2$  або  
 $=B5*\$B\$2+C5*\$C\$2$  або  
 $=\text{СУММПРОИЗВ}(\$B\$2:\$C\$2;B5:C5);$   
в комірку  $B12$ :  
 $=2*B2+1*C2$  або  
 $=B6*\$B\$2+C6*\$C\$2$  або  
 $=\text{СУММПРОИЗВ}(\$B\$2:\$C\$2;B6:C6);$   
в комірку  $B13$ :  $=C2-B2;$   
в комірку  $B14$ :  $=C2$
6. В комірки  $C11:C14$  необхідно ввести знаки обмежень
7. В комірки  $D11:D14$  необхідно ввести праві частини обмежень

Саме таким чином поміщаються дані задачі на лист EXCEL для подальшого розв'язання задачі за допомогою **Вирішального блоку**.

Після розміщення даних задачі на лист EXCEL слід виконати такі дії:

1. Викликати **Вирішальний блок: "Сервис" → "Поиск решения"**.
2. В полі "**Установить целевую ячейку**" вказати мишею комірку, де знаходиться формула для цільової функції (в даному випадку це комірка  $\$C\$9$ ). Потім вказати режим пошуку за допомогою одного з перемикачів "**максимальному значенню**" або "**минимальному значенню**" (в даному випадку активізувати перемикач "**максимальному значенню**").
3. Потім в полі "**Изменяя ячейки**" вказати мишею, де знаходяться комірки, відведені для невідомих величин (в даному випадку це діапазон  $\$B\$2:\$C\$2$ ).
4. Заповнити поле "**Ограничения**". Для цього можна клацнути по кнопці **Добавить**. У вікні, що з'явиться (рис. 8.54), в полі "**Ссылка на ячейку**" вказати мишею комірку, де знаходиться ліва частина 1-го обмеження (в даному випадку це комірка  $\$B\$11$ ), вибрати із списку потрібний знак, а потім в полі "**Ограничение**" вказати комірку, де знаходиться права частина обмеження (в даному випадку це комірка  $\$D\$11$ ), і клацнути по кнопці **Добавить**. Аналогічно заповнити решту обмежень. Якщо правої частини обмежень немає на листі EXCEL, то її можна

ввести в полі "Ограничение" з клавіатури. Після останнього обмеження клацнути по кнопці **ОК**. Якщо яке-небудь обмеження заповнене невірно, його можна відредагувати, виділивши і клацнувши по кнопці **Изменить** або видалити, якщо клацнути по кнопці **Удалить**.

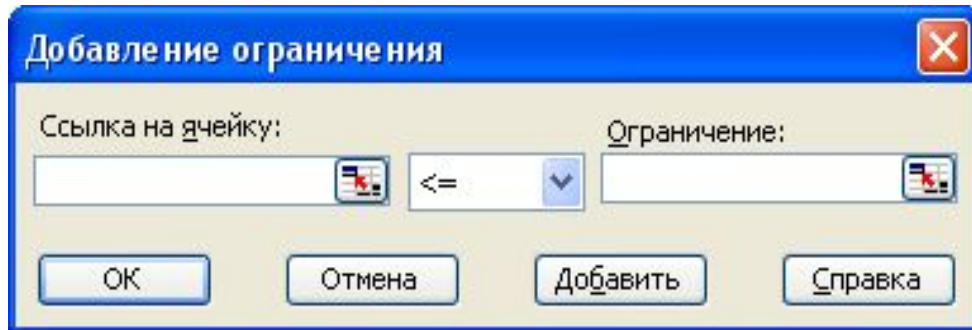


Рис. 8.54 – Вікно додавання обмежень

- Коли всі параметри пошуку заповнені (рис. 8.55), необхідно клацнути по кнопці **Выполнить** і у вікні, що з'явиться (рис. 8.56), дізнатися про результати пошуку. У разі позитивного результату вибрати перемикач "Сохранить найденное решение" і клацнути по кнопці **ОК**, а інакше клацнути по кнопці **Відміна**.

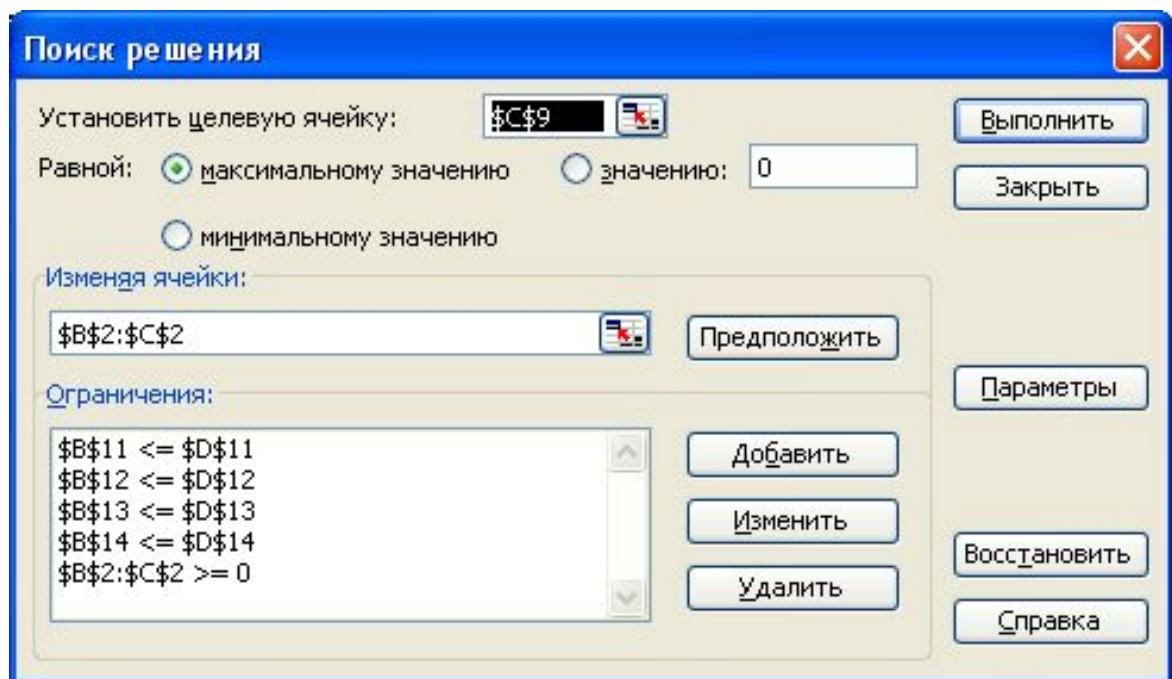


Рис. 8.55 – Вікно параметрів Вирішального блоку



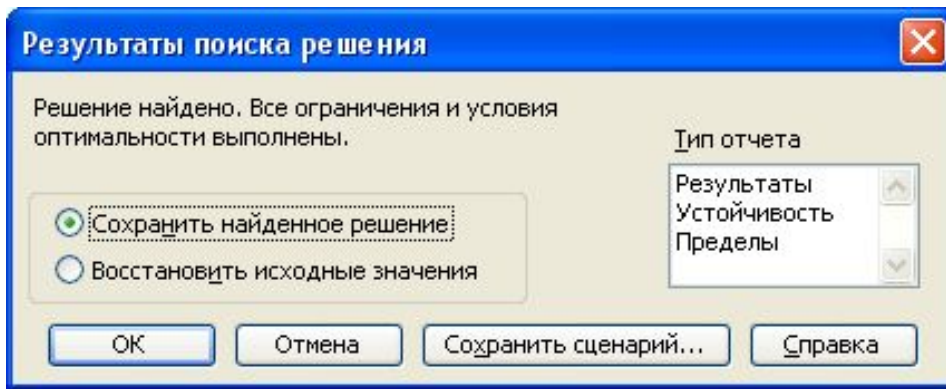
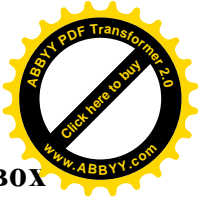


Рис. 8.56 – Вікно результатів пошуку рішення

В результаті розв’язання задачі за допомогою *Вирішального блоку* (рис. 8.57) в комірках B2:C2 з’являться значення результату пошуку рішення – план виробництва кожного виду фарби, в комірці C9 – значення цільової функції – максимальний прибуток, а в комірках B11:B14 – значення лівих частин обмежень при знайдених значеннях невідомих, з яких можна зробити висновок про те, що саме заважає отримати ще більший прибуток .

	A	B	C	D
1	Переменные	X1	X2	
2		3,33333	1,33333	
3				
4		Краска Е	Краска І	Запас продукта
5	Продукт А	1	2	6
6	Продукт В	2	1	8
7	Оптовая цена	3000	2000	
8				
9	Функция цели	Z=	12666,7	
10				
11	Ограничение1	6	≤	6
12	Ограничение2	8	≤	8
13	Ограничение3	-2	≤	1
14	Ограничение4	1,333333	≤	2

Рис. 8.57 – Фрагмент листа EXCEL з результатами розв’язання задачі



### 8.2.43 Використовування функції ПРОСМОТР для зв'язку двох таблиць

Щоб мати можливість зв'язати дві таблиці за допомогою функції ПРОСМОТР необхідно, щоб виконувались такі умови:

1. Обидві таблиці повинні мати спільний стовпець.
2. Одна таблиця повинна бути довідковою для іншої.
3. У стовпці довідкової таблиці, який зв'язує обидві таблиці не повинно бути даних, що повторюються.
4. Довідкова таблиця повинна бути відсортована по спільному полю.

Загальний вид функції ПРОСМОТР :

**ПРОСМОТР("Искомое\_значение";"Просматриваемый\_вектор";"Вектор\_результатов")**

Приклад. Розглянемо 2 таблиці: довідкова – з тарифами для різних розрядів і робоча (рис. 8.58).

	А	В	С	Д
1	<b>Разряд</b>	<b>Стоимость 1 часа работы</b>		
2	Первый	4		
3	Второй	4,5		
4	Высший	7,5		
	...			
8	<b>Ф.И.О.</b>	<b>Разряд</b>	<b>К-во отработ. Часов</b>	<b>Сумма за все часы</b>
9	Иванов	Высший	30	
10	Петров	Первый	20	
11	Сидоров	Высший	40	

Рис. 8.58 – Фрагмент листа EXCEL з двома таблицями

Для заповнення стовпця "Сумма за все часы" робочої таблиці необхідно виконати дії:

1. Відсортувати довідкову таблицю по загальному полю ("Разряд").
2. У комірку D9 за допомогою майстра функцій в категорії "**Ссылки и массивы**" вибрати функцію ПРОСМОТР, а потім вид функції ПРОСМОТР з трьома аргументами (рис. 8.59).

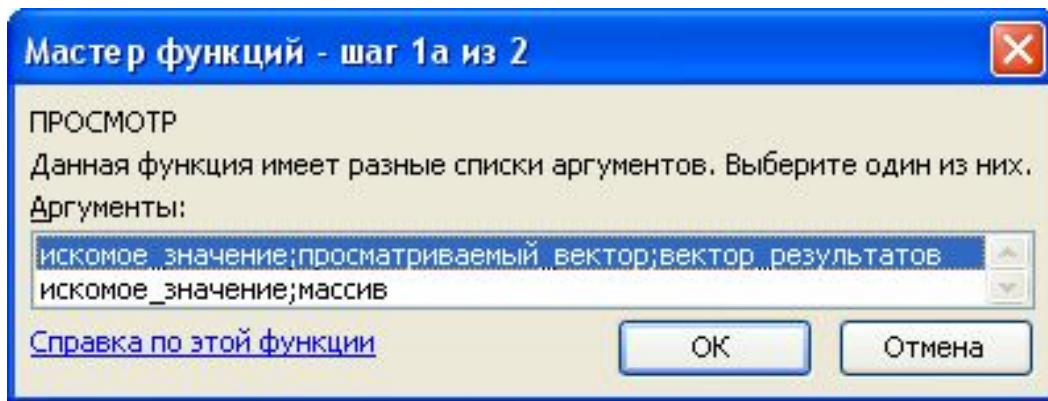


Рис. 8.59 – Вікно вибору виду функції ПРОСМОТР

3. В полі 1-го аргументу вибрати елемент робочої таблиці відповідного рядка спільного стовпця ("*Разряд*"), тобто комірку *B9*.
4. В полі 2-го аргументу вибрати діапазон спільного стовпця довідкової таблиці, тобто діапазон *A2:A4*.
5. В полі 3-го аргументу вибрати діапазон стовпця значень з довідкової таблиці, тобто діапазон *B2:B4*.
6. Діапазони 2-го і 3-го аргументів необхідно перетворити з відносних посилань на абсолютні посилання, виділивши їх і натиснувши клавішу **F4**, оскільки "*Просматриваемый\_вектор*" і "*Вектор\_результатов*" для формул, які будуть створені шляхом копіювання створеної формули в інші комірки, весь час повинні бути одними й тими ж.
7. Клацнути мишею в кінці рядка формул і натиснути \*, а потім клацнути по комірці *C9* і натиснути **ENTER**.
8. У комірці *D9* з'явиться формула  

$$=ПРОСМОТР(B9; \$A\$2: \$A\$4; \$B\$2: \$B\$4) * C9$$
9. Протягнути формулу з комірки *D9* в комірки *D10:D11*.

#### 8.2.44 Операції з датами

1. Якщо до дати додати число, то до неї додається кількість днів, і результатом буде нова дата, але формат комірки повинен бути "*Дата*".
2. Якщо відняти одну дату з іншої, то вийде кількість днів між цими датами, але формат комірки повинен бути "*Числовой*".
3. Щоб одержати різницю між датами в роках, необхідно різницю між датами взяти в круглі дужки і розділити на 365, в місяцях – різницю між датами розділити на 365 і помножити на 12, а в тижнях – різницю між датами розділити на 7 і т.д..



## 8.2.45 Функції категорії "Дата и время"

**СЕГОДНЯ()** – повертає системну дату. Значення цієї функції не оновлюється безперервно, а міняє своє значення тільки при розрахунку.

**ГОД("дата")** - виділяє рік з дати.

**МЕСЯЦ("дата")** - виділяє місяць з дати.

**ДЕНЬ("дата")** - виділяє день місяця з дати.

**ДЕНЬНЕД("дата" [; "тип"])** - повертає порядковий номер дня тижня, але перший день тижні різний залежно від типу.

Якщо "тип"=1 або відсутній, то 1-вс, 2-пн, 3-вт..., 7-сб;

"тип"=2, то 1-пн, 2-вт, 3-ср..., 7-вс;

"тип"=3, то 0-пн, 1-вт, 2-ср..., 6-вс.

Можна використовувати функцію **ТЕКСТ("дата"; "ДДДД")**, щоб перетворити дату в найменування дня тижня.

**ДАТАМЕС("нач\_дата"; "число\_месяцев")** - повертає дату, віддалену на задану кількість місяців вперед або назад від заданої початкової дати.

"Нач\_дата" – це початкова дата;

"Число\_месяцев" – це кількість місяців до або після дати "нач\_дата".

Додатне значення аргументу "число\_месяцев" означає майбутні дати; від'ємне значення – минулі дати.

**ДОЛЯГОДА("нач\_дата"; "кон\_дата" [; "базис"])** - повертає кількість років між двома датами (початковою і кінцевою).

"Нач\_дата" – це дата, яка відповідає початковій даті.

"Кон\_дата" – це дата, яка відповідає кінцевій даті.

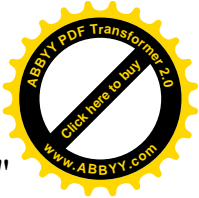
"Базис" – це тип використовуваного способу обчислення дня.

**Зауваження!** Якщо функції **ДАТАМЕС** і **ДОЛЯГОДА** недоступні, їх можна підключити таким чином: "**Сервис**" → "**Надстройки**" → встановити прапорець  **Пакет Аналізу**.

### Приклади

Хай в комірці **B2** зберігається "дата1" – 01.05.99, а в комірці **C2** зберігається "дата2" – 31.03.02.

Нижче розглянуті приклади використання функцій категорії "Дата и время" (табл. 8.6).



Таблиця 8.6 – Приклади використання функцій категорії "Дата и время"

№ п/п	Що треба знайти?	Формула	Результат
1.	Отримати поточну дату.	=СЕГОДНЯ()	поточна дата
2.	Виділити рік з "дати1".	=ГОД(В2)	1999
3.	Виділити місяць з "дати1".	=МЕСЯЦ(В2)	5
4.	Виділити день місяця з "дати1".	=ДЕНЬ(В2)	1
5.	Отримати день тижня для "дати1".	=ДЕНЬНЕД(В2;2) або =ТЕКСТ(В2;"ДДДД")	6 суббота
6.	Отримати дату, віддалену від "дати2" на півроку вперед.	=ДАТАМЕС(С2;6)	30.09.02
7.	Взнати, скільки років між "датою2" і "датою1"	=ДОЛЯГОДА(В2;С2) або =(С2-В2)/365	2,9 2,9

### 8.2.46 Використовування фільтрів

Щоб із загального списку вивести список даних, що задовольняють деякій умові, можна відфільтрувати цей список.

Відфільтрувати список – означає приховати всі рядки, які не задовольняють заданим умовам відбору, і залишити всі рядки, які задовольняють заданим умовам відбору.

**Зауваження!**

Оскільки створення фільтру викликає приховування деяких рядків, то фільтр бажано створювати на копії початкової таблиці.

#### 8.2.46.1 Автофільтр

Для створення автофільтру необхідно виконати такі дії:

1. Спочатку виділити копію таблиці або ті її стовпці, які розташовані поряд і на які в подальшому накладатимуться умови.
2. Потім виконати команди: **"Данные"** → **"Фільтр"** → **"Автофільтр"**.

При цьому EXCEL виведе кнопки автофільтру поряд з кожним заголовком стовпця (кнопки автофільтру – це сірі кнопки з чорними стрілками, дивись приклад на рис. 8.60).

	Наименование товара	Дата продажи	Количество	Цена
3				
4	Вилка	01.09.01	150	12
5	Нож	10.09.01	98	10
6	Ложка	15.08.01	250	8

Рис. 8.60 – Приклад списку з кнопками автофільтру

- Клацнути по кнопці автофільтра потрібного стовпця. При цьому відкриється список ймовірних варіантів вибору. Наприклад, для першого стовпця список вибору буде таким (рис. 8.61)

A3		= Наименование товара			
	A	B	C	D	
1					
2					
3	Наименование товара	Дата продаж	Количество	Цена	
	(Все)	09.01	150	12	
	(Первые 10...)	09.01	98	10	
	(Условие...)				
	Вилка	08.01	250	8	
	Ложка				
	Нож				

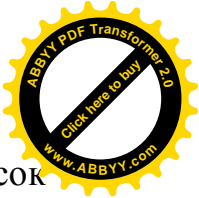
Рис. 8.61 – Приклад списку кнопки авто фільтру

- Тут можна вибрати будь-яке можливе конкретне значення (наприклад, "Нож") або пункт "(Условие...)" для вказівки однієї або двох умов, поєднаних логічними операціями "И" або "ИЛИ" (рис. 8.62).

Рис. 8.62 – Вікно умов для кнопки автофільтру

- Після вибору конкретного значення або вибору умов відбору на екрані залишаться тільки рядки, що задовольняють умовам відбору. Номери відфільтрованих рядків і кнопки автофільтру з накладеними умовами відобразяться синім кольором. А рядки, що не задовольняють умовам відбору, стануть прихованими.





Щоб зняти накладені умови, достатньо розкрити відповідний список кнопки автофільтру і вибрати пункт "**(Все)**".

Щоб видалити кнопки авто фільтру, необхідно повторно виконати команди "**Данные**" → "**Фильтр**" → "**Автофильтр**".

Автофільтр можна застосувати до будь-якої кількості стовпців.

**Зауваження!** На одному листі можна використовувати автофільтр тільки 1 раз. Якщо необхідно отримати декілька різних списків, то необхідно скопіювати початкову таблицю на декілька листів книги Excel і на різних листах виконувати команду "**Автофильтр**" і використовувати різні умови для кнопок автофільтру.

Приклад. Маються наступні дані, розташовані в діапазоні **A1:C6** (рис. 8.63). Вивести список робітників з віком від 20 до 30 років, що народилися в травні.

	А	В	С
1	<b>Ф.И.О. рабочего</b>	<b>Дата рождения</b>	<b>Разряд</b>
2	Иванов	15.05.50	6
3	Петров	22.04.80	4
4	Сидоров	11.05.84	3
5	Зайцев	01.02.89	1
6	Волков	07.05.78	4

Рис. 8.63 – Фрагмент листу Excel з наданим списком

Для отримання потрібного списку необхідно виконати дії:

1. Скопіювати таблицю на новий лист і всі подальші дії виконувати на копії таблиці.
2. У комірці **D2** обчислити вік по формулі  $=(\text{СЕГОДНЯ}()-\text{B2})/365$  і встановити формат комірки "**Числової**" з 2 знаками після коми, а в комірці **E2** обчислити місяць народження по формулі  $=\text{МЕСЯЦ}(\text{B2})$  і протягнути обидві формули за маркер заповнення (скопіювати їх) в нижні комірки і таблиця прийме вигляд, показаний на рис. 8.64.

	А	В	С	Д	Е
1	<b>Ф.И.О. рабочего</b>	<b>Дата рождения</b>	<b>Разряд</b>	<b>Возраст</b>	<b>Месяц</b>
2	Иванов	15.05.50	6	58,95	5
3	Петров	22.04.80	4	28,99	4
4	Сидоров	11.05.84	3	24,94	5
5	Зайцев	01.02.89	1	20,21	2
6	Волков	07.05.78	4	30,95	5

Рис. 8.64 – Фрагмент листу Excel з списком доповненим 2 стовпцями

3. Виділити будь-яку комірку таблиці або всю таблицю з рядком заголовку.
4. Виконати команди **"Данные"** → **"Фильтр"** → **"Автофильтр"**.
5. Відкрити список кнопки автофільтру стовпця **"Месяц"** і вибрати значення 5. При цьому з екрану зникнуть 3 і 5 рядки.
6. Відкрити список кнопки автофільтру стовпця **"Возраст"**, вибрати пункт **"(Условие...)"** і у вікні, що відкриється, сформуванати умови **"больше или равно 20"** і **"меньше 31"** (рис. 8.65), поєднаних логічною операцією **"И"**. Натиснути кнопку **ОК**. При цьому залишаться тільки 4-й і 6-й рядки.

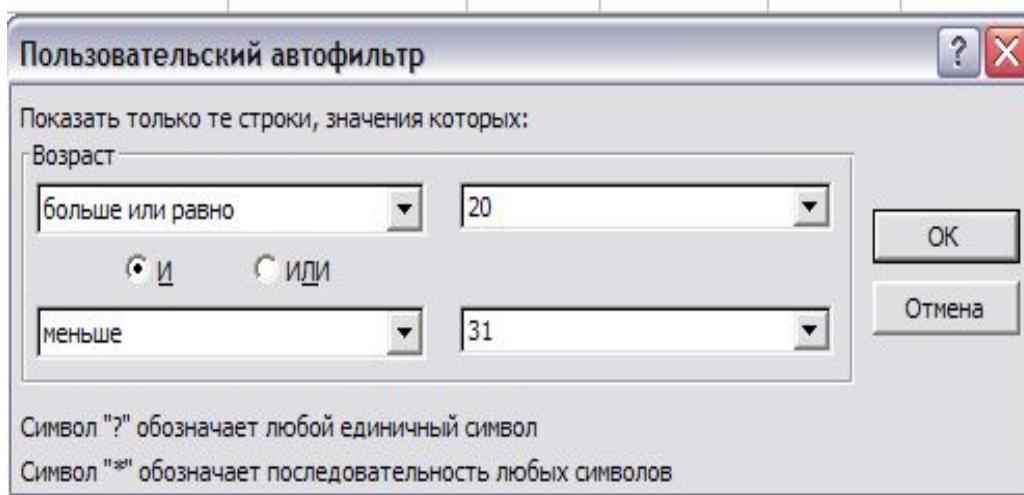


Рис. 8.65 – Вікно умов для кнопки авто фільтру стовпця "Возраст"

### 8.2.46.2 Розширений фільтр

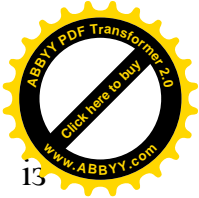
Розширений фільтр використовується в разі, якщо необхідно:

1. Задати умови, поєднані логічною операцією **"ИЛИ"** для декількох стовпців, а якщо декілька стовпців поєднані логічною операцією **"И"** - простіше використати автофільтр.
2. Задати 3 і більше умови відбору для конкретного стовпця з використанням принаймні однієї операції **"ИЛИ"** (наприклад, співробітники з прізвищем, що починається на А, Г або Н)
3. Задати обчислювані умови (наприклад, співробітників, з окладом на 25% більше середнього).

Команда **"Расширенный фильтр"** вимагає створення умов відбору рядків в окремому діапазоні робочого листа до застосування цієї команди.

Умови відбору створюються за наступними правилами:

1. Діапазон умов необхідно поміщати вище або нижче за список, оскільки при фільтруванні ховаються цілі рядки.
2. Діапазони умов повинні містити не менше 2 рядки. Причому у верхньому рядку має знаходитись один або декілька заголовків стовпців. Заголовки в діапазоні умов повинні в точності збігатися із



заголовками стовпців в списку, тому заголовки слід копіювати із заголовка списку. Винятком є заголовки обчислюваних умов, які можуть містити довільний текст або навіть бути порожніми.

3. В діапазон умов вводиться будь-яка кількість умов за наступними правилами:
  - умови в одному рядку вважаються пов'язаними логічною операцією "И", а на різних рядках – логічною операцією "ИЛИ";
  - щоб задати для окремого стовпця три або більше умов відбору, необхідно ввести умови в комірці, розташовані в суміжних рядках;
  - щоб вибрати рядки, що задовольняють одній з декількох умов, накладених на різні стовпці, необхідно ввести умови в комірці, розташовані в різних рядках діапазону умов;
  - щоб накласти складну умову відбору, необхідно ввести його складові частини в окремі стовпці діапазону умов;
  - щоб накласти умови відбору на декілька стовпців одночасно, необхідно ввести умови в комірці, розташовані в одному рядку діапазону умов.
4. При створенні текстових умов користуються наступними правилами:
  - єдина буква означає знайти всі значення, що починаються з цієї букви;
  - <, > означає знайти всі значення, які знаходяться за алфавітом до або після заданого символу;
  - ="Текст" означає знайти текст, точно співпадаючий з умовою, а якщо без знаку =, то всі рядки з входженням цього тексту.
5. При використанні обчислюваних умов користуються наступними правилами:
  - Заголовок над обчислюваною умовою повинен відрізнятися від будь-якого заголовка стовпця в списку. Заголовок умови може бути порожнім або містити довільний текст.
  - Посилання на комірці поза списком мають бути абсолютними
  - Посилання на комірці в списку мають бути відносними.

Після створення діапазону умов можна застосовувати розширений фільтр:

1. Скопіювати таблицю
2. Виділити копію таблиці або будь-яку її комірку.
3. Виконати команди: **"Данные"** → **"Фильтр"** → **"Расширенный фильтр"**.
4. З'явиться вікно показане на рис. 8.66

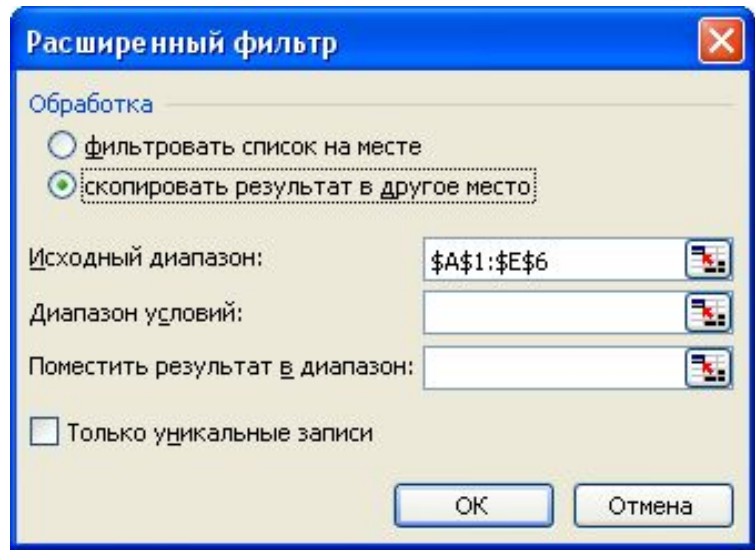


Рис. 8.66 – Вікно розширеного фільтру

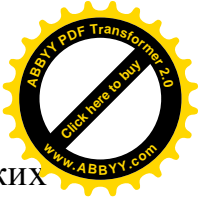
5. Встановити перемикач "филтровать список на месте" (якщо створена копія таблиці, а вхідним і вихідним діапазоном буде та сама копія таблиці) або "скопировать результат в другое место" (якщо не створена копія таблиці, а вихідним діапазоном буде інша вихідна таблиця).
6. Вказати мишею наступні діапазони: "Исходный диапазон", "Диапазон условий", і якщо потрібно - "Поместит результат в диапазон".
7. Натиснути .

Приклади діапазонів умов показані на рис. 8.67.

	A	B	C	D	E	F	G	H	I	J
1	ФИО		Оклад	Возраст		ФИО	Возраст		ФИО	Пол
2	И		>=150			И	>=40		И	м
3	С			<=40		С	>30			
4	Ш					Ш	>=25			
5										

Рис. 8.67 – Приклади діапазонів умов для розширеного фільтру

1. Діапазон умов A1:A4 дозволить відібрати співробітників, "ФИО" яких починається на "И", "С" або "Ш".
2. Діапазон умов C1:D3 дозволить відібрати співробітників, оклад яких не менше 150 або вік не більше 40.
3. Діапазон умов F1:G4 дозволить відібрати співробітників, "ФИО" яких починається на "И", з віком не менше 40, співробітників, "ФИО" яких починається на "С", з віком більше 30 або співробітників, "ФИО" яких починається на "Ш", з віком не менше 25.



4. Діапазон умов *I1:J2* дозволить відібрати співробітників, "ФІО" яких починається на "И", чоловічої статі.

Приклад. Розглядаються наступні дані (рис. 8.68).

Відібрати співробітників, оклад яких не менше 150 або вік не більше 20 років.

	А	В	С	Д	Е
1	Ф.И.О. рабочего	Дата рождения	Пол	Оклад	Возраст
2	Иванов	12.02.58	м	160	44
3	Петров	10.02.56	м	250	46
4	Сидоров	05.04.63	м	430	39
5	Матвеев	07.08.82	м	135	19
6	Шкуро	14.12.64	ж	128	37
7	Жмакин	26.06.78	м	140	23
8					

Рис. 8.68 – Фрагмент листу Excel з наданим списком

Для відбору потрібних співробітників необхідно виконати дії:

1. Для створення умов відбору над таблицею, вставити декілька нових рядків у верхній частині аркуша, наприклад 6. Тоді вихідна таблиця переміститься в діапазон *A7:E13*.
2. Створити діапазон умов (рис. 8.69)

С	Д
Оклад	Возраст
>=150	
	<=20

Рис. 8.69 – Фрагмент листу Excel з діапазоном умов

3. Виділити таблицю або будь-яку комірку таблиці.
4. Виконати команди: **"Данные"** → **"Фильтр"** → **"Расширенный фильтр"**.
5. Встановити перемикач *"скопировать результат в другое место"*, тому що попередньо не створена копія таблиці.
6. Вказати мишею параметри діалогового вікна **"Исходный диапазон"** – діапазон *A7:E13*, **"Диапазон условий"** – діапазон *C1:D3*, **"Поместить результат в диапазон"** – діапазон *A19:E19* (рис. 8.70).

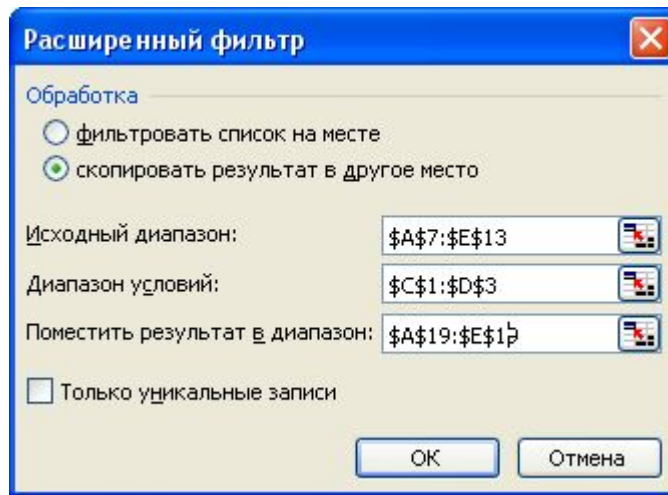


Рис. 8.70 – Приклад заповнення параметрів вікна розширеного фільтру

7. Натиснути кнопку **OK**

В результаті в комірках *A19:E23* з'явиться наступний результат, показаний на рис. 8.71.

	ФИО	Дата рождения	Пол	Оклад	Возраст
19					
20	Иванов	12.02.58	м	160	44
21	Петров	10.02.56	м	250	46
22	Сидоров	05.04.63	м	430	39
23	Матвеев	07.08.82	м	135	19

Рис. 8.71 – Фрагмент листу Excel з отриманим списком

### 8.2.47 Розрахунок проміжних підсумків в таблиці

Один із способів обробки і аналізу даних полягає в підведенні різних підсумків.

Якщо в будь-якому стовпці таблиці є дані, що повторюються, то таблицю можна відсортувати за даними цього стовпця, щоб дані, що повторюються, йшли підряд і склали групи. Тоді для кожної групи з даними, що повторюються, можна автоматично додати рядка проміжних підсумків. При цьому можна використовувати різні функції (СУМА, СЕРЕДНЕ...) для обчислення підсумків на рівні групи. Крім того, створюються загальні підсумки до всього списку.

**Зауваження!** Оскільки створення проміжних підсумків викликає вставку нових рядків з підсумками, то обчислення проміжних підсумків бажано здійснювати на копії початкової таблиці.



Для отримання проміжних підсумків необхідно:

1. Скопіювати таблицю
2. Відсортувати дані в копії таблиці за даними того стовпця, в якому повинні бути групи.
3. Виділити будь-який елемент таблиці або виділити таблицю разом з рядком заголовку (у таблиці не повинно бути об'єднаних комірок)
4. Виконати команди: **"Данные" → "Итоги..."**.
5. У вікні, що з'явиться (рис. 8.72) заповнити поля :

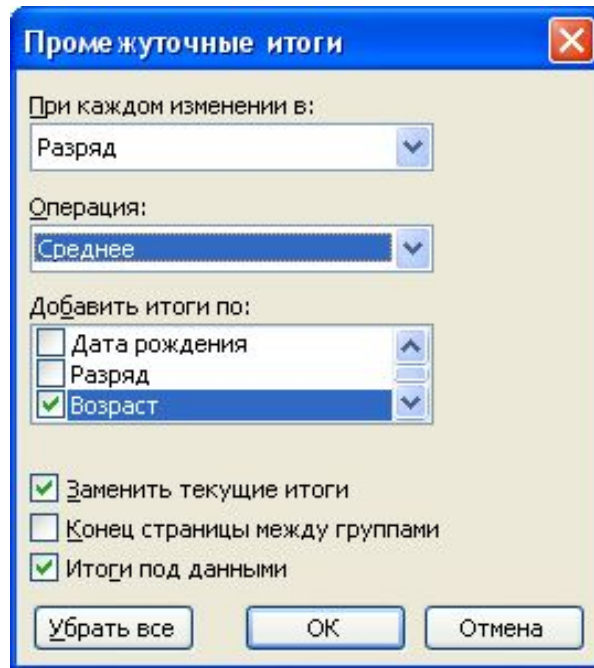
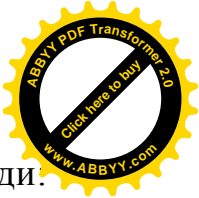


Рис. 8.72 – Вікно створення проміжних підсумків

- а) *"При каждом изменении в:"*, вибравши із списку, що розкриється, стовпець з групами, тобто стовпець, за даними якого попередньо відсортовані дані;
  - б) *"Операция:"*, вибравши функцію, яка використовуватиметься для обчислення проміжних підсумків ("Сумма", "Среднее", "Максимум", "Минимум", ...);
  - в) *"Добавить итоги по:"*, встановивши прапорці для назв тих стовпців, дані яких використовуватимуться для обчислень підсумків, і знявши непотрібні прапорці.
6. Натиснути кнопку **ОК**.

Щоб отримати декілька підсумкових показників з використанням різних підсумкових операцій в одних і тих же стовпцях, необхідно спочатку отримати один з підсумкових показників, а потім знову виконати команди **"Данные" → "Итоги..."**, тільки вибрати іншу операцію і зняти прапорець  *"Заменить текущие итоги"*. Якщо не зняти вказаний прапорець, то новий підсумковий показник замінить раніше створений.



Щоб видалити проміжні підсумки, необхідно виконати команди: виділити будь-яку комірку таблиці → **"Данные"** → **"Итоги..."** → натиснути кнопку .

Разом зі вставкою підсумкових рядків зліва на лист вставляються кнопки  або .

Окрім цих кнопок з'являться кнопки рівнів, клацанням по яких можна вивести відповідний рівень даних.

Приклад.

Розглядається таблиця з даними (табл.8.7).

Таблиця 8.7 – Таблиця з даними

№ ділянки	Рік	Здобич за рік
№1	2000	18
№2	2001	15
№3	1999	20
№1	1999	14
№2	2000	13
№3	2000	21
№1	2001	25
№2	1999	17

I. Знайти загальну здобич за кожен рік .

Для отримання результатів необхідно виконати такі дії:

- 1) Створити копію початкової таблиці.
- 2) Відсортувати копію таблиці за даними стовпця **"Рік"**,
- 3) Виконати команди: **"Данные"** → **"Итоги..."**,
- 4) В полі **"При каждом изменении в:"** вибрати **"Рік"**,
- 5) В полі **"Операция:"** вибрати **"Сумма"**,
- 6) В полі **"Добавить итоги по:"** встановити прапорець  **"Здобич за рік"** (рис. 8.73)
- 7) Натиснути .

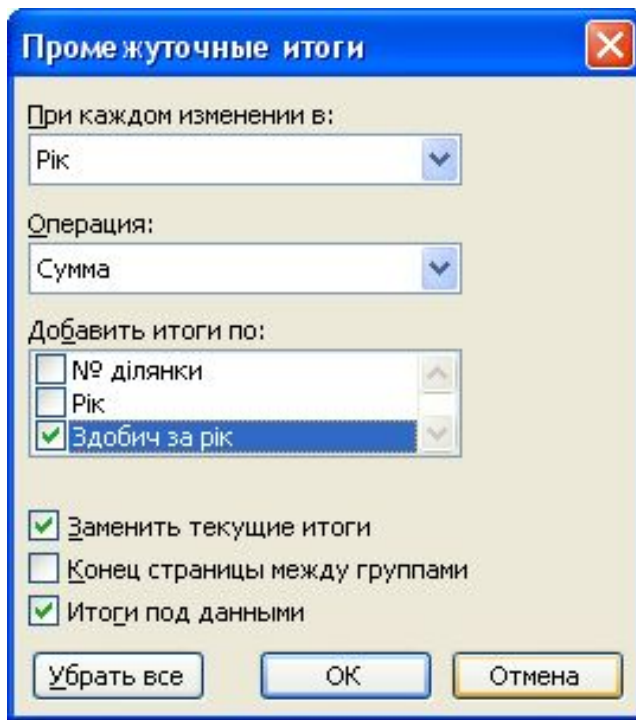


Рис. 8.73 – Приклад створення проміжних підсумків для завдання I

В результаті копія таблиці прийме вигляд, показаний на рис. 8.74.

	А	В	С
11			
12	<b>№ ділянки</b>	<b>Рік</b>	<b>Здобич за рік</b>
13	№3	1999	20
14	№1	1999	14
15	№2	1999	17
16		<b>1999 Итог</b>	51
17	№1	2000	18
18	№2	2000	13
19	№3	2000	21
20		<b>2000 Итог</b>	52
21	№2	2001	15
22	№1	2001	25
23		<b>2001 Итог</b>	40
24		<b>Общий итог</b>	143
25			

Рис. 8.74 – Вікно результату виконання завдання I



II. Знайти середню і максимальну здобич кожної ділянки.

Для отримання результатів необхідно виконати такі дії:

- 1) Створити ще одну копію початкової таблиці.
- 2) Відсортувати нову копію таблиці по стовпцю "№ ділянки",
- 3) Виконати команди: "Данные" → "Итоги...",
- 4) В полі "При каждом изменении в:" вибрати "№ ділянки",
- 5) В полі "Операция:" вибрати "Среднее",
- 6) В полі "Добавить итоги по:" встановити прапорець  "Здобич за рік" (рис. 8.75)
- 7) Натиснути .
- 8) Знову виконати команди: "Данные" → "Итоги...",
- 9) у полі "Операция:" вибрати "Максимум",
- 10) зняти прапорець  "Заменит текущие итоги".
- 11) натиснути .

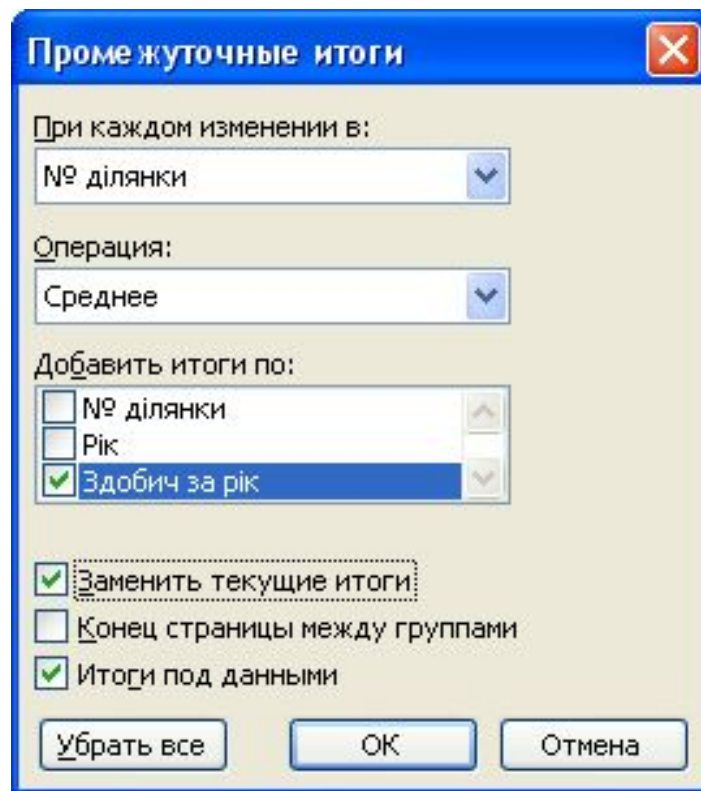


Рис. 8.75 – Приклад створення проміжних підсумків для завдання II



В результаті копія таблиці прийме вигляд, показаний на рис. 8.76.

		A	B	C	D
28	№1		2000	18	
29	№1		1999	14	
30	№1		2001	25	
31	<b>№1 Середнє</b>			19	
32	<b>№1 Максимум</b>			25	
33	№2		2001	15	
34	№2		2000	13	
35	№2		1999	17	
36	<b>№2 Середнє</b>			15	
37	<b>№2 Максимум</b>			17	
38	№3		1999	20	
39	№3		2000	21	
40	<b>№3 Середнє</b>			20,5	
41	<b>№3 Максимум</b>			21	
42	<b>Общее среднее</b>			17,875	
43	<b>Общий максимум</b>			25	
44					

Рис. 8.76 – Вікно результату виконання завдання II

### 8.2.48 Створення зведених таблиць

Зведені таблиці – один з найефективніших засобів обробки і компактного представлення даних. Структуру зведеної таблиці легко видозмінювати.

Зведена таблиця створюється за допомогою *Майстра зведених таблиць*. Щоб викликати *Майстер зведених таблиць*, необхідно виконати команди: "Данные" → "Сводная таблица...".

Для створення зведеної таблиці необхідно виконати такі дії:

1. Виділити будь-яку комірку таблиці або всю таблицю разом з рядком заголовку (у таблиці не повинно бути об'єднаних комірок).
2. Виконати команди: "Данные" → "Сводная таблица...". З'явиться вікно першого кроку *Майстра зведених таблиць* (рис. 8.77).

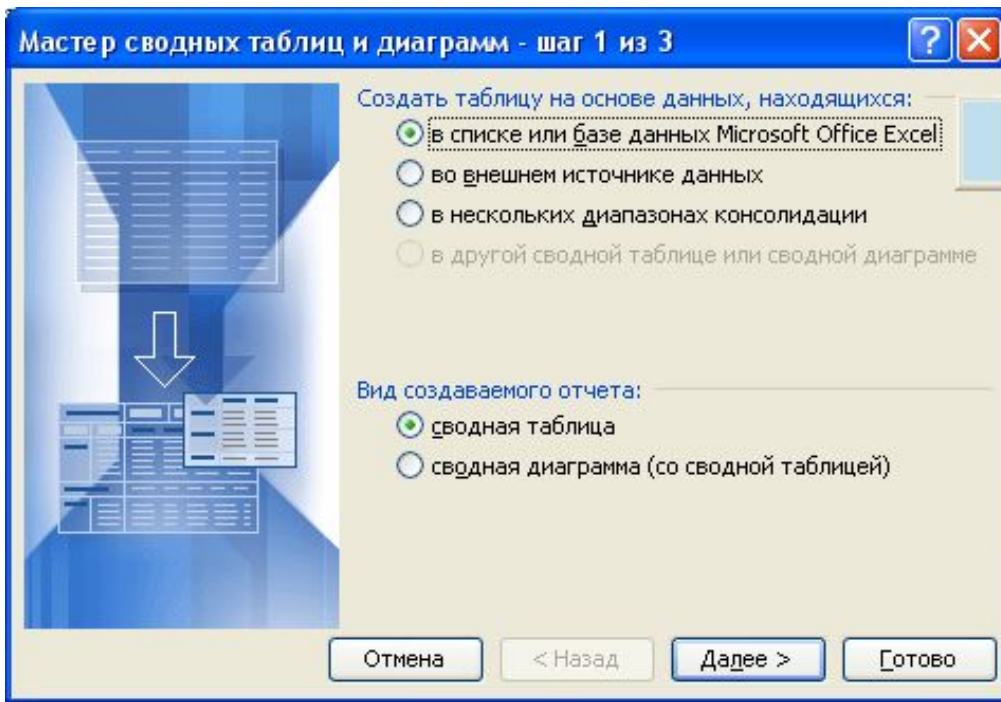


Рис. 8.77 – Вікно 1-го кроку Майстра зведених таблиць

3. Переконайтеся, що встановлені перемикачі "в списке или базе данных Microsoft Office Excel" ( в групі "Создать таблицу на основе данных, находящихся:") і "сводная таблица" ( в групі "Вид создаваемого отчета:"), а потім натиснути кнопку **Далее**.
4. З'явиться вікно другого кроку **Майстра зведених таблиць** (рис. 8.78) для визначення діапазону з наданими даними. І так як таблиця попередньо вже була виділена, то в полі "Диапазон" можна буде побачити координати початку і кінця діапазону з наданими даними. Залишіться натиснути кнопку **Далее** або перевизначити місцезнаходження таблиці з наданими даними і натиснути кнопку **Далее**.

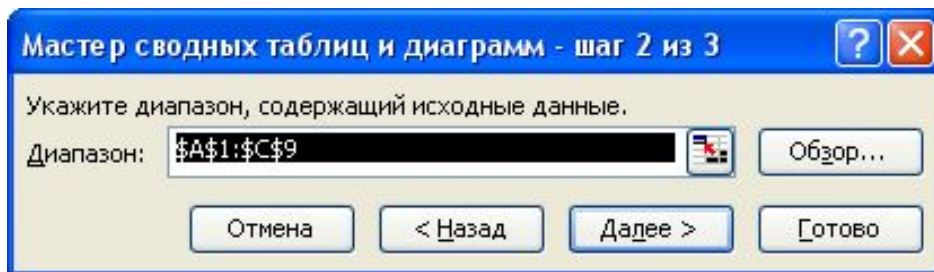


Рис. 8.78 – Вікно 2-го кроку Майстра зведених таблиць

5. З'явиться вікно третього кроку **Майстра зведених таблиць** (рис. 8.79) для визначення місцезнаходження зведеної таблиці. Для конструювання вигляду зведеної таблиці необхідно натиснути кнопку **Макет...**



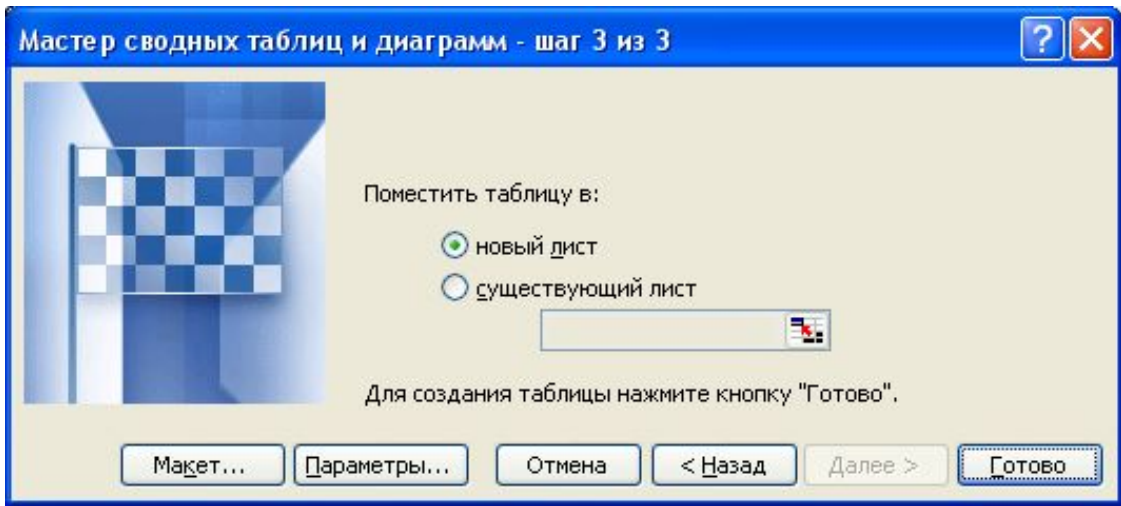


Рис. 8.79 – Вікно 3-го кроку Майстра зведених таблиць

6. З'явиться вікно конструювання макету зведеної таблиці (рис. 8.80). У правій частині вікна знаходяться кнопки з іменами стовпців таблиці наданих даних. Імена цих стовпців можна перетягувати в області **"Строка"**, **"Столбец"**, **"Страница"** і **"Данные"** макету зведеної таблиці. Поля, що опинилися в області **"Строка"**, утворюють рядки зведеної таблиці. Поля, що опинилися в області **"Столбец"**, стануть її стовпцями. А ті стовпці, що опинилися області **"Данные"**, будуть підсумовані функцією СУММ. Щоб використовувати іншу функцію можна двічі клацнути мишею на імені поля в області **"Данные"**, вибрати із списку іншу функцію і натиснути кнопку **ОК**. Область **"Страница"** призначена для створення багатовимірних таблиць.

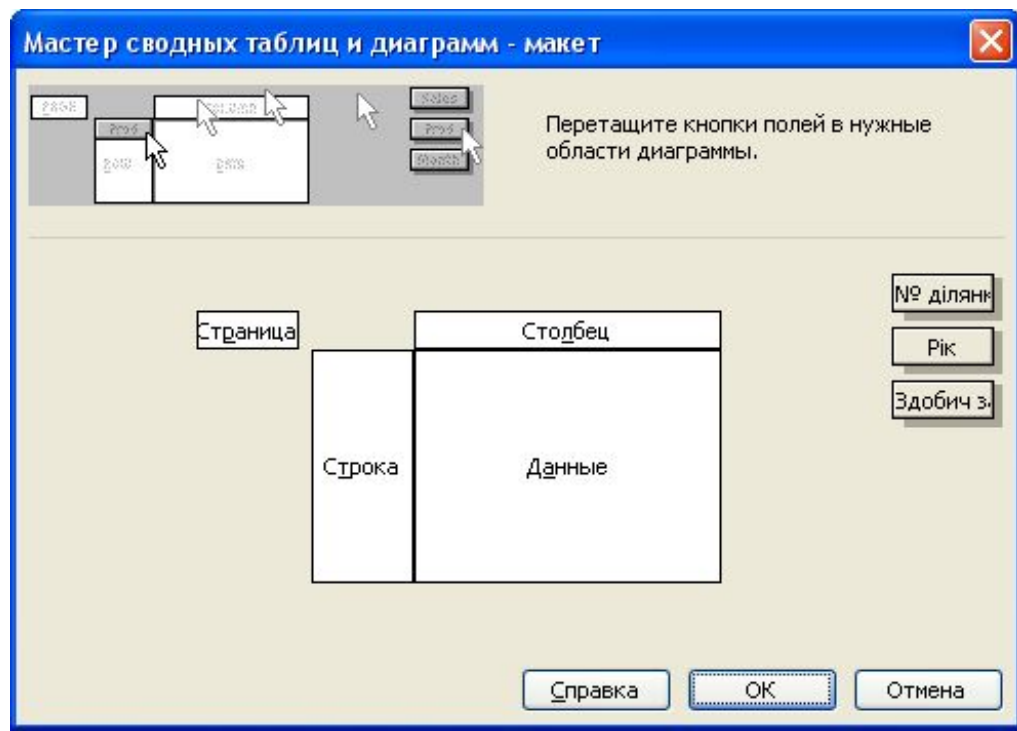


Рис. 8.80 – Вікно макету Майстра зведених таблиць

7. Після завершення перетягування полів в потрібні області макету необхідно натиснути кнопку **ОК**.
8. Знову з'явиться вікно третього кроку *Майстра зведених таблиць* (рис. 8.78) для вказівки, де створювати зведену таблицю: 1) на новому листі або 2) на існуючому. Встановити один із запропонованих перемикачів (а для другого вказати ще місцеположення лівого верхнього кута зведеної таблиці) і натиснути кнопку **Готово**.

Зазвичай Excel для полегшення роботи із зведеною таблицею відображає панель інструментів *"Сводные таблицы"* і *"Список полей сводной таблицы"* (рис.8.81).

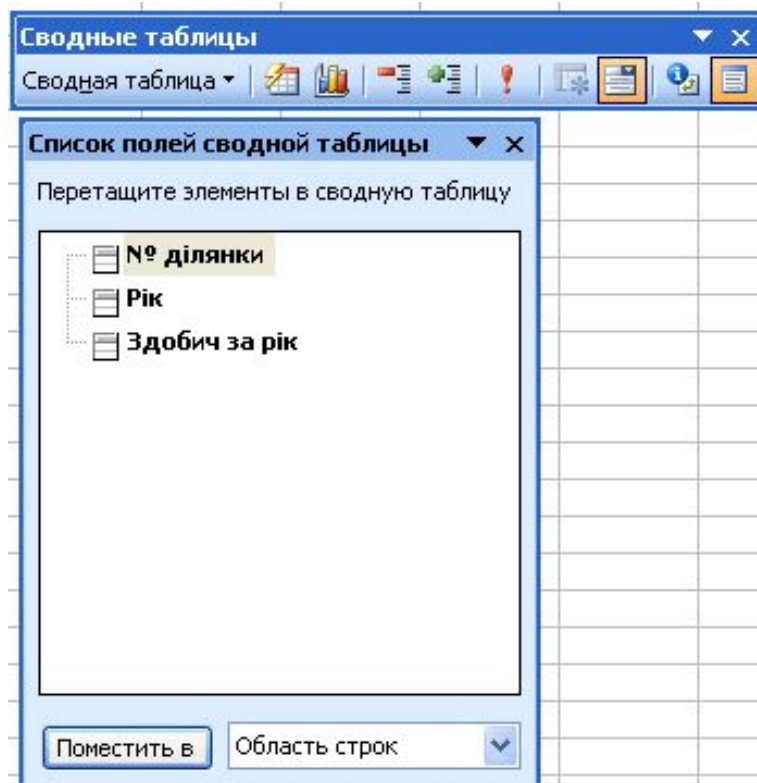


Рис. 8.81 – Панель інструментів "Сводные таблицы"

**Зауваження!** Якщо потрібно отримати підсумкові показники тільки при одному розбитті на групи (наприклад, тільки по роках), то в макеті зведеної таблиці область *"Столбец"* можна залишити порожньою.

#### 8.2.49 Перестановка полів зведеної таблиці

Щоб переставити поля в зведеній таблиці достатньо спочатку встановити курсор в будь-який елемент зведеної таблиці, а потім на панелі інструментів *"Сводные таблицы"* відкрити меню *"Сводная таблица ▼"* (рис. 8.82) і вибрати пункт *"Мастер сводных таблиц"*.

З'явиться вікно третього кроку *Майстра зведених таблиць* (рис. 8.79), де треба натиснути кнопку **Макет...** і перетягнути поля на нові позиції, а потім натиснути **ОК** і **Готово**.

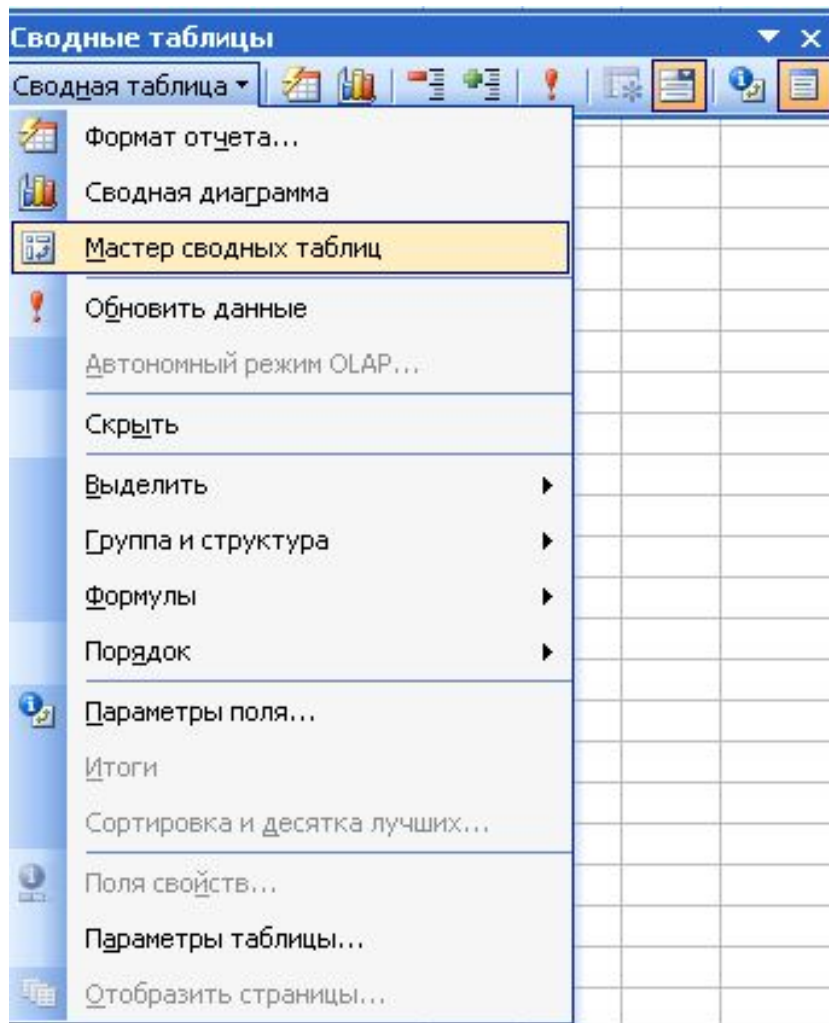


Рис. 8.82 – Пункт меню "Сводная таблица" панелі "Сводные таблицы"


Переставити поля в зведеній таблиці можна і в інший спосіб. Для цього достатньо спочатку встановити курсор в будь-який елемент зведеної таблиці, а потім з панелі *"Список полей сводной таблицы"* по черзі перетягнути поля на нові позиції.

В той час, коли здійснюється реорганізація зведеної таблиці, вона автоматично перераховується. Реорганізація зведеної таблиці не впливає на початкові дані.

### 8.2.50 Зміна функції в зведеній таблиці

За умовчанням майстер зведених таблиць використовує для обробки значень в області *"Данные"* функцію підсумовування, але замість неї легко можна перейти до обчислення іншої функції (*наприклад*, середнього, максимального, мінімального і т.д. значення).

Для зміни функції в зведеній таблиці необхідно:

1. Виділити будь-яку комірку з даними зведеної таблиці або лівий верхній елемент зведеної таблиці.
2. На панелі інструментів **"Сводные таблицы"** натиснути кнопку **"Параметры поля"** або на панелі інструментів  **"Сводные таблицы"** відкрити меню **"Сводная таблица ▼"** (рис. 8.81) і вибрати пункт **"Параметры поля..."** або двічі клацнути на лівому верхньому елементі зведеної таблиці.
3. З'явиться вікно **"Вычисление поля сводной таблицы"** (рис. 8.83). Тут в полі "Операция" можна вибрати з пропонуваного списку іншу функцію.

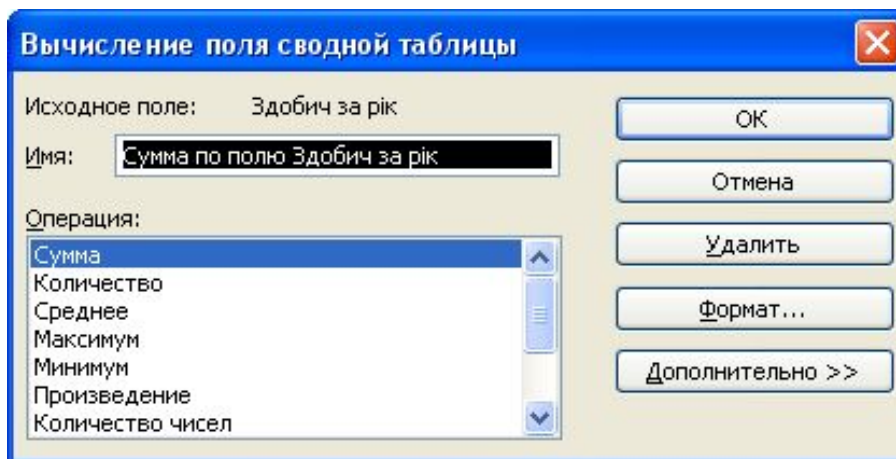



Рис. 8.83 – Вікно "Вычисление поля сводной таблицы"

4. Клацнути по кнопці **ОК**.

### 8.2.51 Оновлення даних в зведеній таблиці

При зміні даних в початковій таблиці, в зведеній таблиці дані не перераховуються автоматично.

Для перерахунку  зведеної таблиці необхідно виділити будь-який елемент зведеної таблиці, а потім на панелі інструментів **"Сводные таблицы"** натиснути кнопку **"Обновить данные"** або на панелі інструментів **"Сводные таблицы"** відкрити меню **"Сводная таблица ▼"** (рис. 8.81) і вибрати пункт **"Обновить данные"**.

**Зауваження:** Якщо в початковій таблиці додані нові рядки, то для включення цих даних в зведену таблицю, необхідно викликати **Майстер зведених таблиць**, повернутися до 2-го діалогового вікна за допомогою кнопки **<Назад** і наново вказати діапазон з початковою таблицею, а потім натиснути кнопку **Готово**.

Приклад.

Розглядається таблиця з даними (табл.8.7). Знайти загальну здобич кожної ділянки за кожен рік .

Для отримання результатів необхідно виконати такі дії:

1. Виділити будь-яку комірку таблиці або всю таблицю разом з рядком заголовку.
2. Виконати команди: **"Данные"** → **"Сводная таблица..."**. З'явиться вікно першого кроку **Майстра зведених таблиць** (рис. 8.76).
3. Переконайтеся, що встановлені перемикачі **"в списке или базе данных Microsoft Office Excel"** ( в групі **"Создать таблицу на основе данных, находящихся:"**) і **"сводная таблица"**( в групі **"Вид создаваемого отчета:"**), а потім натиснути кнопку **Далее**.
4. З'явиться вікно другого кроку **Майстра зведених таблиць** (рис. 8.77) для визначення діапазону з наданими даними. І так як таблиця попередньо вже була виділена, то в полі **"Диапазон"** можна буде побачити координати початку і кінця діапазону з наданими даними. Залишиться натиснути кнопку **Далее**.
5. З'явиться вікно третього кроку **Майстра зведених таблиць** (рис. 8.78) для визначення місцезнаходження зведеної таблиці. Для конструювання вигляду зведеної таблиці необхідно натиснути кнопку **Макет...**
6. З'явиться вікно конструювання макету зведеної таблиці (рис. 8.79). У правій частині вікна знаходяться кнопки з іменами стовпців таблиці наданих даних. Ім'я стовпця **"Рік"** можна перетягнути в область **"Столбец"**, ім'я стовпця **"№ ділянки"** можна перетягнути в область **"Строка"**, а ім'я стовпця **"Здобич за рік"** можна перетягнути в область **"Данные"** макету зведеної таблиці (рис. 8.84).

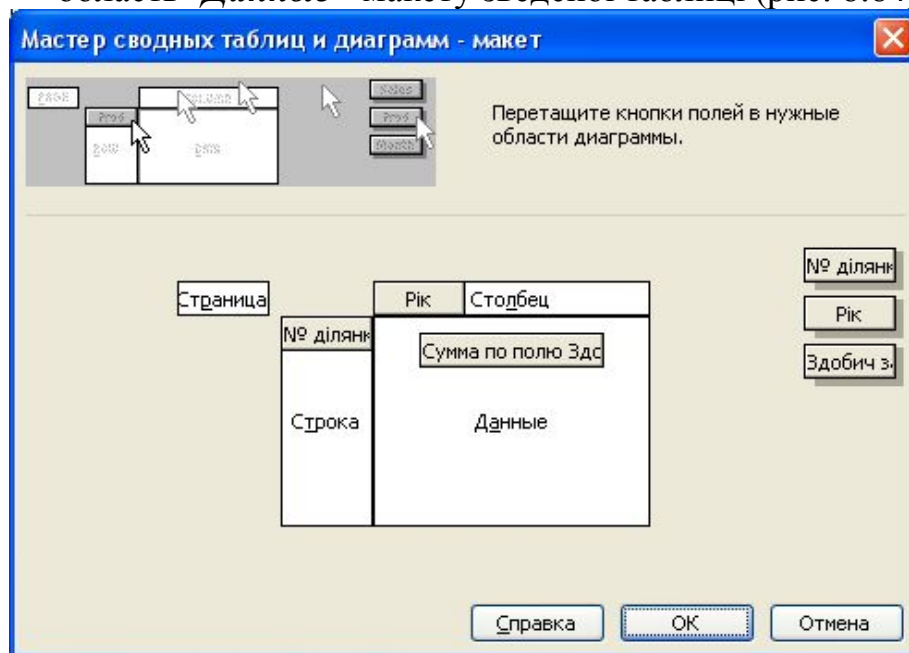


Рис. 8.84 – Макет зведеної таблиці



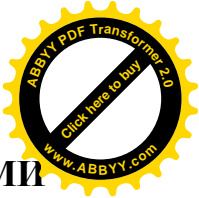
- Після завершення перетягування полів в потрібні області макету необхідно натиснути кнопку **ОК**.
- Знову з'явиться вікно третього кроку *Майстра зведених таблиць* (рис. 8.78). Встановити перемикач "новый лист" і натиснути кнопку **Готово**.

В результаті буде створено новий лист (рис. 8.85).

	A	B	C	D	E	F
1	Перетащите сюда поля страниц					
2						
3	Сумма по полю Добич за рік	Рік				
4	№ ділянки	1999	2000	2001	Общий итог	
5	№1	14	18	25	57	
6	№2	17	13	15	45	
7	№3	20	21		41	
8	Общий итог	51	52	40	143	
9						
10	<b>Сводные таблицы</b>					
11	Сводная таблица					
12						
13	<b>Список полей сводной таблицы</b>					
14	Перетащите элементы в сводную таблицу					
15						
16	№ ділянки					
17	Рік					
18	Здобич за рік					
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29	Поместить в					
30	Область строк					
31						

Рис. 8.85 –Фрагмент листу Excel зі зведеною таблицею





## 9 ПРОГРАМНІ ЗАСОБИ РОБОТИ З БАЗАМИ ТА СХОВИЩАМИ ДАНИХ

База даних (БД) означає сукупність даних, призначених для спільного використання.

Розглянемо реляційну модель бази даних, у якій база даних представляється у виді однієї або декількох однорідних таблиць.

Для рішення задач підприємства потрібно всі дані упорядкувати по об'єктах і рознести по таблицях і запитам бази даних. Цей етап інакше називається нормалізацією, у результаті якого виробляється найбільш ефективний і гнучкий спосіб збереження даних.

Нормалізація даних – це теоретична і практична процедура звільнення таблиць бази даних від зайвих даних.

Первинним ключем називається поле (або кілька полів), значення якого не повторюється в жодному записі даної таблиці.

### 9.1 Використання СУБД ACCESS для обробки даних

#### 9.1.1 Правила нормалізації таблиці

Правило 1. Кожне поле будь-якої таблиці повинне бути унікальним, тобто не може бути полів з однаковими іменами.

Правило 2. Кожна таблиця повинна мати унікальний ідентифікатор, або первинний ключ, що може складатися з одного або декількох полів, тобто значення в ключовому полі повинно бути унікальним.

Правило 3. Для кожного значення первинного ключа значення в стовпцях даних повинні відповідати об'єкту таблиці і цілком його описувати.

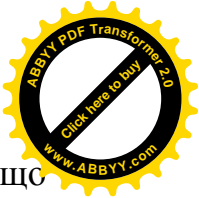
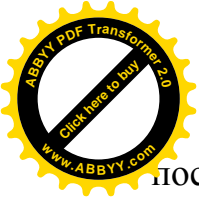
Правило 4. Користувач повинний мати можливість змінювати значення кожного поля (не вхідного в первинний ключ) без впливу на дані інших полів.

Приклад. Для ведення обліку постачання матеріалів створити базу даних, що містить наступну інформацію:

- дата постачання матеріалу;
- найменування матеріалу;
- кількість поставленого матеріалу;
- ціна матеріалу за одиницю;
- найменування організації постачальника;
- місто місцезнаходження постачальника;
- контактний телефон.

Здійсимо нормалізацію даних.

1. Щоб уникнути повторень інформації про міста при введенні даних про



постачальників з того самого міста, варто створити окрему таблицю, що містить зведення про міста, установлюючи при цьому кожному місту окремий числовий код. Структуру таблиці можна представити в такий спосіб:

*МІСТО(Код\_міста, Місто)*

У цій таблиці поле "Код\_міста" є первинним ключем, тому що повторних значень цього поля в таблиці не повинно бути.

2. Щоб уникнути повторення інформації про постачальників, у випадку декількох постачань від того самого постачальника, необхідно створити окрему таблицю, у якій помістити зведення про кожного постачальників, передбачаючи для кожного постачальника окремий числовий код. Структуру відносин у цій таблиці можна записати в такий спосіб:

*ПОСТАЧАЛЬНИК(Код\_постачальника, Постачальник, Телефон,  
Код\_міста)*

Первинним ключем цієї таблиці є поле "Код\_постачальника", тому що жодне значення цього поля повторюватися не повинне.

3. Щоб уникнути повторень інформації про матеріали при введенні даних про постачання того самого матеріалу, варто створити окрему таблицю, що містить зведення про матеріали, установлюючи при цьому кожному матеріалові окремий числовий код. Якщо ціна матеріалу не залежить від дати постачання і постачальника, структуру таблиці можна представити в такий спосіб:

*МАТЕРІАЛ(Код\_матеріалу, Матеріал, Ціна\_за\_одиницю)*

У цій таблиці поле "Код\_матеріалу" є первинним ключем, тому що повторних значень цього поля в таблиці не повинно бути.

4. Найбільше часто змінюваною і поповнюваною буде таблиця про постачання матеріалів із вказівкою дати постачання, матеріалу, кількості поставленого матеріалу, постачальника. Для кожного запису цієї таблиці теж можна вказати унікальний числовий код. Структуру відносин у цій таблиці можна записати в такий спосіб:

*ПОСТАЧАННЯ(Код\_постачання, Дата\_постачання, Код\_матеріалу,  
Кількість\_матеріалу, Код\_постачальника)*

Первинним ключем цієї таблиці є поле "Код\_постачання", тому що жодне значення цього поля повторюватися не повинне. Поля "Код\_матеріалу" і "Код\_постачальника" є чужим ключем, тому що кожний з них є ключем іншої таблиці. У таблиці ПОСТАЧАННЯ значення кожного з них можуть повторюватися, тобто однакових значень кожного з цих полів може бути багато.



### 9.1.2 Типи даних у СУБД ACCESS

ACCESS підтримує типи даних, приведені в таблиці 9.1.

Для полів можна вказати такі властивості: розмір, формат, число десяткових знаків, маску вводу, підпис, значення за умовчанням, умову на значення, повідомлення про помилку, чи є поле обов'язковим, чи є поле індексованим.

Таблиця 9.1 – Типи даних

Тип даних	Застосування
Текстовий	Містить будь-які символи – букви, цифри і спеціальні символи
Числовий	Числові дані
Дата / Час	Дати і час
Грошовий	Містить числа, що будуть формуватися у виді грошових сум і зазвичай містять 2 десяткових знаки
Лічильник	Автоматично вставляє в поле унікальні значення довгого цілого числа при кожному створенні нового запису. Використовується для генерації значень первинного ключа.
Логічний	Логічні дані "Так/Ні", "Істина/Неправда". Значення може бути представлене на формі у виді прапорця.
Поле MEMO	Містить символи тексту обсягом до 64000 байт. Використовується для збереження великих текстових фрагментів
Поле об'єкта OLE	Картинки, діаграми й інші об'єкти OLE з додатків WINDOWS
Майстер підстановок	Змушує поле приймати лише ті дані, що заздалегідь перераховані в списку або в іншій таблиці

Встановимо тип і властивості полів для таблиць МІСТО, ПОСТАЧАЛЬНИК, МАТЕРІАЛ, ПОСТАЧАННЯ, як показано в таблицях 9.2 – 9.5.

Таблиця 9.2 - МІСТО

Ім'я поля	Тип поля	Розмір	Формат	Ключове поле (*), чужий ключ або звичайне поле
Код_міста	лічильник			*
Місто	текстовий	20		



Таблиця 9.3 - ПОСТАЧАЛЬНИК

Ім'я поля	Тип поля	Розмір	Формат	Ключове поле (*), чужий ключ або звичайне поле
Код_постачальника	лічильник			*
Постачальник	текстовий	20		
Телефон	текстовий	10		
Код_міста	числовий (Майстер підстановки)	Довге ціле		Чужий ключ з таблиці МІСТО

Таблиця 9.4 - МАТЕРІАЛ

Ім'я поля	Тип поля	Розмір	Формат	Ключове поле (*), чужий ключ або звичайне поле
Код_матеріалу	лічильник			*
Матеріал	текстовий	20		
Ціна за одиницю	грошовий		грошовий	

Таблиця 9.5 - ПОСТАЧАННЯ

Ім'я поля	Тип поля	Розмір	Формат	Ключове поле (*), чужий ключ або звичайне поле
Код_постачання	лічильник			*
Дата_постачання	дата/час		короткий формат дати	
Код_матеріалу	Числовий (Майстер підстановок)	довге ціле		Чужий ключ з таблиці МАТЕРІАЛИ
Кількість_матеріалу	числовий	з крапкою, що плаває, (4 байт)		
Код_постачальника	Числовий (Майстер підстановок)	довге ціле		Чужий ключ з таблиці ПОСТАЧАЛЬНИК

### 9.1.3 Зв'язки таблиць БД

Дві таблиці можуть бути зв'язані відношенням "один-до-багатьох" або "один-до-одного". Зв'язок "один-до-одного" (1 до 1) означає, що будь-який запис однієї таблиці зв'язана тільки з одним записом іншої таблиці.

Встановимо зв'язок таблиць ПОСТАЧАННЯ, МАТЕРІАЛ, ПОСТАЧАЛЬНИК і МІСТО.

Таблиця МІСТО може бути зв'язана з таблицею "ПОСТАЧАЛЬНИК" по полю "Код\_міста" відношенням "один-до-багатьох", тому що можуть бути постачальники з того самого міста.

Таблиця МАТЕРІАЛ може бути зв'язана з таблицею ПОСТАЧАННЯ по полю "Код\_матеріалу" відношенням "один-до-багатьох", тому що той самий матеріал може бути поставлений у декількох постачаннях.

Таблиця ПОСТАЧАЛЬНИК може бути зв'язана з таблицею ПОСТАЧАННЯ по полю "Код\_постачальника" відношенням "один-до-багатьох", тому що той самий постачальник може брати участь у декількох постачаннях".

Схематично зв'язки таблиць можна представити у вигляді, що показаний на рис. 9.1.

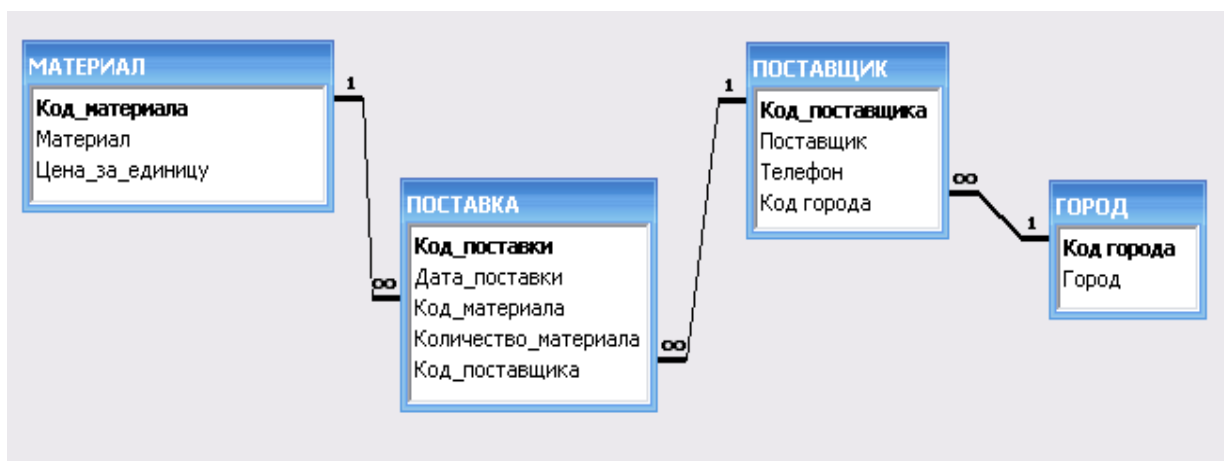


Рис. 9.1 – Схема даних

## 9.2 Створення бази даних

Для запуску додатка ACCESS і створення нової бази даних варто виконати наступні дії: "Пуск" → "Програми" → "MS ACCESS" → "Нова база даних",  ОК . → вказати ім'я бази даних і папку, у якій цю базу зберегти, і натиснути кнопку  Сохранить.

## 9.3 Створення таблиць в базі даних

Створення таблиці в базі даних здійснюється в два етапи:



1 етап. Створення структури таблиці.

2 етап. Введення даних у таблицю.



### 9.3.1 Створення структури таблиці

Для створення структури таблиці необхідно:

1. У вікні створеної бази даних на вкладці "Таблицы" клацнути по кнопці Створити , а потім вибрати пункт "Конструктор".
2. Далі варто по черзі занести зведення про кожне поле таблиці ("Имя поля" і "Тип данных"). Ім'я поля може містити символи латинського або російського алфавіту, арабські цифри і знак підкреслення. Тип поля вибирається з падаючого списку стандартних типів в комірці стовпця "Тип данных".
3. Коли тип поля встановлений, то можна установити властивості поля в нижній частині вікна. Для цього у вікні вкладки "Общие" установити курсор в комірку рядка, що містить установлювану властивість і в списку властивостей, що розкривається, що з'явився, вибрати потрібну властивість.
4. Якщо в таблиці є поле, що є чужим ключем, то необхідно в стовпці "Тип поля" цього поля скористатися майстром підстановок (див. нижче). Але структура таблиці, з якої буде здійснюватися підстановка, вже повинна бути створена.
5. Указати, яке поле буде ключовим. Для цього необхідно клацнути в області виділення, розташованої ліворуч від імені потрібного поля. Потім у меню "Правка" вибрати команду "Ключевое поле" або клацнути на панелі інструментів по кнопці  . При цьому ліворуч від імені обраного поля з'явиться символ ключа .
6. Після занесення зведень про всі поля таблиці необхідно закрити вікно конструктора ( і підтвердити збереження, натиснувши кнопку  ).
7. З'явиться вікно з запитом імені таблиці, у якому необхідно набрати ім'я створюваної таблиці і зберегти структуру таблиці.

### 9.3.2 Використання Майстра Підстановок

Властивість "Підстановка" використовується для заповнення значень поля таблиці з фіксованого списку або з іншої таблиці, якщо це поле є чужим ключем, тобто є ключем іншої таблиці. У цьому випадку зручно застосовувати "Майстер підстановок".

**Примітка!** Перед підключенням Майстра підстановок для поля таблиці треба:

- створити структуру таблиці, ключом якої є це поле.
- не встановлювати зв'язок поточної таблиці по цьому полю з іншою таблицею, або видалити зв'язок, якщо



Щоб підключити підстановку для поля таблиці варто виконати наступні дії (наприклад, у таблиці *ПОСТАЧАННЯ* для поля *Код\_матеріалу*):

1. Відкрити таблицю, у якій створюється поле з властивістю *Підстановка* в режимі конструктора (для нашого приклада, таблицю *ПОСТАЧАЛННЯ*).

2. Установити курсор в комірку стовпця "Тип данных" поля, для якого підключається підстановка (для нашого приклада, поля *Код\_матеріалу*), і вибрати зі списку, що розкривається, пункт "Мастер подстановок".

3. У вікні, що з'явилося, активізувати пункт "Таблица или... запрос..." і натиснути кнопку

4. У вікні, що з'явилося (рис. 9.2), варто вибрати таблицю, з якої будуть підставлятися значення поля (для нашого приклада, таблицю *МАТЕРІАЛ*), і натиснути кнопку

5. У вікні, що з'явилося, необхідно помітити спочатку ключове поле (для нашого приклада, поле *Код\_матеріалу*) і натиснути кнопку . А потім виділяти поля, що пояснюють ключове поле (для нашого приклада, поле *Матеріал*) і кнопкою  поміщати їх у праву частину, а потім клацнути по кнопці .

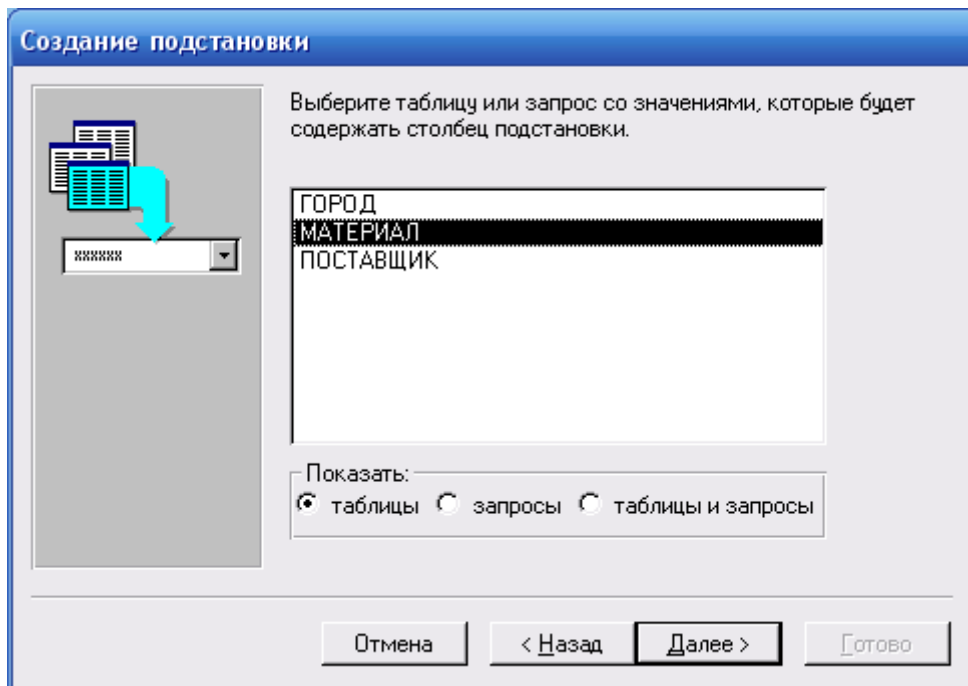


Рис. 9.2 – Вікно вибору таблиці-джерела в Майстрі Підстановок

6. З'явиться вікно, показане на рис. 9.3, у якому клацнути по кнопці **Далее**.

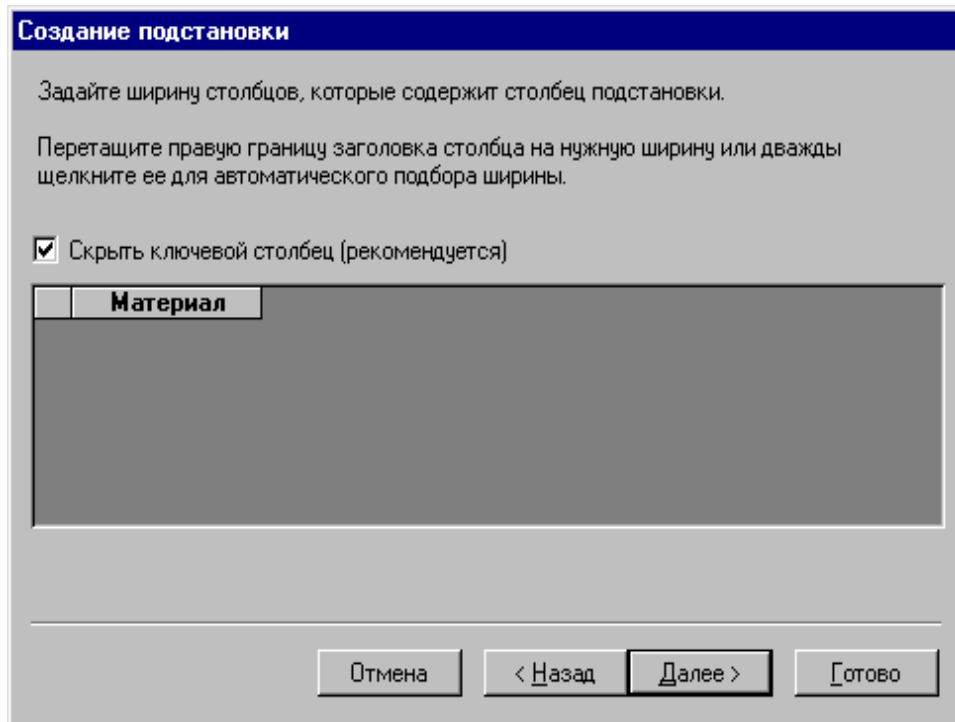



Рис. 9.3 – Вікно Майстра Підстановок

7. У вікні, що з'явиться, буде запропоновано ім'я поля і якщо ім'я зазначене вірно, клацнути по кнопці **Готово**.

### 9.3.3 Установка і видалення зв'язку таблиць бази даних

Зв'язок таблиць можна здійснити у вікні "Схема данных".

Вікно "Схема данных" можна відкрити:

- 1) за допомогою команди **"Сервис" → "Схема данных"** або
- 2) за допомогою кнопки  на панелі інструментів.

При першому виконанні цієї команди крім вікна "Схема данных", на екрані з'являється вікно "Добавление таблицы" (рис. 9.4), за допомогою якого варто помістити потрібні таблиці. Для цього у вікні "Добавление таблицы" впливає по черзі виділяти імена таблиць, що зв'язуються, і натискати кнопку **Добавить**. Обрані таблиці з'являться у вікні "Схема данных". Щоб у вікні "Схема данных" розташувати таблиці зручно, можна переміщати них, схопивши за заголовок і перетаскуючи мишею.

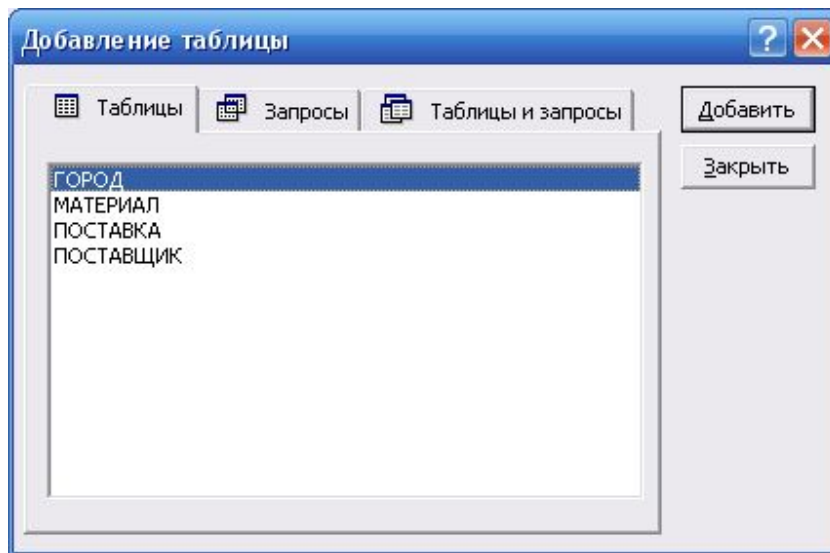


Рис. 9.4 – Вікно "Добавление таблицы"

**Примітка!** Якщо потрібно додати таблицю у вікно "Схема даних", а вікно "Добавление таблицы" закрито, тобто його немає на екрані, то для його відкриття варто викликати меню правим клацанням кнопки миші на вільному місці робочої частини вікна "Схема даних" і вибрати пункт "Добавить таблицу".

Щоб створити зв'язок між двома таблицями потрібно: схопити лівою кнопкою миші поле таблиці, по якому встановлюється зв'язок і протягти його до поля, що зв'язується з ним, в іншій таблиці і відпустити кнопку миші, а потім у вікні, що з'явилося, "Зв'язку" активізувати всі три режими:

- Забезпечення цілісності даних;*
- Каскадне відновлення зв'язаних полів;*
- Каскадне видалення зв'язаних записів,*

а потім і підтвердити, клацнувши по кнопці **Создать**.

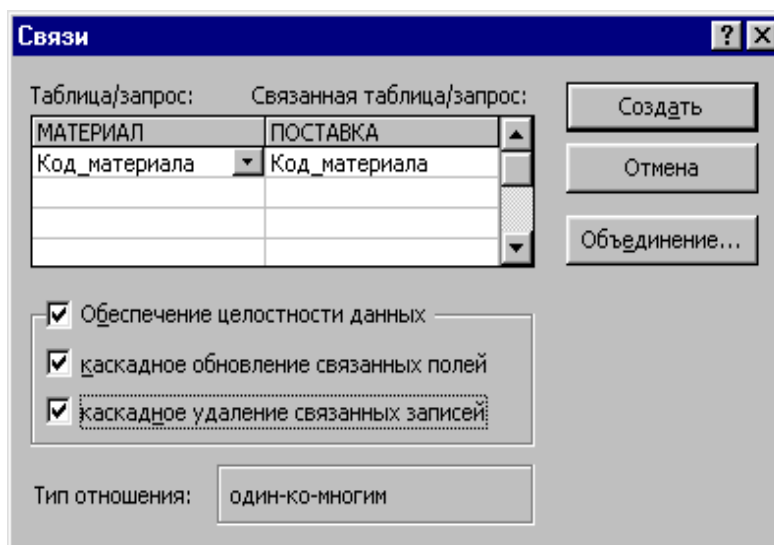
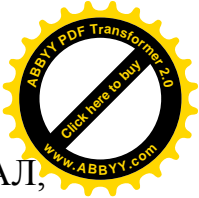


Рис. 9.5 – Вікно встановлення зв'язку між таблицями



Якщо встановити зв'язок між таблицями ПОСТАЧАННЯ, МАТЕРІАЛ, ПОСТАЧАЛЬНИК і МІСТО, то схема даних прийме вид, показаний на рис. 9.1.

Якщо характер зв'язку типу "один-до-одного" або "один-до-багатьох" не встановлений або його варто змінити, то клацнути правою кнопкою миші на лінії зв'язку і вибрати пункт "Изменить связь...". З'явиться вікно "Связь", показане на рис. 9.5. У ньому визначити або змінити зв'язок.

Для видалення зв'язку необхідно клацнути правою кнопкою миші на лінії зв'язку й у контекстному меню вибрати пункт "Удалить".

Для видалення таблиці зі схеми даних варто видалити зв'язок з нею, а потім помітити її клацанням кнопки миші по ній і натиснути клавішу

### 9.3.4 Введення даних у таблицю

Для введення даних у таблицю можна використовувати:

1. вкладку "**Таблицы**", у якій необхідно виділити ім'я потрібної таблиці і клацнути по кнопці , а потім увести дані в таблицю, що відкрилася;
2. вкладку "**Формы**", у якій необхідно виділити ім'я потрібної форми і клацнути по кнопці , а потім увести дані в поля форми, що відкрилася.

## 9.4 Використання форм

### 9.4.1 Створення простої форми за допомогою майстра

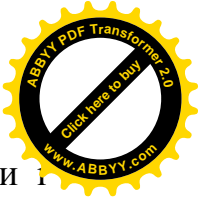
Форми використовуються для введення даних у таблиці, а також для перегляду і редагування таблиць. Щоб створити форму необхідно:

1. На вкладці "Формы" відкритої Базы даних натиснути кнопку , а потім виділити пункт "Мастер форм" і підтвердити .
2. У списку "**Таблицы/зароси**", що розкриваються, запиту вибрати таблицю, для якої варто створювати форму. У нижній частині ліворуч в області "**Доступные поля**" виділяти потрібні поля і кнопкою  переміщати них в область "**Выбранные поля**"

**Зауваження1:** Якщо потрібно помістити у форму поля іншої таблиці, то її варто вибрати в списку "**Таблицы/зароси**", що розкриваються, а потім уже поміщати її поля у форму.

**Зауваження2:** Якщо потрібно помістити у форму всі поля таблиці одночасно, те їх можна помістити кнопкою

Після вибору всіх полів, що поміщаються у форму варто натиснути кнопку .



3. У вікні, що з'явилося, необхідно вибрати зовнішній вигляд форми і натиснути кнопку **Далее**.
4. Далі вибрати зручний стиль оформлення форми і натиснути **Далее**.
5. Ввести ім'я для форми і натиснути кнопку **Готово**.

#### 9.4.2 Створення підпорядкованих форм

Підпорядкована форма створюється для двох або декількох зв'язаних таблиць (*наприклад*, для таблиць **ПОСТАЧАЛЬНИК** і **ПОСТАЧАННЯ**). Для її створення необхідно:

1. На вкладці "Формы" відкритої Бази даних натиснути кнопку **Создать**, а потім вибрати пункт "Мастер форм" і, не вказуючи як джерело даних таблицю, натиснути кнопку **Далее**.

2. У вікні "Создание форм" вибрати спочатку таблицю, що є довідником для інший, тобто таблицю, що зв'язує поле якої має відношення "один" (*для нашого приклада*, таблицю **ПОСТАЧАЛЬНИК**). Потрібні поля цієї таблиці помістити у форму. Потім вибрати таблицю, з яким ця таблиця-довідник зв'язана відношенням "багато" або  $\infty$  (*для нашого приклада*, таблицю **ПОСТАЧАННЯ**). Потрібні поля другої таблиці помістити також у форму і натиснути кнопку **Далее**.

3. У вікні, що з'явилося, вибрати тип представлення даних у формі. Якщо на попередньому кроці першої була обрана таблиця довідник, то у вікні, що з'явилося, натиснути кнопку **Далее**.

4. Потім вибрати зовнішній вигляд підпорядкованої форми (зручніше форма таблична) і натиснути кнопку **Далее**.

5. Потім вибрати потрібний стиль форми і натиснути **Далее**.

6. Вказати ім'я форми і натиснути кнопку **Готово**.

**Зуваження:** Якщо поля таблиць поміщалися в правильній послідовності, то у вікні для вказівки імені форми будуть запропоновані 2 імені для головної і підпорядкованої форм (ім'я підпорядкованої форми закінчується фразою "...подчиненная форма", а ім'я головної форми бажано перейменувати за змістом). Якщо запропоновано тільки одне ім'я для створюваної підпорядкованої форми, то дії були виконані невірно.

*Приклад.* Створити прості форми для таблиць **МІСТО**, **ПОСТАЧАЛЬНИК**, **МАТЕРІАЛ**, **ПОСТАЧАННЯ**. Підпорядковану форму створити для таблиць **ПОСТАЧАННЯ** і **ПОСТАЧАЛЬНИК**, щоб була можливість побачити для кожного постачальника зведення про його постачання і змінювати них. Заповнити таблиці за допомогою створених форм.

Хід виконання:

Для якої з таблиць створювати форму в першу чергу не має значення, тобто створювати форми можна в будь-якій послідовності, а здійснювати введення даних варто починати з таблиць, що беруть участь у зв'язках з боку "один" (1), тобто яких-небудь таблиць є, що довідниками для.

Почнемо зі створення форми для таблиці МІСТО. Для цього необхідно відкрити вікно бази даних. У цьому вікні необхідно відкрити вкладку "Форми" і клацнути по кнопці **Создать**. Потім вибрати пункт "Мастер форм" і підтвердити, натиснувши кнопку **Ok**.

У вікні "Создание Форм", що з'явилося, у списку, що розкривається, вибрати таблицю МІСТО, а потім обидва поля (Код\_міста, Місто) необхідно помістити в *перелік "Выбранные поля"* вікна "Создание форм". Після вибору всіх полів, що поміщаються у форму варто натиснути кнопку **Далее**. З'явиться вікно, якому необхідно вибрати зовнішній вигляд форми (наприклад, "в один стовпець") і натиснути кнопку **Далее**. У вікні, що з'явилося, необхідно вибрати зручний стиль оформлення форми (наприклад, "Хмари") і натиснути кнопку **Далее**. Потім необхідно набрати з клавіатури ім'я таблиці (наприклад, МІСТО) і натиснути кнопку **Готово**.

У результаті буде створена проста форма для таблиці МІСТО, показане на рис. 9.6.

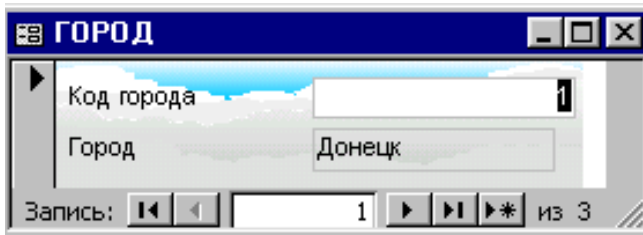


Рис.9.6

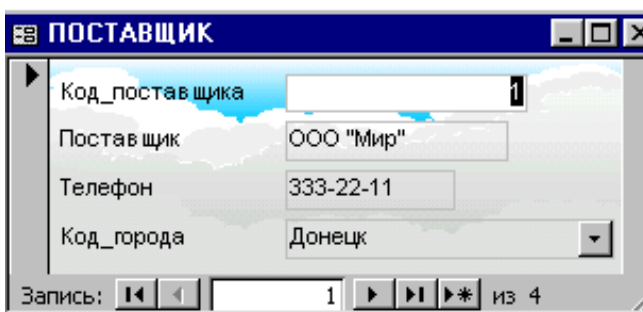


Рис.9.7

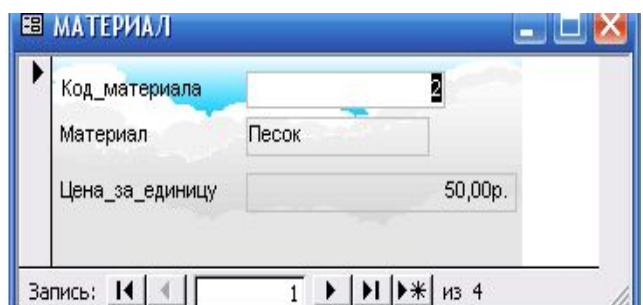


Рис.9.8



Рис.9.9

Аналогічно можна створити прості форми для інших таблиць. Їхній вид показаний на рисунках 9.7 - 9.9.

Перед створенням підпорядкованої форми необхідно визначитися, яка з таблиць є довідником. У даному випадку довідником буде таблиця ПОСТАЧАЛЬНИК, тому що в таблиці ПОСТАЧАННЯ можуть зберігатися інформація про різні поставання того самого постачальника.

При створенні підпорядкованої форми для таблиць ПОСТАЧАННЯ і ПОСТАЧАЛЬНИК, необхідно у вкладці "Форми" вікна бази даних натиснути кнопку **Создать**.

У вікні, що з'явилося, вибрати пункт "Майстер форм" і натиснути кнопку **Далее**.

З'явиться вікно "Создание форм", у якому необхідно вибрати спочатку таблицю, що є довідником для іншої, тобто таблицю ПОСТАЧАЛЬНИК.

Усі поля цієї таблиці помістити у форму можна аналогічно, як і при створенні простої форми. Потім вибрати таблицю, з яким цей довідник зв'язаний відношенням "багато" ( $\infty$ ), тобто таблицю ПОСТАЧАННЯ. Усі поля другої таблиці необхідно також помістити у форму і натиснути кнопку **Далее**.

З'явиться вікно, у якому пропонується вибрати тип представлення даних у формі (наприклад, "в один столбец") і натиснути кнопку **Далее**.

У вікні, що з'явилося, необхідно вибрати зовнішній вигляд підлеглої форми (наприклад, "табличный") і натиснути кнопку **Далее**.

Потім необхідно вибрати стиль форми (наприклад, "Хмари") і натиснути **Далее**.

У вікні, що з'явилося, необхідно вказати ім'я форми (наприклад, ПОСТАЧАЛЬНИК1) і натиснути кнопку **Готово**.

У результаті буде створена підпорядкована форма, показана на рис. 9.10.

**ПОСТАВЩИК1**

Код\_поставщика: 1  
 Поставщик: ООО "Мир"  
 Телефон: 333-22-11  
 Код\_города: Донецк

ПОСТАВКА

Код_поставки	Дата_поставки	Код_материала	Количество_материала	Код_поставщика
3	20.11.01	Песок		6 000 "Мир"
9	20.04.02	Доски		10 000 "Мир"
10	15.05.02	Шифер		150 000 "Мир"
(Счетчик)				0 000 "Мир"

Запись: 4 из 4

Запись: 1 из 4

Рис.9.10

За допомогою цих форм можна заповнити всі таблиці, як показано на рис. 9.11 – 9.14. При активізації для заповнення поля, для якого була установлена властивість "Підстановка", буде запропонований список можливих варіантів, що розкривається, значень цього поля.

**ГОРОД : таблиця**

Код города	Город
1	Донецк
2	Макеевка
3	Луганск
(Счетчик)	

Запись: 1

Рис.9.11

**МАТЕРИАЛ : таблиця**

Код_материала	Материал	Цена_за_единицу
2	Песок	50,00р.
3	Цемент	100,00р.
4	Доски	60,00р.
5	Шифер	10,00р.
(Счетчик)		0,00р.

Запись: 1

Рис.9.12



Код_поставщика	Поставщик	Телефон	Код_города
1	ООО "Мир"	333-22-11	Донецк
2	АО "Дон"	323-15-17	Донецк
3	АО "Лидер"	65-257	Макеевка
4	"Алиса"	55-12-77	Луганск
*	(Счетчик)		

Запись: 1 из 4

Рис.9.13

Код_поставки	Дата_поставки	Код_материала	Количество_материала	Код_поставщика
1	01.10.01	Песок		3 "Алиса"
2	15.10.01	Цемент		5 АО "Дон"
3	20.11.01	Песок		6 ООО "Мир"
4	14.12.01	Песок		5 "Алиса"
5	10.01.02	Доски		10 АО "Лидер"
6	15.02.02	Шифер		100 АО "Лидер"
7	15.02.02	Цемент		10 АО "Дон"
8	25.03.02	Доски		15 АО "Лидер"
9	20.04.02	Доски		10 ООО "Мир"
10	15.05.02	Шифер		150 ООО "Мир"
*	(Счетчик)			0

Запись: 1 из 10

Рис.9.14

## 9.5 Вибірка даних за запитом

### 9.5.1 Запити на вибірку

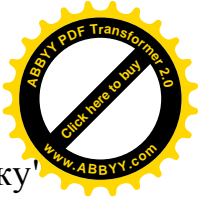
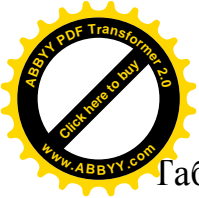
Запити на вибірку використовуються для добору потрібної користувачеві інформації, що утримується в таблицях. Вони створюються тільки для зв'язаних таблиць.

Для створення запиту на вибірку необхідно:

1. На вкладці "Запросы" відкритої Базы даних натиснути кнопку Создать .

2. У вікні, що з'явилося, вибрати пункт "Конструктор" і натиснути ОК . При цьому з'являться два накладені один на одне вікна: "Добавление таблицы" і "Запрос на выборку". Вікно "Добавление таблицы" активно.

3. У вікні "Добавление таблицы" зі списку необхідно виділити таблиці, поля яких будуть використовуватися в запиті, і натиснути кнопку Добавить .



Таблиці, що додаються, з'являться у верхній частині вікна "Запрос на выборку" із установленими раніше зв'язками.

4. Закрити вікно "Добавление таблицы".

5. У нижній частині конструктора вікна "Запрос на выборку" кожен стовпець – це одне поле, яке потрібно вибрати з таблиці, поміщеної у верхню частину вікна, або сконструювати у виді поля, що обчислюється.

а) Щоб у запит помістити поле з таблиці можна : двічі клацнути лівою кнопкою миші у верхній частині вікна "Запрос на выборку" по потрібному полю таблиці. При цьому його ім'я з'явиться автоматично у вільному стовпці нижньої частини вікна в рядку "Поле".

б) Для створення поля, що обчислюється, у запиті необхідно в нижній частині вікна "Запрос на выборку" установити курсор у рядку "Поле" вільного стовпця і клацнути правою кнопкою миші, а потім вибрати команду "Построить...", а потім у верхній частині вікна "Построитель выражений" у текстовому полі набрати з клавіатури ім'я створюваного поля, що обчислюється, (без пробілів, крапок і ін.) і за ним поставити двокрапку. Після двокрапки побудувати вираз для обчислення значень цього поля і натиснути кнопку .

**Зауваження1.** Для побудови виразу в "Построитель выражений" бажано скористатися нижньою частиною вікна "Построитель выражений", розбитої на 3 поруч розташовані частини.

При цьому в *лівій* частині можна подвійним щигликом вибирати папки ТАБЛИЦІ, ФУНКЦІЇ і т.д.

І якщо обрано папку ТАБЛИЦІ і виділене ім'я якоїсь таблиці, то в *середній* частині відобразяться імена полів виділеної таблиці, кожне з яких можна помістити споруджуване вираження двома способами: 1) подвійне клацання лівою кнопкою миші по імені поля в середній частині або 2) виділити ім'я поля в середній частині і натиснути кнопку  .

А якщо обрано папку ФУНКЦІЇ, а потім об'єкт ВБУДОВАНІ ФУНКЦІЇ, то в *середній* частині будуть відображатися категорії функцій, а в *правій* – імена функцій виділеної в середній частині категорії. У вираження містяться аналогічно.

**Зауваження2.** При вставці поля таблиці або стандартної функції в споруджуваному виразі може з'явитися об'єкт <Выражение>, який необхідно замінити або видалити.

Приклад. З бази даних вивести наступні зведення: найменування матеріалу, найменування постачальника, місцезнаходження постачальника, дату постачання, кількість поставленого матеріалу.

Хід виконання:

Вивести необхідні зведення з бази даних можна за запитом на вибірку.

Для створення простого запиту на вибірку необхідно у верхню частину запиту помістити всі таблиці, тому що знадобляться поля з усіх таблиць. Для приміщення полів у нижню частину запиту необхідно вибрати потрібні поля з відповідних таблиць. Так наприклад, після подвійного клацання по полю Матеріал з таблиці МАТЕРІАЛ це поле з'явиться в нижній частині вікна. Аналогічно можна помістити в нижню частину вікна запиту інші потрібні поля. У результаті буде сформований запит, що у режимі конструктора має вигляд, показаний на рис. 9.15.

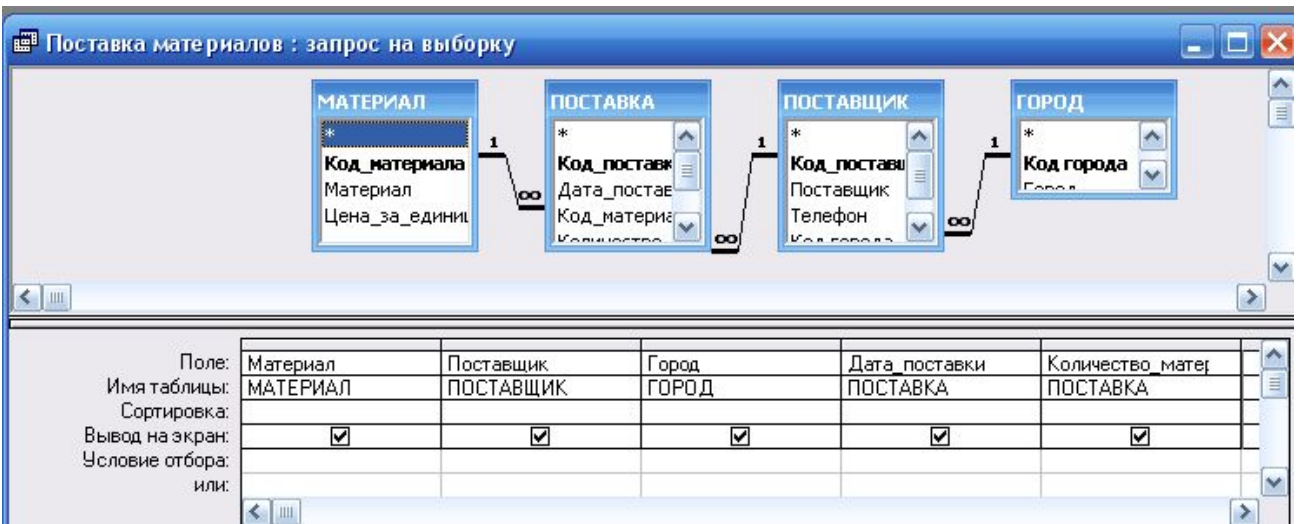


Рис. 9.15

Якщо виконати цей запит, клацнувши мишею по його назві двічі, то можна побачити набір записів цього запиту, показаний на рис. 9.16.

	Материал	Поставщик	Город	Дата поставки	Количество
▶	Песок	"Алиса"	Луганск	01.10.01	3
	Цемент	АО "Дон"	Донецк	15.10.01	5
	Песок	ООО "Мир"	Донецк	20.11.01	6
	Песок	"Алиса"	Луганск	14.12.01	5
	Доски	АО "Лидер"	Макеевка	10.01.02	10
	Шифер	АО "Лидер"	Макеевка	15.02.02	100
	Цемент	АО "Дон"	Донецк	15.02.02	10
	Доски	АО "Лидер"	Макеевка	25.03.02	15
	Доски	ООО "Мир"	Донецк	20.04.02	10
	Шифер	ООО "Мир"	Донецк	15.05.02	150
*					

Запись: 1 из 10

Рис.9.16



## 9.5.2 Створення запиту з полем, що обчислюється

В ACCESS можна виконувати обчислення з будь-якими полями зв'язаних таблиць і створити нове поле, що обчислюється. При цьому можна використовувати будь-які вбудовані функції ACCESS. Крім того поля запиту можуть включати дані, які отримані за допомогою арифметичних дій над полями таблиці.

Приклад. З бази даних вивести наступні зведення: дату постачання, найменування постачальника, найменування матеріалу, кількість поставленого матеріалу, ціну матеріалу за одиницю, вартість поставленого матеріалу.

### Хід виконання:

Для цього необхідно створити простий запит на вибірку з полем, що обчислюється. Назвемо це поле "Вартість".

У верхню частину запиту необхідно помістити таблиці ПОСТАЧАННЯ, МАТЕРІАЛ, ПОСТАЧАЛЬНИК, тому що знадобляться поля тільки з цих таблиць.

У нижній частині запиту необхідно вибрати потрібні поля з відповідних таблиць.

Для побудови поля, що обчислюється, "Вартість" необхідно викликати "Построитель выражений". Для цього необхідно в нижній частині вікна запиту установити курсор у рядку "Поле" вільного стовпця і, клацнувши правою кнопкою миші, вибрати команду "Построить..."

У вікні "Построитель выражений", що з'явилося, необхідно написати ім'я поля, що обчислюється, тобто "Вартість:", а потім перемножити вартість матеріалу на його кількість.

Для цього в лівій нижній частині вікна "Построитель выражений" необхідно двічі клацнути по значку "Таблицы".

При цьому нижче значка "Таблицы" з'являться папки всіх таблиць бази даних.

Якщо клацнути мишею по значку однієї з таблиць (наприклад, МАТЕРІАЛ), то в середній нижній частині вікна "Построитель выражений" з'явиться перелік полів цієї таблиці.

Далі необхідно виділити потрібне поле (наприклад, *Ціна\_за\_одиницю*) і клацнути по кнопці **Вставити**.

При цьому обране поле з'явиться у верхній частині вікна "Построитель выражений" (із вказівкою імені таблиці, з якого узяті це поле).

Потім необхідно ввести знак множення (\*) і аналогічно помістити у верхню частину вікна "Построитель выражений" поле *Кількість\_матеріалів* з таблиці ПОСТАЧАННЯ.

Вікно "Построитель выражений" прийме вид, показаний на рис. 9.17. У цьому вікні необхідно клацнути по кнопці **ОК**.



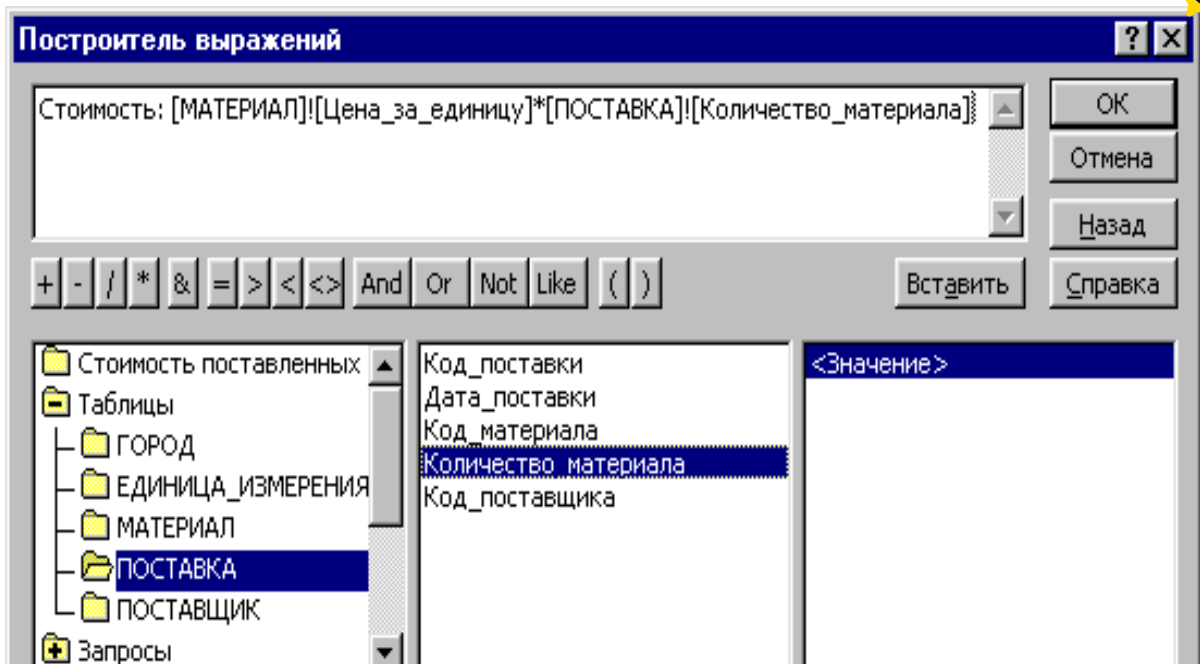


Рис.9.17

У результаті буде сформований запит, що у режимі конструктора має вигляд, показаний на рис. 9.18.

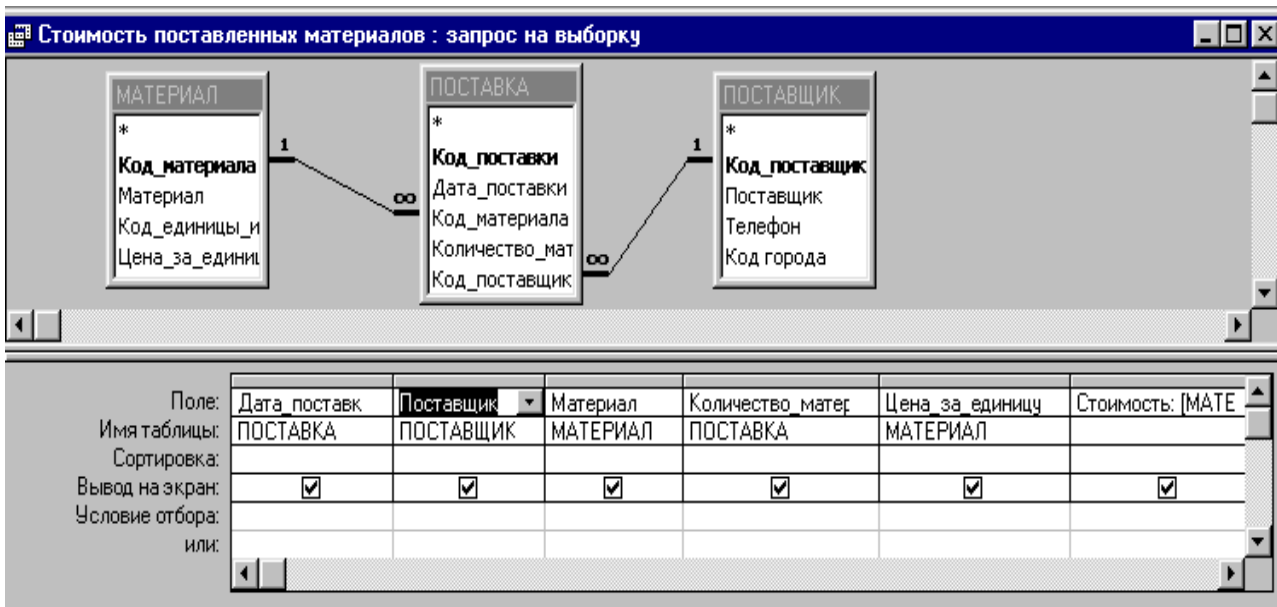
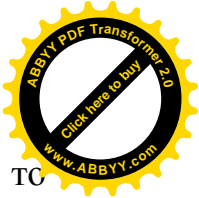


Рис.9.18



Якщо виконати цей запит, клацнувши мишею по його назві двічі, то можна побачити набір записів цього запиту, показаний на рис. 9.19.

Стоимость поставленных материалов : запрос на выборку						
	Дата_поста	Поставщик	Материал	Количество	Цена_за_ед	Стоимость
▶	01.10.01	"Алиса"	Песок	3	50,00р.	150,00р.
	15.10.01	АО "Дон"	Цемент	5	100,00р.	500,00р.
	20.11.01	ООО "Мир"	Песок	6	50,00р.	300,00р.
	14.12.01	"Алиса"	Песок	5	50,00р.	250,00р.
	10.01.02	АО "Лидер"	Доски	10	60,00р.	600,00р.
	15.02.02	АО "Лидер"	Шифер	100	10,00р.	1 000,00р.
	15.02.02	АО "Дон"	Цемент	10	100,00р.	1 000,00р.
	25.03.02	АО "Лидер"	Доски	15	60,00р.	900,00р.
	20.04.02	ООО "Мир"	Доски	10	60,00р.	600,00р.
	15.05.02	ООО "Мир"	Шифер	150	10,00р.	1 500,00р.
*						

Запись: 1 из 10

Рис.9.19

### 9.5.3 Сортировка данных у запросе

Зазвичай ACCESS виводить записи в тім порядку, у якому вони вибираються з бази даних. Щоб упорядкувати дані в запиті за зростанням або за зменшенням якого-небудь поля необхідно відкрити запит у режимі конструктора, установити курсор у рядку "Сортировка" відповідного поля запиту і вибрати зі списку, що розкривається, порядок сортування.

Якщо встановлено сортування в декількох полях, то спочатку дані сортуються по лівому полю, а потім по наступному за ним полю із сортуванням, тобто порядок сортування зліва направо.

Приклад. З бази даних вивести наступні зведення: дату постачання, найменування постачальника, найменування матеріалу, кількість поставленого матеріалу, причому упорядкувати записи за найменуванням постачальника, а потім для кожного постачальника упорядкувати записи за найменуванням матеріалів.

#### Хід виконання:

Необхідно створити простий запит на вибірку зі складним сортуванням. У верхню частину запиту необхідно помістити таблиці ПОСТАЧАННЯ, МАТЕРІАЛ, ПОСТАЧАЛЬНИК, тому що знадобляться поля тільки з цих таблиць. У нижній частині запиту необхідно вибрати потрібні поля з відповідних таблиць. Причому поле *Постачальник* повинне розташовуватися зліва від поля *Матеріал*. Для полів *Постачальник* і *Матеріал* у рядку "Сортировка" необхідно вибрати в списку, що розкривається, пункт "по зростанню". У результаті буде сформований запит, що у режимі конструктора має вигляд, показаний на рис. 9.20.

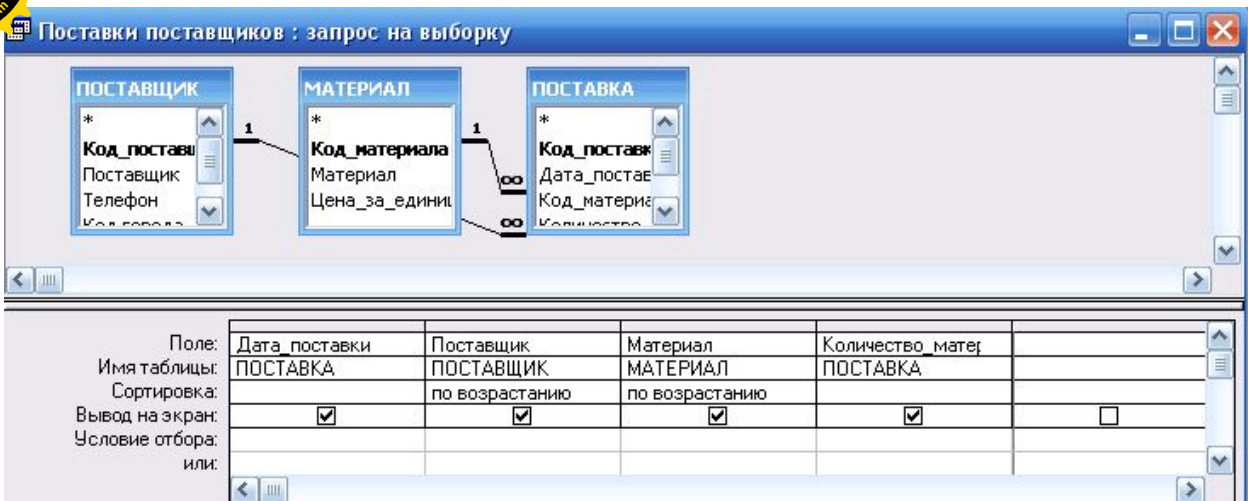


Рис. 9.20

Якщо виконати цей запит, клацнувши мишею по його назві двічі, то можна побачити набір записів цього запиту, показаний на рис. 9.21.

Дата_поставки	Поставщик	Материал	Количество_материала
14.12.01	"Алиса"	Песок	5
01.10.01	"Алиса"	Песок	3
15.02.02	АО "Дон"	Цемент	10
15.10.01	АО "Дон"	Цемент	5
25.03.02	АО "Лидер"	Доски	15
10.01.02	АО "Лидер"	Доски	10
15.02.02	АО "Лидер"	Шифер	100
20.04.02	ООО "Мир"	Доски	10
20.11.01	ООО "Мир"	Песок	6
15.05.02	ООО "Мир"	Шифер	150

Запись: 11 из 11

Рис.9.21

### 9.5.4 Вибірка даних за умовою

Для вибірки даних за умовою спочатку необхідно створити простий запит, що містить виведені поля, а потім відкрити цей запит у режимі конструктора і накласти умови відбору.

Для накладення умов відбору використовуються рядки в нижній частині вікна запиту, починаючи з рядка "Условия отбора" і нижче.

Якщо відбору записані в одному рядку, то вони сприймаються як умови, з'єднані логічною операцією "И", а якщо вони розташовані в різних рядках, те як умови, з'єднані логічною операцією "ИЛИ".

Для побудови умов відбору можна використовувати "Построитель выражений", за допомогою якого можна вводити навіть знаки операцій. Якщо потрібно використовувати операцію "=" (дорівнює), то цей знак можна не

писати.

Значення дати необхідно поміщати між знаками # (*наприклад*, #09.01.00#).

Текстові значення необхідно укласти в подвійних лапках (*наприклад*, "Київ").

В умовах добору кожного поля можна використовувати:

- Знаки відносин: >, <, >=, <=, =, <>
- Знаки логічних операцій: OR, AND, NOT
- Оператори BETWEEN, IN і LIKE:

*Приклад.* З бази даних вивести наступні зведення: дату постачання і кількість поставленого матеріалу для постачальників, найменування яких починається на "ТОВ" і таких, що доставили пісок.

Хід виконання:

Для рішення задачі необхідно створити простий запит на вибірку з умовами відбору. У верхню частину запиту необхідно помістити таблиці ПОСТАЧАННЯ, МАТЕРІАЛ, ПОСТАЧАЛЬНИК, тому що знадобляться поля тільки з цих таблиць. У нижній частині запиту необхідно вибрати потрібні поля з відповідних таблиць. Для полів *Постачальник* і *Матеріал* у рядку "Условия отбора" необхідно ввести наступні обмеження: LIKE "ООО\*" (для поля *Постачальник*) і "песок" (для поля *Матеріал*). У результаті буде сформований запит, що у режимі конструктора має вигляд, показаний на рис. 9.22.

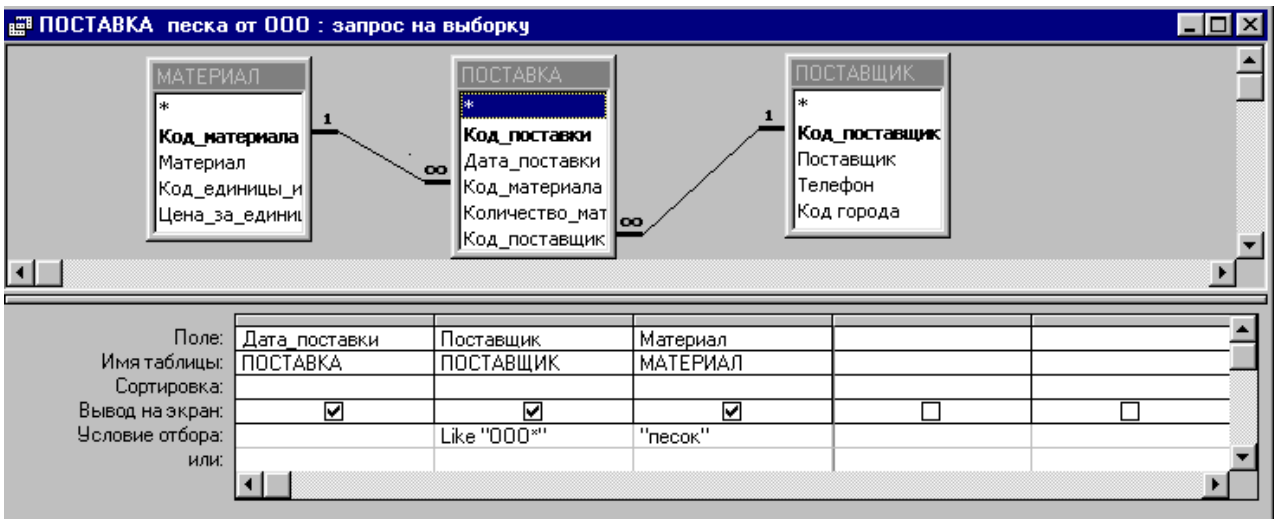


Рис.9.22

Якщо виконати цей запит, клацнувши мишею по його назві двічі, то можна побачити набір записів цього запиту, показаний на рис. 9.23.



Дата_поставки	Поставщик	Материал
20.11.01	ООО "Мир"	Песок

Запись: 2 из 2

Рис.9.23

### 9.5.5 Параметричний запит

Параметричний запит – це запит, при виконанні якого задається змінний параметр.

Для створення параметричного запиту необхідно спочатку створити простий запит для вибору потрібних полів (у тому числі і полів, для яких будуть вводитися параметри).

Щоб визначити параметр, необхідно в поле, для якого задається змінне значення в рядку "Условия отбора" замість конкретного значення ввести фразу, взяту в квадратні дужки.

Те, що взяте в квадратні дужки, ACCESS розглядає як ім'я параметра. Воно виводиться у вікні діалогу при виконанні запиту. Тому як ім'я параметра розумно використовувати змістовну фразу.

В одному запиті можна задати кілька параметрів. При цьому ім'я кожного параметра повинне бути унікальним і змістовним.

При виконанні запиту ACCESS попросить увести по черзі значення для кожного з параметрів, використовуючи вікна діалогів.

*Приклад.* З бази даних вивести зведення про постачання постачальниками конкретного матеріалу (вводиться за запитом) у заданий період (зазначений за запитом).

#### Хід виконання:

Щоб вивести ці зведення необхідно створити параметричний запит із трьома параметрами. У верхню частину запиту необхідно помістити таблиці ПОСТАЧАННЯ, МАТЕРІАЛ, ПОСТАЧАЛЬНИК, тому що знадобляться поля тільки з цих таблиць. У нижній частині запиту необхідно вибрати потрібні поля з відповідних таблиць. Для поля *Material* у рядку "Условия отбора" необхідно ввести параметр [Введіть матеріал]. Для вказівки діапазону дат потрібного періоду в поле *Дата\_постачання* в рядку "Условия отбора" необхідно ввести BETWEEN [Введіть початкову дату] AND [Введіть кінцеву дату].

У результаті буде сформований запит, що у режимі конструктора має вигляд, показаний на рис. 9.24.

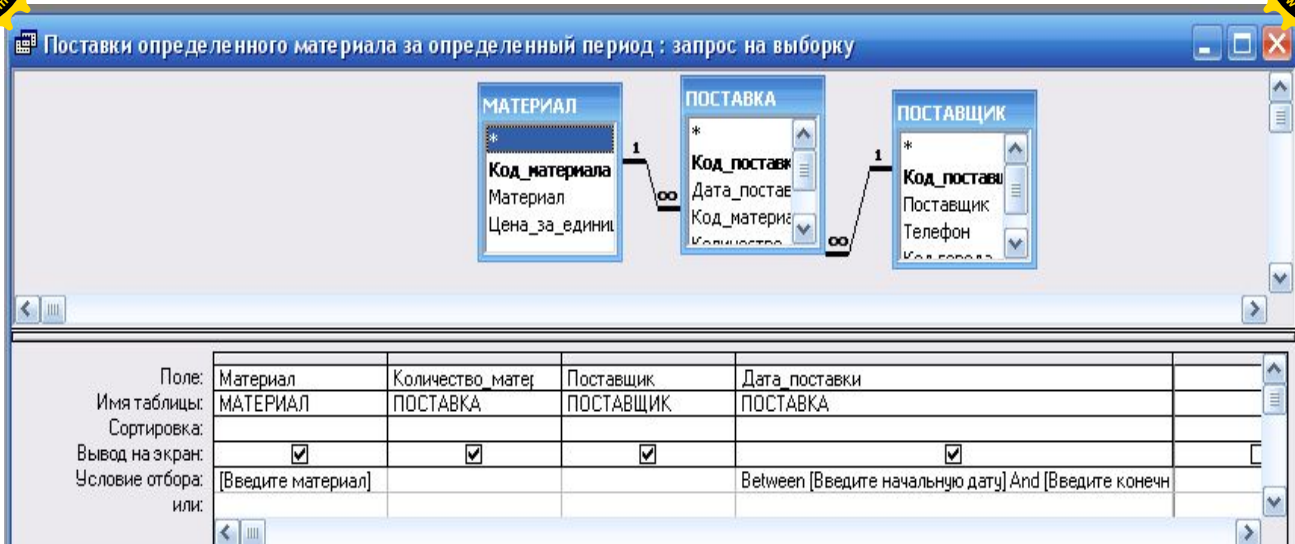


Рис.9.24

Якщо виконати цей запит, клацнувши мишею по його назві двічі, то спочатку з'явиться діалогове вікно для введення значення параметра [Введите материал], показане на рис. 9.25, а після введення значення "песок" і клацання по кнопці **OK** цього діалогового вікна по черзі з'являться діалогові вікна для введення значень параметрів [Введите начальную дату] і [Введите конечную дату]. Після введення в них значень "1.07.01" і "31.12.01" відповідно, як

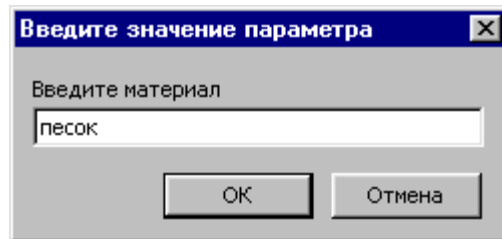


Рис.9.25

показано на рис. 9.26 – 9.27, можна буде побачити набір записів цього запиту, показаний на рис. 9.28.

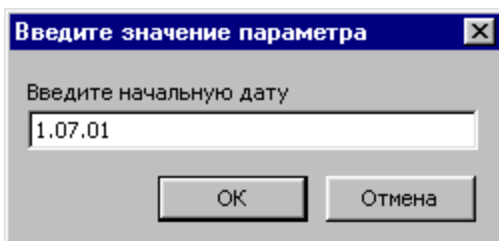


Рис.9.26

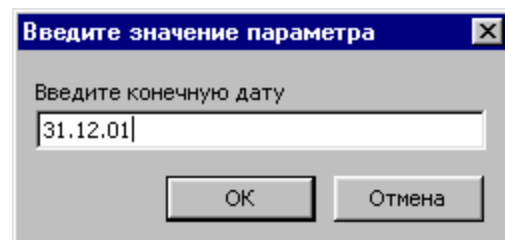
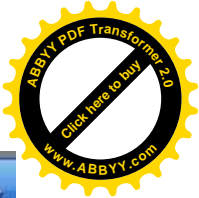
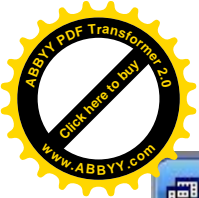


Рис.9.27





Поставки определенного материала за определенный период : запрос на выборку

Материал	Количество_матер	Поставщик	Дата_поставки
Песок	3	"Алиса"	01.10.2001
Песок	6	ООО "Мир"	20.11.2001
Песок	5	"Алиса"	14.12.2001

Запись: 4 из 4

Рис.9.28

### 9.5.6 Підсумковий запит

Іноді потрібні не окремі записи таблиці, а підсумкові значення по групах даних.

Підсумковий запит дозволяє одержати значення таких підсумкових показників, як сумарного, середнього, мінімального, максимальне значення й інш. Для розрахунку цих показників використовуються наступні групові операції: Sum (сума), Count (кількість), Avg (середнє) і інш.

Для отримання підсумкових показників, дані необхідно згрупувати, тобто відсортувати їх за даними того стовпця, при зміні значень якого, підводяться підсумки. Якщо групування здійснюється за даними декількох стовпців, то дані спочатку сортуються за даними самого лівого стовпця з групуванням, а потім за даними наступного стовпця з групуванням. Тому в запиті потрібно розташовувати лівіше те поле з групуванням, за яким повинно здійснюватися зовнішнє сортування.

Для створення підсумкового запиту необхідно виконати наступні дії:

1. Створити простий запит на вибірку, що дозволяє вивести наступні поля:

- поля, за даними яких потрібно групувати дані, тобто при зміні значення яких необхідно підводити підсумки;
- поля, у яких потрібно одержати підсумкові показники.

*Примітка.* Якщо для якого-небудь поля потрібно отримати кілька підсумкових показників (наприклад, максимальне, мінімальне, сумарне значення й інш. показники), то це поле потрібно помістити в запит стільки разів, скільки показників для нього потрібно отримати.

2. У режимі конструктора запиту виконати команду "Групповые операции", натиснувши на панелі інструментів кнопку  $\Sigma$  або клацнувши правою кнопкою миші на будь-якому полі й у контекстному меню вибравши пункт "Групповые операции".

3. У нижній частині запиту в режимі конструктора з'явиться додатковий рядок "Групповая операция", у кожному стовпці якої з'явиться операція

"Группировка". Операція "Группировка" дозволяє групувати дані у тій стовпці, в якому вона знаходиться. Якщо в якому-небудь полі необхідно отримати підсумковий показник, то необхідно змінити для цього поля тип групової операції. Для цього установити курсор у рядку "Групповая операция" цього стовпця і за допомогою списку, що розкривається, вибрати потрібну групову операцію.

Приклад. З бази даних вивести наступні дані: загальну, мінімальну і максимальну кількість кожного виду матеріалу, поставленого кожним постачальником, указавши кількість однотипних поставчань.

Хід виконання:

Для виведення потрібних зведень необхідно створити підсумковий запит із двома рівнями групування і чотирма підсумковими показниками. У верхню частину запиту необхідно помістити таблиці ПОСТАЧАННЯ, МАТЕРІАЛ, ПОСТАЧАЛЬНИК, тому що знадобляться поля тільки з цих таблиць. У нижній частині запиту необхідно вибрати поля *Постачальник*, *Матеріал*, *Кількість\_матеріалу* і *Код\_постачання* з відповідних таблиць. Причому поле *Постачальник* варто розташувати лівіше ніж поле *Матеріал*, тому що зовнішнє сортування повинне здійснюватися за даними поля *Постачальник*, а поле *Кількість\_матеріалу* варто помістити в запит три рази, тому що для цього поля потрібно отримати три підсумкових показники. Потім виконати команду "Групповые операции", натиснувши на панелі інструментів кнопку  $\Sigma$ . У нижній частині запиту в режимі конструктора з'явиться додатковий рядок "Групповая операция", у кожній стовпці якої з'явиться операція "Группировка". Потім у першому стовпці *Кількість\_матеріалу* варто змінити тип групової операції на Sum, у другому – на Min і в третьому - на Max, а в стовпці *Код\_постачання* – на Count. У результаті буде сформований запит, що у режимі конструктора має вигляд, показаний на рис. 9.29.

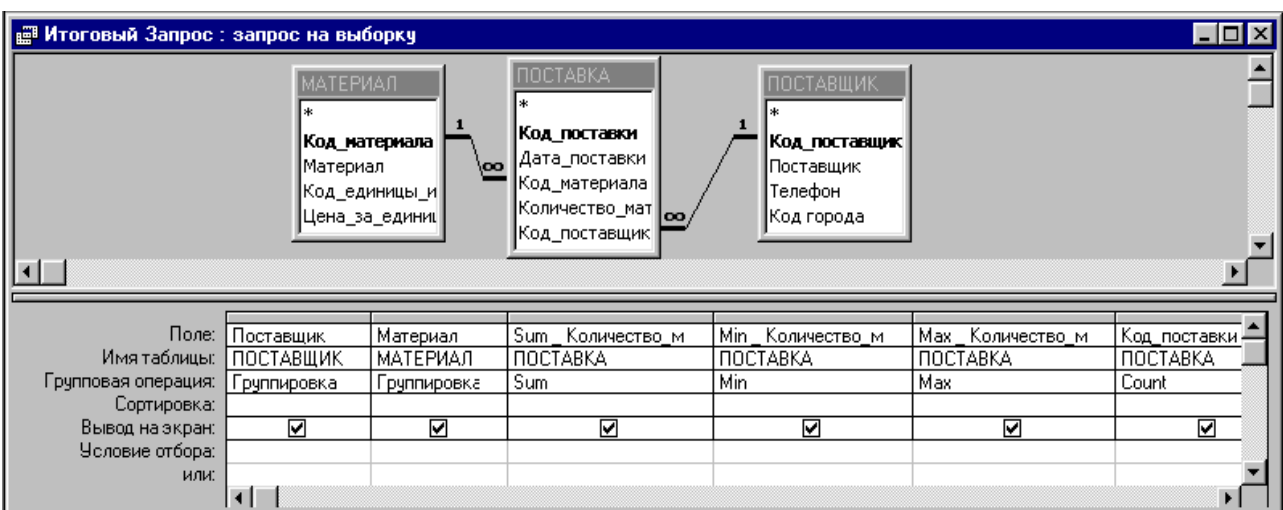
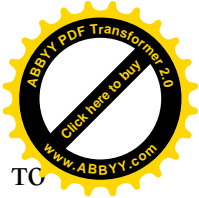


Рис.9.29



Якщо виконати цей запит, клацнувши мишею по його назві двічі, то можна побачити набір записів цього запиту, показаний на рис. 9.30.

Поставщик	Материал	Sum_Количество	Min_Количество	Max_Количество	Count_Код_поставки
"Алиса"	Песок	8	3	5	2
АО "Дон"	Цемент	15	5	10	2
АО "Лидер"	Доски	25	10	15	2
АО "Лидер"	Шифер	100	100	100	1
ООО "Мир"	Доски	10	10	10	1
ООО "Мир"	Песок	6	6	6	1
ООО "Мир"	Шифер	150	150	150	1

Рис.9.30

## 9.6 Створення простих звітів

В основному звіти застосовуються для перегляду "картини в цілому", тому за основу часто використовуються багатотабличні запити. Тому для створення звіту попередньо необхідно створити базовий запит, у який помістити всю потрібну нам інформацію. Потім можна приступити до створення звіту. При створенні звіту можна скористатися допомогою Майстра по розробці звітів. Для цього необхідно виконати наступні дії:

1. У вікні відкритої бази даних вибрати вкладку "Отчеты" і клацнути по кнопці **Создать**. З'явиться вікно "Новый отчет".
2. У вікні "Новый отчет" виділити пункт "Мастер отчетов" і в списку таблиць і запитів, що розкривається, вибрати попередньо створений базовий запит, а потім клацнути по кнопці **ОК**.
3. З'явиться вікно, у якому впливає всі потрібні доступні поля базового запиту перемістити з області "Доступные поля" в область "Выбранные поля" аналогічно тому, як це здійснюється для створення форм, а потім клацнути по кнопці **Далее**.
4. З'явиться вікно, як показано на рис. 9.31. У вікні, що з'явилося, вибрати потрібний тип представлення даних, а потім клацнути по кнопці **Далее**.
5. З'явиться вікно, у якому можна додати рівні групування, а потім клацнути по кнопці **Далее**.

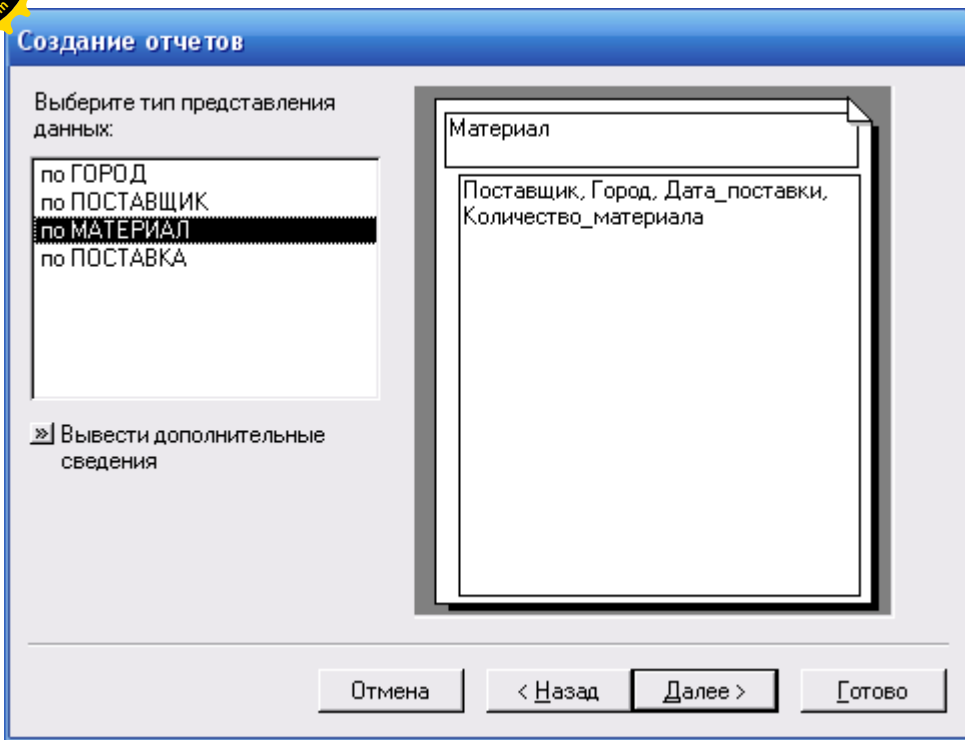


Рис. 9.31

6. З'явиться вікно, у якому можна вибрати порядок сортування даних і вказати підсумкові операції після клацання по кнопці **Итоги...**, а потім клацнути по кнопці **Далее**.

7. З'явиться вікно, у якому необхідно вибрати вид макета для звіту, а потім клацнути по кнопці **Далее**.

8. З'явиться вікно, у якому необхідно вибрати стиль для звіту, а потім клацнути по кнопці **Далее**.

9. З'явиться вікно, у якому необхідно задати ім'я звіту, а потім клацнути по кнопці **Готово**.

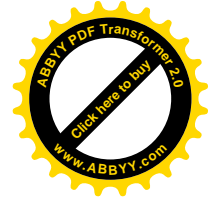
*Приклад.* Якщо створити звіт на базі вже створеного запиту "Постачання матеріалів" (див. вище рис. 9.15 – 9.16), виконавши послідовно кроки майстра по створенню звітів, і назвавши звіт "Звіт про постачання матеріалів", можна одержати звіт, що у режимі перегляду виглядає як на рис. 9.32.



## ОТЧЕТ О ПОСТАВКАХ МАТЕРИАЛОВ

<i>Материал</i>	<i>Поставщик</i>	<i>Город</i>	<i>Дата_поставки</i>	<i>количество_материала</i>
<i>Доски</i>				
	ООО "Мир"	Донец	20.04.2002	10
	АО "Лидер"	Мариуполь	25.03.2002	15
	АО "Лидер"	Мариуполь	10.01.2002	10
<i>Итого для Материал = Доски (3 записей)</i>				
<b>Sum</b>				35
<i>Песок</i>				
	"Алиса"	Луганск	14.12.2001	5
	ООО "Мир"	Донец	20.11.2001	6
	"Алиса"	Луганск	01.10.2001	3
<i>Итого для Материал = Песок (3 записей)</i>				
<b>Sum</b>				14
<i>Цемент</i>				
	АО "Дол"	Донец	15.02.2002	10
	АО "Дол"	Донец	15.10.2001	5
<i>Итого для Материал = Цемент (2 записей)</i>				
<b>Sum</b>				15
<i>Шифер</i>				
	ООО "Мир"	Донец	15.05.2002	150
	АО "Лидер"	Мариуполь	15.02.2002	100
<i>Итого для Материал = Шифер (2 записей)</i>				
<b>Sum</b>				250
<b>ИТОГО</b>				314

Рис. 9.32



## 9.7 Вбудовані функції

В таблиці 9.6 представлені функції категорії "Дата/время"

Таблиця 9.6 – Функції категорії "Дата/время"

Функція	Обчислює
<b>Date()</b>	поточну дату у форматі дд.мм.гг,
<b>Now()</b>	поточну дату і час форматі дд.мм.гг чч:мм:сс.
<b>Year(Дата)</b>	рік з Дати(чотири цифри, наприклад, 2003)
<b>Month(Дата)</b>	місяць з Дати у вигляді цілого числа (1 - 12)
<b>Day(Дата)</b>	день з Дати у вигляді цілого числа (1-31)
<b>Hour(Дата)</b>	година з Дати у вигляді цілого числа (1-23)
<b>Minute(Дата)</b>	хвилини з Дати у вигляді цілого числа 1-59)
<b>Secunde(Дата)</b>	секунди з Дати у вигляді цілого числа (1-59)
<b>DatePart("вр.інтервал"; Дата)</b>	компоненту дати відповідну вказаному тимчасовому інтервалу
<b>DateAdd("вр.інтервал"; n; Дата)</b>	нову дату, віддалену на <b>n</b> тимчасових інтервалів від вказаної
<b>DateDiff("вр.інтервал"; Дата1; Дата2)</b>	число тимчасових інтервалів між двома датами
<b>DateSerial (Рік; Місяць; День)</b>	Перетворить три цілі числа <b>рік, місяць і номер дня в місяці</b> , в дане типу Дата/Час .
<b>Weekday(Дата)</b>	порядковий номер дня тижня (1 - неділя)
<b>Weekday(Дата;2)</b>	порядковий номер дня тижня (1 - понеділок)





Значення використовуваного у функціях **DatePart**, **DateAdd**, **DateDiff** параметра *часовий інтервал* ("вр.інтервал"):

- уууу Рік.
- Q Квартал.
- m Місяць.
- Y День року.
- D День місяця.
- w День тижня.
- ww Тиждень.
- H Годинник.
- N Хвилини.
- S Секунди.

В таблиці 9.7 представлені логічні функції.

Таблиця 9.7 – Логічні функції

Функція	Обчислює
<b>Iif</b> (Умова; Дія 1; Дія 2)	виконує <b>Дію 1</b> , якщо виконується <b>Умови1</b> , інакше виконується <b>Дія 2</b> .
<b>Choose</b> (Поле; Значення 1; Значення 2; .... Значення n)	виконує ту дію, номер якого співпадає із значенням, що зберігається в Полі.
<b>Switch</b> (Умови1; Действие1; Умова 2; Дія 2; ... Умова n; Дія n)	послідовно перевіряє <b>Умови</b> і виконує ту <b>Дію</b> , для якої виконується відповідна умова.

## 10 ОСНОВИ ОФІСНОГО ПРОГРАМУВАННЯ

### 10.1 Інтегроване середовище розробки додатків на мові VBA

Для виклику інтегрованого середовища розробки додатків (IDE) необхідно, вибрати в меню наступні команди: "Сервіс" → "Макрос" → "Редактор Visual Basic". Загальний вигляд IDE приведений на рис. 10.1. IDE складається з декількох компонентів: головного меню, панелі інструментів, вікна проекту, вікна властивостей, панелі елементів, конструктора форм, вікна контрольних значень і декілька інших допоміжних вікон.

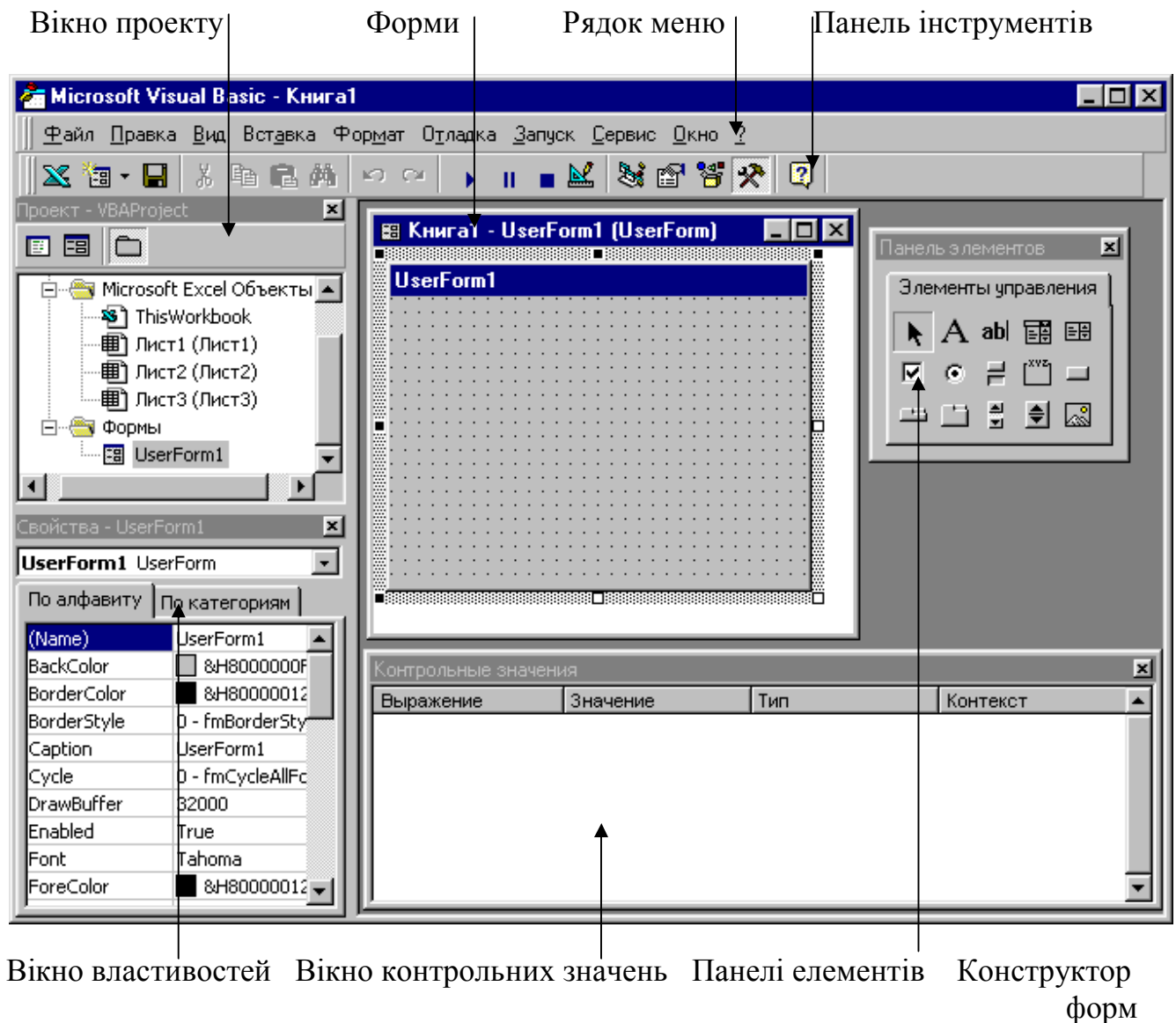


Рис. 10.1 - Загальний вигляд IDE VBA

**Головне меню** – це рядок тексту, розташований у верхній частині вікна Visual Basic, і складається з декількох пунктів.

Меню "Файл" призначено для роботи з файлами, з яких утворюються додатки. У нім можна створювати, зберігати і друкувати проекти. Меню

"**Правка**" виконує стандартні операції з буфером обміну – вирізування, копіювання і вставка. Вони застосовуються не лише до фрагментів програми, але і до елементів, що управляють. У меню "**Вид**" включаються режими перегляду різних компонентів і інструментів. Переглядати можна форми і програмні модулі. Меню "**Вставка**" дозволяє додавати процедури, форми, модулі і модулі класу. Команди меню "**Формат**" визначають розташування і розміри елементів і форм. За допомогою команд меню "**Отладка**" можна запустити і зупинити додаток, розставити точки переривання і вибрати об'єкти, що переглядаються, а також виконати інші операції, що допомагають стежити за роботою додатка. Команди меню "**Запуск**" запускають і зупиняють застосування, переривають і відновлюють виконання програми, що особливо зручно в процесі відладки. Меню "**Сервіс**" дозволяє включити додаткові елементи, запустити макроси і настроїти параметри редактора. Меню "**Окно**" дозволяє збудувати вікна IDE (каскадне або мозаїчне розташування), упорядкувати значки згорнутих форм, а також створює список, що дозволяє швидко перейти до одного з відкритих вікон IDE. Меню "?" – допомога користувачеві. Для швидкого виклику головного меню необхідно натискувати клавішу F10.

**Панель інструментів** знаходиться під головним меню (рис. 10.2). Якщо вона відсутня, необхідно виконати команду "**Вид**" → "**Панели инструментов**" → "**Стандарт**".

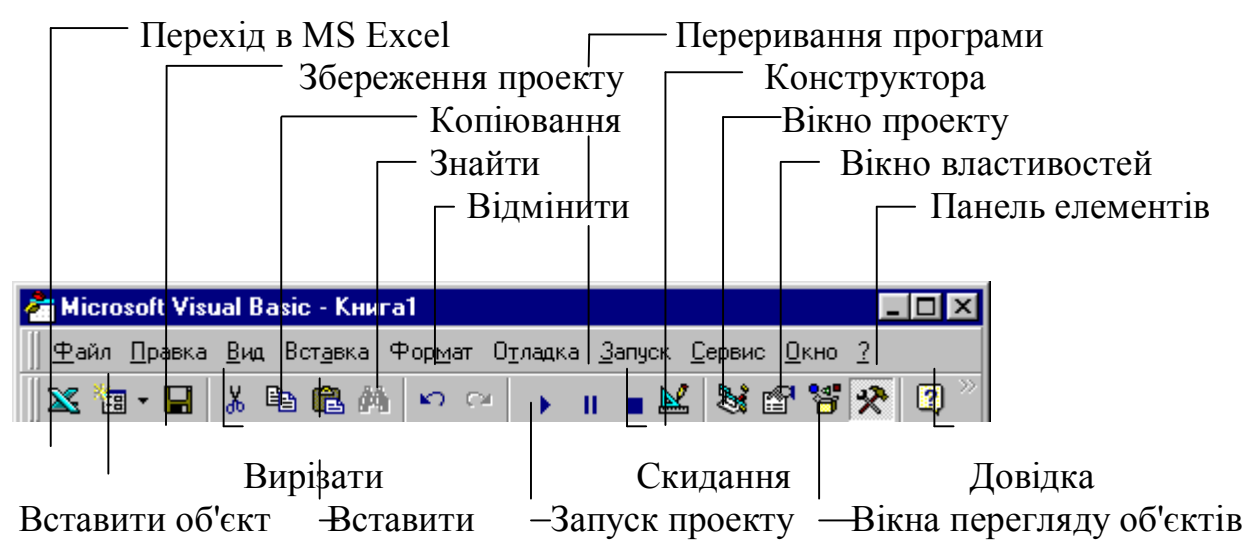


Рис. 10.2 - Панель інструментів VBA

**Вікно проекту** нагадує собою вікно Провідника Windows, і призначено для швидкого перегляду складових проекту, який об'єднує в собі всі об'єкти, складові додатка (рис. 10.3). Це стандартні об'єкти відкритого додатка Microsoft Office (документ редактора MS Word, книга і її листи MS Excel), форми, модулі і класи.

**Вікно властивостей** відображує різні атрибути виділеного об'єкту (рис. 10.4). Всі об'єкти (форми, елементи, що управляють, і так далі) мають атрибути, які змінюють не лише зовнішній вигляд об'єкту, але і його поведінку.

Всі ці атрибути називаються властивостями. Отже, кожен об'єкт володіє набором властивостей.

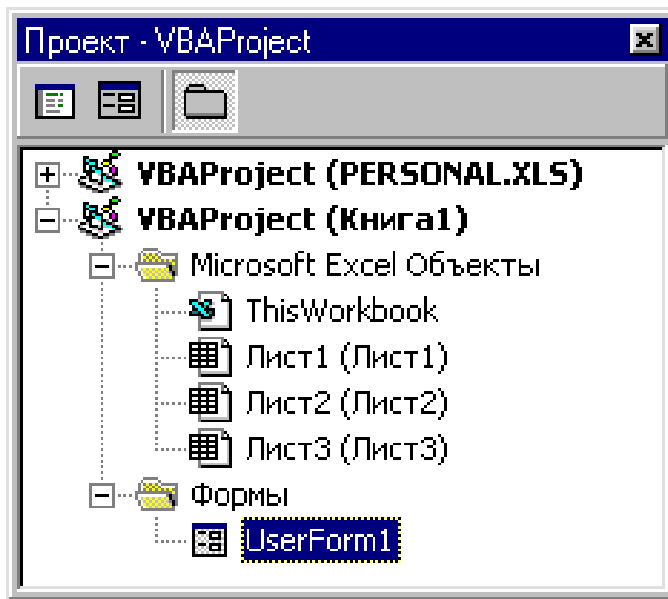


Рис. 10.3 - Вікно проекту

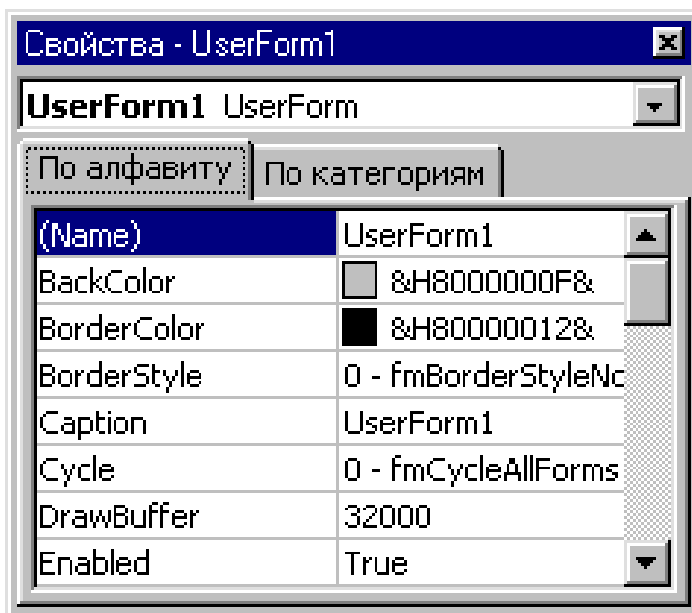


Рис. 10.4 - Вікно властивостей


**Вікно контрольних значень** дозволяє переглядати значення контрольних змінних в процесі перевірки правильності роботи (відладки) проекту, що дозволяє знаходити помилки в логіці роботи програм.

**Конструктор форм** розташований в центрі екрану редактора VBA. Тут виводиться або зображення форми, що дозволяє проводити візуальне конструювання макету форми і розташованих на ній елементів, або вікно програми (детальніше про це буде розказане нижче).



## 10.2 Особливості програмування на мові VBA

Процес розробки програми на мові VBA – **проекту**, може складатися з декількох етапів, залежно від кінцевого результату. Якщо необхідно отримати програму, яка проводитиме певні обчислення або дії, що розширюють математичні можливості стандартного застосування Microsoft Office, то досить створити **програмний модуль**. Для вживання цієї програми можна помістити в робочій області додатка кнопку, натиснення якої викликатиме виконання програми. Для цього в додатку необхідно включити панель інструментів за допомогою команди "**Вид**" → "**Панели інструментов**" → "**Элементы управления**", а потім створити кнопку з відповідним програмним кодом. Або виконувати програму за допомогою команди "**Сервис**" → "**Макрос**" → "**Макросы**".

Розробка "повноцінної" програми (для виконання якої потрібне окреме вікно, з різними елементами управління) включатиме два етапи. Перший етап – етап візуального програмування, на якому створюється вікно (**форма**) програми, де розташовуються необхідні елементи управління. Другий – етап програмування, на якому створюються частини програми (**процедури**), що виконуються у відповідь на певні події. Подією є, наприклад, клацання лівою кнопкою миші на командній кнопці (подія Click), натиснення клавіші на клавіатурі (подія KeyPress) і так далі Використовувати таке застосування можна натисненням кнопки  – "Запуск проекта".

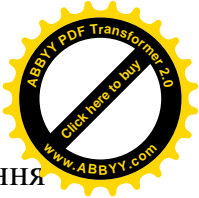
### 10.2.1 Об'єкти, властивості і методи VBA

Одним з основних понять VBA є об'єкт. **Об'єкт** – це те, чим ви управляєте за допомогою програми на мові VBA, наприклад, форма, кнопка, робочий аркуш або діапазон комірок MS Excel. Кожен об'єкт володіє деякими **властивостями**. Наприклад, форма може бути видимою або невидимою в даний момент на екрані. Інший приклад властивості об'єкту – шрифт для відображення інформації у комірці (об'єкті) робочого аркуша.

Об'єкт містить також список методів, які до нього застосовні. **Методи** – це те, що ви можете робити з об'єктом. Наприклад, показати форму на екрані або прибрати її можна за допомогою методів Show і Hide.

Таким чином, **об'єкт** – це програмний елемент, який має своє відображення на екрані, містить деякі змінні, визначальні його **властивості**, і деякі **методи** для управління об'єктом. Наприклад, в MS Excel є багато вбудованих об'єктів:

<b>Range("Адреса")</b>	- діапазон комірок (може включати лише одну комірку).
<b>Cells(i, j)</b>	- комірка, що знаходиться на пересіченні i-й рядка і j-го стовпця робочого аркуша MS Excel (i і j – цілі числа).
<b>Rows(№ рядки)</b>	- рядок із заданим номером.
<b>Columns(№ стовпця)</b>	- стовпець із заданим номером
<b>Sheets("Ім'я")</b>	- аркуш з вказаним ім'ям.
<b>Sheets(№ аркуша)</b>	- аркуш з вказаним номером.
<b>Worksheet</b>	- робочий аркуш.



Установка значень властивостей – це один із способів управління об'єктами. Синтаксис установки значення властивості об'єкту наступний:

**Об'єкт. Властивість = вираз**

Основною властивістю об'єктів **Cells** і **Range**, є **Value** (значення), яке, проте, можна не вказувати. Наприклад:

**Range("A5:A10").Value = 0** або **Range("A5:A10")= 0** – в діапазон вічок A5:A10 заноситься значення 0.

**Cells(2, 4).Value = n** або **Cells(2, 4)= n** – у комірку, що знаходиться на пересіченні 2-го рядка і 4-го стовпця (комірка з адресою "D2"), заноситься значення змінної n.

Синтаксис читання властивостей об'єкту наступний:

**Змінна = Об'єкт. Властивість**

Наприклад:

**Xn = Cells(1, 2).Value** або **Xn = Range("B1").Value** – змінною Xn привласнюється значення з комірки B1 поточного робочого аркуша.

Синтаксис застосування методів до об'єкту:

**Об'єкт. Метод**

Наприклад:

**Sheets(2).Activate** – зробити активним аркуш з №2.

**Sheets("Діаграма").Delete** – видалити аркуш з ім'ям "Діаграма".

**Range("A5:A10").Clear** – очистити діапазон комірок A5:A10.

**Range("A2:B10").Select** – виділити діапазон комірок A2:B10.

У MS Excel є об'єкти, які містять інші об'єкти. Наприклад, робоча книга містить робочі листи, робочий аркуш містить діапазон комірок і так далі. Об'єктом самого високого рівня є **Application** (додатка). Якщо ви змінюєте його властивості або викликаєте його методи, то результат застосовується до поточної роботи MS Excel.

Наприклад:

**Application.Quit** - завершення роботи з Excel.

Відзначимо, що крапка після імені об'єкту може використовуватися для переходу від одного об'єкту до іншого.

Наприклад, наступний вираз очищає другий рядок робочого аркуша **Май** в робочій книзі **Звіт**:

**Application.Workbooks("Звіт").Worksheets("Май").Rows(2).Delete**

Потрібно відзначити наступне:

- Можна не писати ім'я об'єкту **Application**, оскільки це мається на увазі за умовчанням.
- При роботі з підоб'єктом вже активізованого об'єкту немає необхідності вказувати об'єкт, що містить його.
- VBA використовує деякі властивості і методи, які повертають об'єкт до якого вони відносяться (це дозволяє швидко вказувати потрібний об'єкт). Приклади таких властивостей: **ActiveCell** (активна комірка), **ActiveSheet** (активний аркуш), **ActiveWorkBook** (активна робоча книга). Так, встановити значення активної комірки можна таким чином:





ActiveCell.Value = "Да".

### 10.2.2 Опис даних

Всі об'єкти, якими оперує мова програмування VBA, відносяться до певного типу.

Тип даних визначає:

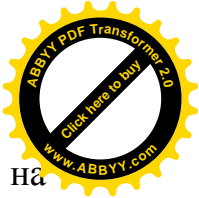
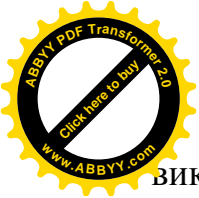
- область можливих значень змінної;
- структуру організації даних;
- операції, визначені над даними цього типу.

Типи даних підрозділяються на прості (скалярні) і складні (структуровані). У простих типів даних можливі значення даних єдині і неділимі. Складні ж типи мають структуру, в яку входять різні прості типи даних. Скалярні типи даних представлені в таблиці 10.1.

Таблиця 10.1 - Скалярні типи VBA

Ім'я типу	Українська назва типу	Можливі значення
Boolean	Логічний	True, False
Byte	Байтовий	0..255
Integer	Ціле	-32768..+32767
Long	Довге ціле	-2147483648..+2147483647
Single	Число з плаваючою крапкою	-3,4E38..-1,4E-45 для негативних значень. 1,4E-45..3,4E38 для позитивних значень.
Double	Число з плаваючою крапкою подвійної точності	-1,7E308..-4,9E-324 для негативних значень. 4,9E-324..1,7E308 для позитивних значень.
Currency	Грошовий	Десяткові числа з фіксованою позицією коми. Можливі 15 цифр до коми і 4 після.
String	Строковий	Є два види рядків: рядки фіксованої довжини (до 216 символів) і рядка змінної довжини (до 231 символу). Дані записуються в лапках.
Date	Дата	Дати змінюються в діапазоні від 1.01.100г. до 31.12.9999г.
Object	Об'єкт	Заслання на об'єкт (показчик)
Variant	Варіант	Універсальний тип, значенням якого можуть бути дані будь-якого з перерахованих вище типів, об'єкти, значення NULL і значення помилок ERROR.

Змінні в програмі можна описувати або не описувати. У останньому випадку їй буде привласнений тип **Variant**. Явно описувати змінну можна як на початку блоку, так і в будь-якому місці, де виникла необхідність



використовувати нову змінну. Краще всі змінні описувати явно і, як правило, на початку блоку. Для заборони використання змінних, які не були описані явно, в початок програми необхідно вставити оператора **Option Explicit**.

### 10.2.2.1 Опис простих змінних

Опис простих змінних має наступний синтаксис:

**Dim** *ІМ'Я\_ЗМІННОЇ* **As** *ІМ'Я\_ТИПА*

Одним оператором **Dim** можна описати довільне число змінних, але конструкція **As** має бути вказана для кожної з них, інакше змінним без **As** буде привласнений тип **Variant**.

Наприклад.

**Dim X As Byte, Z As Integer, C, Слово As String**

Тут змінна *X* - це змінна байтового типу, змінна *Z* - цілого типу, змінна *C* - типа варіант (за умовчанням), змінна *Слово* - строкового типу.

### 10.2.2.2 Опис констант

Дані програми, що не змінюються всередині, можна вважати константами. Їх можна описати таким чином:

**Const** *ІМ'Я\_КОНСТАНТИ* **As** *ІМ'Я\_ТИПА=ПОСТІЙНИЙ\_ВИРАЗ*

Наприклад.

**Const Pi As Double = 3.141593**

### 10.2.2.3 Опис масивів

Для зберігання векторів, матриць і так далі можна використовувати масиви.

**Масив** - це структурований тип даних, який є послідовністю елементів пам'яті, що мають загальне ім'я і що зберігають дані одного типу. Кожен елемент масиву визначається індексом (номером). Кількість елементів в масиві називається **розмірністю масиву**. Масив описується наступною конструкцією:

**Dim** *ІМ'Я\_МАСИВА(СПИСОК\_РОЗМІРНОСТЕЙ)* **As** *ІМ'Я\_ТИПА*

У списку розмірностей масиву кожен вимір відокремлюється комою і визначається задаванням нижньої і верхньої меж зміни індексів.

Наприклад.

**Dim X(1 TO 5) As Integer, Y(1 To 10, 1 To 20) As Double**

Тут *X* - одномірний масив, що складається з 5 елементів цілого типу, *Y* - двомірний масив, в якого 10 рядків і 20 стовпців з елементами числового типу подвійної точності.



### 10.2.3 Вирази

Вирази встановлюють порядок виконання дій над елементами даних. Вирази складаються з операндів і знаків операцій. Операндами є константи, змінні, покажчики функцій, вирази, взяті в дужки.

#### 10.2.3.1 Види операцій

Операції бувають арифметичні, відношення і логічні:

– **арифметичні операції:**

- ^ піднесення до ступеня
- \* множення
- / ділення
- ділення без остачі,
- mod залишок від ділення
- + плюс
- мінус;

– **операції відношення:**

- < менше
- > більше,
- <= менше або рівно,
- >= більше або рівно
- = рівно,
- <> не рівно;

– **логічні операції:**

- Not логічне заперечення,
- And логічне "И",
- Or логічне "ИЛИ".

Результатом логічної операції може бути одне з двох значень: True ("істина") або False ("неправда").

#### 10.2.3.2 Пріоритет виконання операцій

Якщо вираз містить декілька операцій, то пріоритет їх виконання наступний:

1. Спочатку виконуються арифметичні операції в такому порядку, як вони представлені в таблиці 10.2.

Таблиця 10.2 - Пріоритет арифметичних операцій

Опис операції	Позначення в VBA
Унарний мінус (зміна знаку)	- (на початку виразу)
Піднесення до ступеня	^
Множення і ділення	*, /
Ділення без остачі і залишок від ділення	\, Mod
Складання і віднімання	+, -



2. Далі операції відношення (вони мають однаковий пріоритет).
3. Останніми виконуються логічні операції в такому порядку, як вони представлені в таблиці 10.3.

Таблиця 10.3 - Пріоритет логічних операцій

Опис операції	Позначення в VBA
Логічне заперечення	Not
Логічне "И"	And
Логічне "ИЛИ"	Or

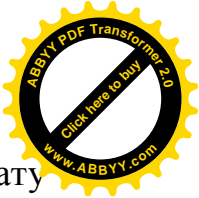
Якщо вираз містить декілька операцій однакового пріоритету, то порядок їх виконання зліва направо. Щоб змінити порядок дій у виразі використовуються круглі дужки.

Вирази бувають арифметичні, відношення і логічні.

**Арифметичні вирази** записуються за допомогою операндів числових типів і арифметичних операцій, а результатом є числове значення. У арифметичному виразі можна використовувати стандартні математичні функції, які приведені в таблиці 10.4.

Таблиця 10.4 - Стандартні математичні функції VBA

Математичний запис	Ім'я функції в VBA	Опис
$ X $	Abs(число)	Повертає значення, тип якого збігається з типом переданого аргументу, рівне абсолютному значенню вказаного числа.
$\arctg X$	Atn(число)	Повертає значення типа Double, що містить арктангенс числа.
$\cos X$	Cos(число)	Повертає значення типа Double, що містить косинус кута.
$]X[$	Int(число)	Повертає значення типа, співпадаючого з типом аргументу, яке містить цілу частину числа.
$\ln X$	Log(число)	Повертає значення типа Double, що містить натуральний логарифм числа.
$e^X$	Exp(число)	Повертає значення типа Double, що містить результат зведення числа $e$ (підстава натуральних логарифмів) у вказану міру.
Sign X	Sgn(число)	Повертає значення типа Variant (Integer), відповідне знаку вказаного числа.
$\sin X$	Sin(число)	Повертає значення типа Double, що містить синус кута.
$\sqrt{X}$	Sqr(число)	Повертає значення типа Double, що містить квадратний корінь вказаного числа.
$\operatorname{tg} X$	Tan(число)	Повертає значення типа Double, що містить тангенс кута.



**Вирази відношення** визначають істинність або помилковість результату при порівнянні двох операндів. Порівнювати можна дані будь-якого однакового типу. Результат операції відношення лише логічний: True - "істина" або False - "неправда".

**Логічні вирази.** Результатом логічного виразу є логічне значення True або False. Простими видами логічних виразів є: логічна константа, логічна змінна, логічна функція, вираз відношення. Логічні операції виконуються лише над операндами логічного типу.

*Приклад.* Записати  $1 \leq X \leq 5$  і визначити значення вираження при  $X=3.1$   
Вираз в VBA виглядатиме так:

$$X \geq 1 \text{ And } X \leq 5$$

Результатом виразу буде *True*.

Нижче, в таблиці 10.5, приведений список функцій, які можуть бути побудовані за допомогою вбудованих математичних функцій.

Таблиця 10.5 - Похідні математичні функції

Матем. запис	Назва функції	Комбінація вбудованих функцій
sc X	Секанс	1/Cos(X)
csc X	Косеканс	1/Sin(X)
ctg X	Котангенс	1/Tan(X)
arcsin X	Арксинус	Atn(X/Sqr(-X*X+1))
arccos X	Арккосинус	Atn(-X/Sqr(-X*X+1))+2*Atn(1)
arcsc X	Арксеканс	Atn(X/Sqr(X*X-1))+Sgn((X)-1)* *2*Atn(1)
arccsc X	Арккосеканс	Atn(X/Sqr(X*X-1))+(Sgn(X)-1)* *2*Atn(1)
arcctg X	Арккотангенс	Atn(X)+2*Atn(1)
sh X	Гіперболічний синус	(Exp(X) -Exp(-X))/2
ch X	Гіперболічний косинус	(Exp(X)+Exp(-X))/2
th X	Гіперболічний тангенс	(Exp(X) -Exp(-X))/(Exp(X)+Exp(-X))
sch X	Гіперболічний секанс	2/(Exp(X)+Exp(-X))
csch X	Гіперболічний косеканс	2/(Exp(X) -Exp(-X))
cth X	Гіперболічний котангенс	(Exp(X)+Exp(-X))/(Exp(X) -Exp(-X))
arsh X	Гіперболічний арксинус	Log(X+Sqr(X*X+1))
arch X	Гіперболічний арккосинус	Log(X+Sqr(X*X-1))
arth X	Гіперболічний арктангенс	Log((1+X)/(1-X)) /2
arsch X	Гіперболічний арксеканс	Log((Sqr(-X*X+1)+1)/X)
arsch X	Гіперболічний арккосеканс	Log((Sgn(X)*Sqr(X*X+1)+1)/X)
arcth X	Гіперболічний арккотангенс	Log((X+1)/(X-1)) /2
log <sub>n</sub> X	Логарифм по підставі N	Log(X) /Log(N)



## 10.3 Основні оператори мови VBA

### 10.3.1 Правила запису операторів

При написанні операторів необхідно дотримуватися наступних правил:

- Кожен новий оператор може записуватись з нового рядка.
- Щоб записати декілька операторів на одному рядку, їх розділяють між собою двокрапкою (:).
- Якщо оператор не поміщається в одному рядку, то необхідно поставити наприкінці рядка пропуск і знак підкреслення ( \_ ), а потім продовжити частину, що не помістилася, на наступному рядку.

### 10.3.2 Оператор присвоєння

Оператор присвоєння має наступний синтаксис:

*ІМ'Я\_ЗМІННОЇ = ВИРАЗ*

### 10.3.3 Оператори вводу-виводу

#### 10.3.3.1 Оператор і функція *MsgBox*

Оператор **MsgBox** здійснює виведення інформації в діалоговому вікні і встановлює режим чекання натиснення кнопки користувачем. Він має наступний синтаксис:

**MsgBox** *ПОВІДОМЛЕННЯ*[,*КНОПКИ*][, *ЗАГОЛОВОК*]

Аргументи:

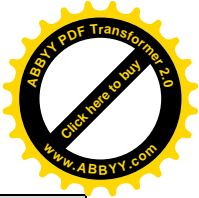
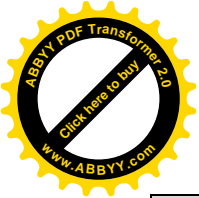
*ПОВІДОМЛЕННЯ* - обов'язковий аргумент, який задає у вікні інформаційне повідомлення, що виводиться. Може складатися з декількох текстових рядків, об'єднаних знаком **&**. Використання в цьому аргументі **Chr(13)** наводить до переходу на новий рядок при виведенні інформації.

*КНОПКИ* - значення цього аргументу визначає категорії кнопок, що з'являються у вікні. Від значення аргументу кнопки залежить також, чи з'являється у вікні який-небудь значок. Якщо не вказано, які кнопки необхідно відображувати у вікні повідомлень, то використовується значення за умовчанням, відповідне кнопці ОК. У таблиці 10.6 приведені можливі комбінації кнопок і значків у вікні повідомлень.


*ЗАГОЛОВОК* - задає заголовок вікна.

Функція **MsgBox** повертає значення типа **Integer**, що вказує, яку кнопку натискували в діалоговому вікні.





Таблиця 10.6 - Допустимі значення змінної кнопки

Відображення	Аргумент
Кнопка "ОК"	VbOKOnly
Кнопки "ОК" та "Отмена"	VbOKCancel
Кнопки "Да" та "Нет"	VbYesNo
Кнопки "Да", "Нет" та "Отмена"	VbYesNoCancel
Кнопки "Прекратити", "Повторити" та "Игнорировать"	VbAbortRetryIgnore
Кнопки "Повторити" та "Отмена"	VbRetryCancel
Інформаційний знак	VbInformation
Знак 	VbCritical
Знак питання	VbQuestion
Знак оклику	VbExclamation

### 10.3.3.2 Функція InputBox

Функція **InputBox** здійснює введення значень змінних за допомогою вікна введення і має наступний синтаксис:

*ІМ'Я\_ЗМІННОЇ* = **InputBox**( *ПОВІДОМЛЕННЯ* [, *ЗАГОЛОВОК*] )

Аргументи:

*ПОВІДОМЛЕННЯ* - обов'язковий аргумент. Задає у вікні інформаційне повідомлення, що зазвичай пояснює сенс величини, що вводиться

*ЗАГОЛОВОК* - задає заголовок вікна.

### 10.3.4 Умовний оператор IF

Для реалізації обчислювального процесу, що розгалужується, в VBA використовується оператор **If...Then...Else**, який є простою формою перевірки умов. Він має наступний синтаксис:

**If** *УМОВА* **Then** *ОПЕРАТОР\_1* **Else** *ОПЕРАТОР\_2*

*ОПЕРАТОР\_1* виконується, якщо *УМОВА* істинна, інакше виконується *ОПЕРАТОР\_2*. При цьому оператор **If...Then...Else** записується в один рядок.

*УМОВА* – це вираз логічного типу. Результат виразу завжди має булевого типу. Вираз може бути простим і складним. При записі простих умов можуть використовуватися всі можливі операції відношення, вказані в таблиці 10.7.



Таблиця 10.7 - Логічні відношення

Операція	Назва	Вираз	Результат
=	Рівно	$A = B$	True, якщо A рівно B
$\neq$	Не рівно	$A \neq B$	True, якщо A не рівно B
>	Більше	$A > B$	True, якщо A ,більше B
<	Менше	$A < B$	True, якщо A менше B
$\geq$	Більше або рівно	$A \geq B$	True, якщо A більше або рівно B
$\leq$	Менше або рівно	$A \leq B$	True, якщо A менше або рівно B

Складні умови утворюються з простих шляхом застосування логічних операцій і круглих дужок. Список логічних операцій приведений в таблиці 10.8.

Таблиця 10.8 - Логічні операції

	Назва	Вираз	A	B	Результат
<b>Not</b>	Логічне заперечення	Not A	False True		True False
<b>And</b>	Логічне "И"	A And B	True True False False	True False True False	True False False False
<b>Or</b>	Логічне "ИЛИ"	A Or B	True True False False	True False True False	True True True False

У умовному операторі допустиме використання блоку операторів замість будь-якого з операторів. В цьому випадку умовний оператор має вид:

```

If УМОВА Then
    БЛОК_ОПЕРАТОРІВ_1
Else
    БЛОК_ОПЕРАТОРІВ_2
End If
    
```

У умовному операторі може перевірятися декілька умов. В цьому випадку умовний оператор має вид:



```
If УМОВА_1 Then  
    БЛОК_ОПЕРАТОРІВ_1  
ElseIf УМОВА_2 Then  
    БЛОК_ОПЕРАТОРІВ_2  
Else  
    ....  
End If
```

### 10.3.5 Оператори циклу

Для реалізації циклічного обчислювального процесу, тобто багатократного виконання одного або декількох операторів, служить оператор циклу **For...Next**, який має наступний синтаксис:

```
For ЛІЧИЛЬНИК=ПОЧ_ЗНАЧЕННЯ To КІН_ЗНАЧЕННЯ Step КРОК  
    БЛОК_ОПЕРАТОРІВ  
    [Exit For]  
    БЛОК_ОПЕРАТОРІВ  
Next ЛІЧИЛЬНИК
```

Цикл **For...Next** перебирає значення змінної *ЛІЧИЛЬНИК*, яка є параметром циклу, від початкового до кінцевого значення з вказаним кроком зміни. При цьому забезпечується виконання блоку операторів тіла циклу при кожному новому значенні лічильника. Якщо **Step КРОК** в конструкції відсутній, то за умовчанням вважається, що крок дорівнює 1. По операторові **Exit For** можна вийти з оператора циклу до того, як *ЛІЧИЛЬНИК* досягне останнього значення.\*

Для перебору об'єктів з групи подібних об'єктів, наприклад, комірок з діапазону або елементів масиву, зручно використовувати оператора циклу **For...Each...Next**.

```
For Each ЕЛЕМЕНТ In ГРУПА  
    БЛОК_ОПЕРАТОРІВ  
    [Exit For]  
    БЛОК_ОПЕРАТОРІВ  
Next ЕЛЕМЕНТ
```

У VBA для організації циклів з невідомим заздалегідь числом повторень використовуються і інші оператори циклу:

---

\* **Примітка.** Не рекомендується примусово змінювати значення параметра циклу, його початкового і кінцевого значення в тілі циклу **For...Next**.



цикли з передумовою – **Do While ... Loop,**  
**Do Until ... Loop;**

цикли з постумовою – **Do ... Loop While,**  
**Do ... Loop Until.**

Нижче приведенний синтаксис цих операторів циклу:

' Цикл з передумовою **Do While ... Loop**

**Do While** УМОВА  
    БЛОК\_ОПЕРАТОРІВ  
    [Exit Do]  
    БЛОК\_ОПЕРАТОРІВ  
**Loop**

' Цикл з передумовою **Do Until ... Loop**

**Do Until** УМОВА  
    БЛОК\_ОПЕРАТОРІВ  
    [Exit Do]  
    БЛОК\_ОПЕРАТОРІВ  
**Loop**

' Цикл з постумовою **Do ... Loop While**

**Do**  
    БЛОК\_ОПЕРАТОРІВ  
    [Exit Do]  
    БЛОК\_ОПЕРАТОРІВ  
**Loop While** УМОВА

' Цикл з постумовою **Do ... Loop Until**

**Do**  
    БЛОК\_ОПЕРАТОРІВ  
    [Exit For]  
    БЛОК\_ОПЕРАТОРІВ  
**Loop Until** УМОВА

Оператор **Do While...Loop** забезпечує багатократне повторення блоку операторів до тих пір, поки *УМОВА* дотримується, а оператор **Do Until...Loop** поки *УМОВА* не дотримується. Оператори **Do...Loop While**, **Do...Loop Until** відрізняються від перерахованих вище операторів тим, що спочатку блок операторів виконується принаймні один раз, а потім перевіряється *УМОВА*. Для уникнення зациклення в тілі циклу має бути хоч би один оператор, який змінює значення змінних, що стоять в *УМОВІ*.

Оператор **Exit Do** забезпечує достроковий вихід з оператора циклу.



## 10.4 Структура програми. Модулі, процедури та функції

**Модуль** є текстовим ASCII-файлом з програмним кодом, що містить підпрограми, змінні і константи. Проект може складатися з безлічі програмних модулів. Для їх створення необхідно виконати команду "Вставка" → "Модуль".

Робоче вікно модуля представлено на рис. 10.5.

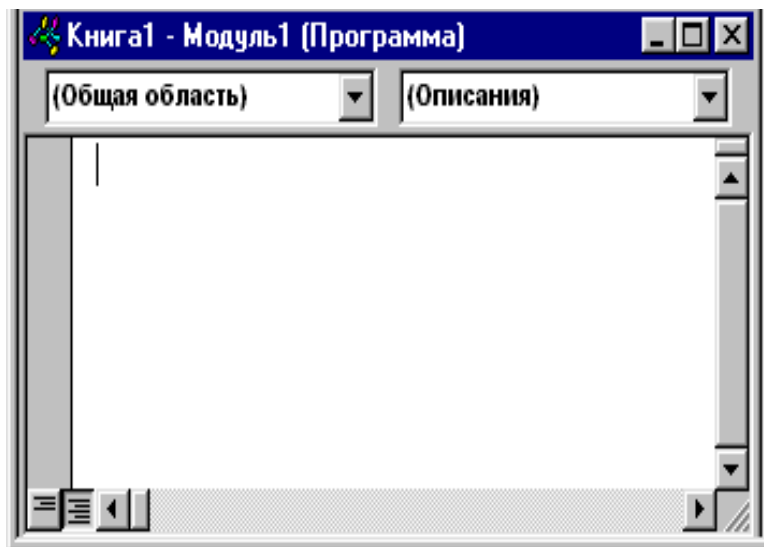


Рис. 10.5 - Вікно модуля

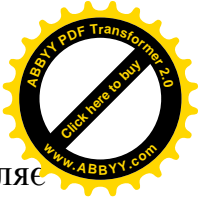
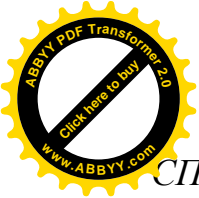
Основу програм в VBA складають процедури і функції.

**Процедура Sub** – це відособлена сукупність операторів VBA, що виконує певні дії. У загальному випадку процедура приймає деякі параметри (змінні, які передаються процедурі як похідні дані), виконує програму і може повертати результуючі значення, які привласнюються параметрам усередині процедури. Проте частіше використовуються процедури без параметрів.

Наприклад, процедури, що виконуються при виникненні певних подій. Вкладеність процедур в інші процедури не допускається. Структура процедури наступна:

```
[ДОСТУП] Sub ІМ'Я_ПРОЦЕДУРИ ([СПИСОК_ПАРАМЕТРІВ])  
    ТІЛО_ПРОЦЕДУРИ  
End Sub
```

Ключове слово *ДОСТУП* є необов'язковим і визначає зону видимості процедури. **Public** вказує, що процедура доступна для всіх інших процедур у всіх модулях (глобальна). **Private** вказує, що процедура доступна для інших процедур лише того модуля, в якому вона описана (локальна).



*СПИСОК ПАРАМЕТРІВ* також є необов'язковим елементом і дозволяє передавати процедурі різні похідні дані при виклику, які називаються формальними параметрами. При цьому ключове слово **Dim** не вказується. *ТИЛО ПРОЦЕДУРИ* складається з описової частини і блоку операторів, що виконуються один за іншим. Якщо необхідно припинити виконання процедури в деякому конкретному місці, це можна зробити за допомогою оператора **Exit Sub**. *ІМ'Я ПРОЦЕДУРИ* – це будь-який ідентифікатор, визначений користувачем. **Ідентифікатор** – це послідовність букв, цифр і символу підкреслення, що починається з букви (пропуски усередині ідентифікатора недопустимі). Ім'я процедури завжди визначається на рівні модуля. Для використання процедури в тексті програми (тобто для її виклику), необхідно вказати ім'я процедури і список фактичних параметрів, які повинні за типом і порядком розташування збігатися з формальними параметрами.

**Функція Function** багато в чому схожа на процедуру, але на відміну від неї після виклику завжди повертає значення. Функція отримує параметри, звані **аргументами**, і виконує з ними деякі дії, результат яких повертається функцією. Структура функції наступна:

[ДОСТУП] **Function** *ІМ'Я\_ФУНКЦІЇ*(*СПИСОК\_АРГУМЕНТІВ*) **As** *ТИП*  
*ТИЛО\_ФУНКЦІЇ*  
*ІМ'Я\_ФУНКЦІЇ* = *ВИРАЗ*

**End Function**

*ТИП* визначає тип даних повертаного результату. У тілі функції обов'язково має бути присутнім, принаймні, один оператор, що присвоює імені функції значення обчислюваного виразу. Дострокове завершення функції можливе за допомогою оператора **Exit Function**. У програмі виклик функції здійснюється за допомогою оператора присвоювання, в правій частині якого вказується ім'я функції з переліком фактичних параметрів, як і будь-якій іншій вбудованій функції, наприклад, Sqr, Cos або Chr.

Процедури і функції, які не описані явно за допомогою ключових слів **Public** або **Private**, за умовчанням є загальними.

Для швидкого додавання в модуль підпрограм зручно скористатися командою **"Вставка" → "Процедура"**. У вікні (рис. 4.2), що з'явиться, потрібно вибрати необхідні опції.

У MS Excel з функціями, створеними користувачем, можна працювати за допомогою Майстра функцій так само, як і зі вбудованими функціями робочого аркуша.

Приклад. У MS Excel створити функцію користувача, математично визначену таким чином:

$$y = \sin(x) \cdot e^{-5x}$$

Створимо модуль, як вказано вище (рис. 10.6), і введемо в нього текст наступної програми:



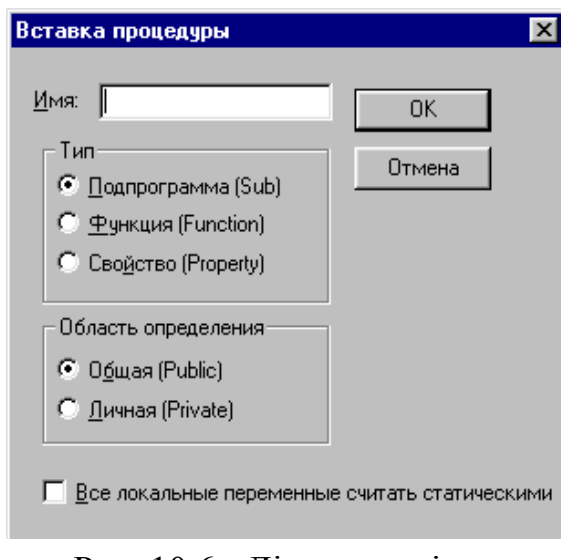


Рис. 10.6 - Діалогове вікно

“Вставка процедуры”

### Public Function Y(x As Single) As Single

$Y = \text{Sin}(x) * \text{Exp}(- 5 * x)$

End Function

Для використання створеної функції на робочому аркуші MS Excel введемо у комірку A2 число 0.1. У комірці B2 обчислимо значення функції Y при  $x = 0.1$ . Для цього у комірку B2 досить ввести формулу  $=Y(A2)$ . Це можна зробити і за допомогою Майстра функцій, який міститиме функцію Y разом з іншими вбудованими функціями MS Excel (рис. 10.7).

Загальні принципи організації програм VBA в модулі наступні. Зазвичай текст програми починається з опцій, які управляють описом змінних, способом порівняння рядків і так далі

Потім розташовують оголошення **глобальних** для даного модуля змінних і констант, тобто таких, які використовуються у всіх процедурах модуля.

Далі розташовують безпосередньо текст функцій і процедур, які складають саму програму.

Роздільником операторів в одному рядку при записі програми є символ “.”.

Для перенесення оператора на інший рядок використовується символ “\_” (знак підкреслення).

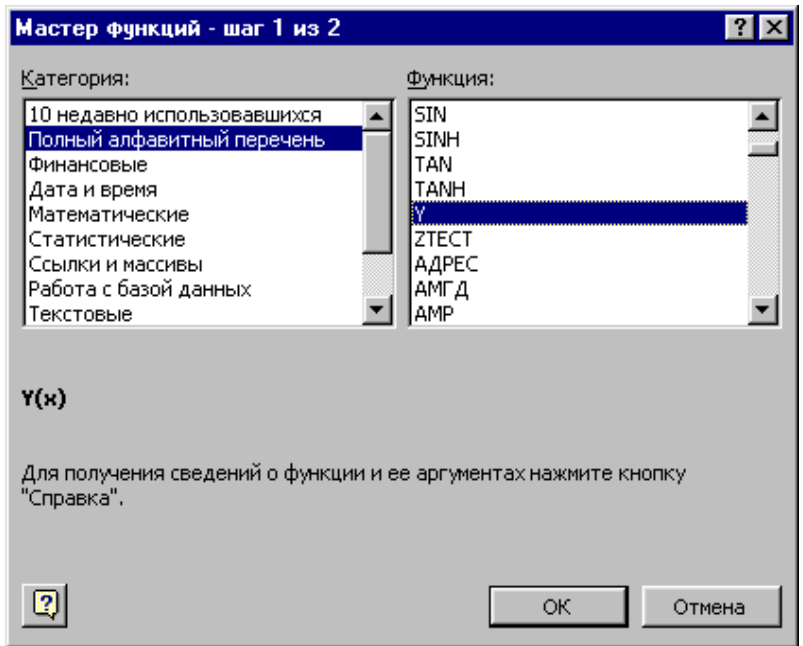


Рис. 10.7 - Виклик функції Y( )

Інколи усередині програми зручно поміщати **коментарі** – текст пояснення, який ігнорується компілятором і може бути записаний в будь-якому місці програми. Коментарі зручно також використовувати при відладці програми для тимчасового відключення операторів. Кожен рядок коментарів починається із знаку апострофа.

Приклад організації модуля:

**Option Base 1**

**Option Explicit**

*' PI – глобальна константа*

**Const PI As Double = 3.14159**

*' x - глобальна змінна*

**Dim x As Double**

*' Функція Disc вичисляє площу круга*

**Public Function Disc(R As Double) As Double**

**x = 2**

**Disc = PI \* R ^ 2**

**End Function**

*' Функція Rec вичисляє площу трикутника*

**Public Function Rec(a As Double, b As Double, c As Double) As Double**

*' p – локальна змінна*

**Dim p As Double**

**p = ( a + b + c) / 2**

**Rec = Sqr ( p \* ( p - a) \* ( p - b) \* ( p - c))**

**End Function**



*Процедура Result вызывает функции Disc и Rec и выводит результаты*

**Public Sub Result ( )**

*'R\_1, R\_2, a, b, c - локальные переменные*

**Dim R\_1 As Double**

**Dim R\_2 As Double**

**Dim a As Double**

**Dim b As Double**

**Dim c As Double**

**R\_1 = Disc( 2.5 )**

**x = x + 2**

**MsgBox "Площадь круга = " & CStr( R\_1) & ", x = " & CStr( x)**

**a = 1**

**b = 1**

**c = Sqr( 2)**

**R\_2 = Rec( a, b, c)**

**MsgBox "Площадь треугольника = " & CStr( R\_2)**

**End Sub**

Тут інструкція **Option Explicit** вказує на необхідність опису типів змінних, використовуваних на даному аркуші модуля.

Інструкція **Option Base 1** вказує, що індексація елементів масиву починатиметься з 1.

Функція **Sqr** обчислює квадратний корінь аргументу.

Функція **CStr** переводить числовий формат в строковий.

## 10.5 Створення екранних форм і включення їх в проекти

### 10.5.1 Створення форм. Властивості, події і методи форм

**Форма** - це головний об'єкт, що створює візуальну основу додатка.

За своєю суттю форма є вікном, в якому можна розміщувати різні елементи, що управляють, при створенні додатків.

Для створення форми необхідно виконати команду **"Вставка" → "UserForm"**. У вікні Конструктора форм з'явиться форма, що має стандартний вигляд для ОС Windows (рис. 10.8).

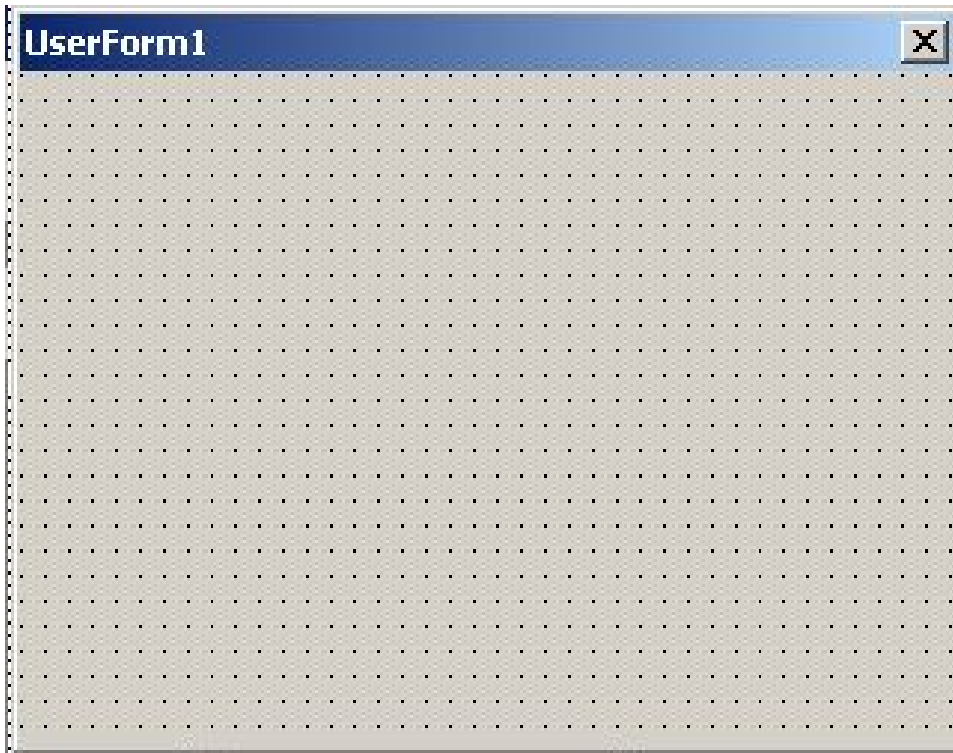


Рис. 10.8 - Об'єкт форма

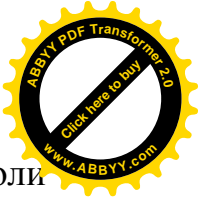
Як і будь-який інший об'єкт VBA форма має набір **властивостей**, основні з яких приведені в таблиці 10.9. Для здобуття довідки по будь-якій властивості досить виділити його у вікні властивостей і натискувати F1.

Таблиця 10.9 - Основні властивості форми

Властивість	Опис
<i>BackColor</i>	Колір фону для форми.
<i>BorderStyle</i>	Визначає тип межі, що оточує форму
<i>Caption</i>	Текст, який виводиться в заголовок форми.
<i>Font</i>	Визначає тип і вигляд шрифту у формі.
<i>Height</i>	Визначає висоту форми в твіпах.
<i>(Name)</i>	Ім'я об'єкту, для програми VBA.
<i>Width</i>	Визначає ширину форми в твіпах.

Властивості можна змінювати в режимі конструювання у вікні властивостей, або програмно в режимі виконання. Наприклад, в ході виконання програми можна змінити заголовок форми командою:

```
frmForm1.Caption = "Привет"
```



Програми в ОС Windows управляються **подіями**. Кожного разу, коли натискується кнопка, переміщається миша, змінюються розміри форми і так далі, ОС генерує повідомлення. Повідомлення доставляється відповідному об'єкту, наприклад формі, а та генерує відповідну подію. Отже, можна скласти фрагмент програми, в якому об'єкт реагуватиме на подію певним чином, тобто будь-якій стандартній події відповідає певна процедура. Щоб проглянути події пов'язані з формою, необхідно в режимі конструювання двічі клацнути на ній - з'явиться вікно програми, в якому клацнути на списку "Процедура". У таблиці 10.10 приведені найбільш часто використовувані події.

Таблиця 10.10 - Основні події форм

Подія	Опис
<i>Initialize</i>	Відбувається під час конфігурації і до завантаження форми в пам'ять.
<i>Activate</i>	Відбувається після завантаження форми в пам'ять.
<i>Deactivate</i>	Відбувається, якщо форма перестає бути активною.
<i>Click</i>	Відбувається при натисненні лівої кнопки миші на формі.

Наступний приклад змінює заголовок форми при активізації, і зменшує розмір форми після клацання лівою кнопкою миші на формі.

```
Private Sub UserForm_Activate()
    frmForm1.Caption = "Щелчок на формі зменшує її розміри"
End Sub
```

```
Private Sub UserForm_Click()
    frmForm1.Width = frmForm1.Width / 2
    frmForm1.Height = frmForm1.Height / 2
    frmForm1.Caption = "Сделай это еще раз!"
End Sub
```

Також форма володіє набором методів і інструкцій. **Метод** визначає дію, яка може бути виконана з об'єктом. **Інструкція** ініціює дію. Вона може виконати метод або функцію, В таблицях 10.11 і 10.12 приведені найбільш часто використовувані методи і інструкції для роботи формами.

Таблиця 10.11 - Основні методи форм

Метод	Опис
<i>Hide</i>	Приховує об'єкт UserForm, але не вивантажує його.
<i>Show</i>	Виводить на екран об'єкт UserForm.



Таблиця 10.12 - Основні інструкції форм

Інструкція	Опис
<i>Load</i>	Завантажує об'єкт UserForm, але не відображує його на екрані.
<i>Unload</i>	Видаляє об'єкт UserForm з пам'яті.

У наступному прикладі передбачається, що в проекті створено дві форми frmForms. При запуску проекту відбувається подія Initialize для форми frmForm1, форма frmForm2 завантажується і виводиться на екран. Коли за допомогою миші вибирається frmForm2, вона робиться невидимою, і з'являється форма frm Form1. Якщо ж вибирається frmForm1, frmForm2 з'являється знов.

'Событие Initialize формы frmForm1.

**Private Sub UserForm\_Initialize()**

**Load frmForm2**

**frmForm2.Show**

**End Sub**

' Событие Click для формы frmForm2

**Private Sub UserForm\_Click()**

**frmForm2.Hide**

**End Sub**

' Событие Click для формы frmForm1

**Private Sub UserForm\_Click()**

**frmForm2.Show**

**End Sub**

### 10.5.2 Вибір і використання елементів, що управляють

Створення елементів, що управляють, на формі виконується за допомогою **Панелі інструментів**, яка виводиться на екран командою "Вид" → "Панель елементів" (рис. 10.9).

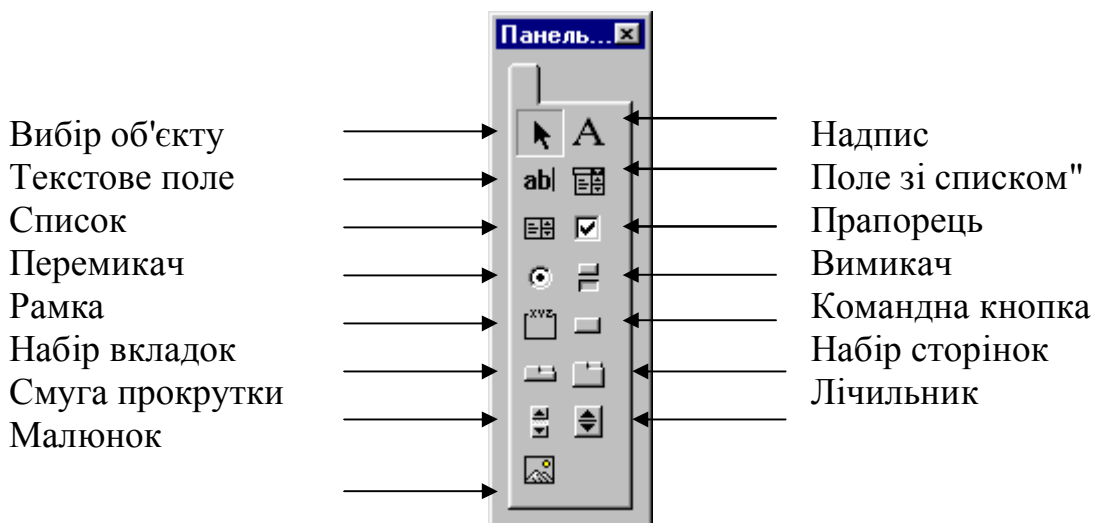
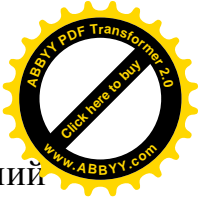


Рис. 10.9 - Панель елементів VBA





За допомогою кнопок цієї панелі можна помістити на форму необхідний елемент управління. Для цього потрібно клацнути на значку елементу управління, далі при натиснутій лівій кнопці миші визначити розмір і місце розташування елементу на формі. Коли елемент на формі виділений (рамка об'єкту містить маленькі прямокутники) можна змінювати його розміри і переміщати за допомогою миші, а також переглядати і змінювати його властивості у вікні властивостей.

Кожен елемент (об'єкт), що управляє, характеризується набором властивостей (які можна змінювати в режимах конструювання або виконання), подій і методів.

Для кожного об'єкту проекту необхідно визначити його ім'я. Відповідно до загальноприйнятих угод про імена об'єктів перші три символи імені повинні відображати вид елементу, а останні символи - призначення. У таблиці. 10.13 представлені поєднання перших трьох символів для найчастіше використовуваних елементів.

Таблиця 10.13 - Рекомендоване поєднання перших трьох символів імен

Об'єкт	Перші 3 символи імені	Приклад імені
Форма	<b>frm</b>	<b>frmMyForm</b>
Надпис	<b>lbl</b>	<b>lblInfo</b>
Текстове поле	<b>txt</b>	<b>txtInput</b>
Командна кнопка	<b>cmd</b>	<b>cmdExit</b>
Прапорець	<b>chk</b>	<b>chkSound</b>
Перемикач	<b>opt</b>	<b>optLevel</b>
Список	<b>lsb</b>	<b>lsbTypes</b>
Рамка	<b>fra</b>	<b>fraChoices</b>
Смуга прокрутки	<b>vcb</b>	<b>vcbSpeed</b>
Малюнок	<b>pic</b>	<b>picChema</b>

**Командна кнопка** є найпоширенішим елементом управління, і може використовуватися для організації виконання обчислень і інших дій, виклику процедур і функцій користувача, відкриття форм і так далі. Основні властивості командної кнопки представлені в таблиці. 10.14. У властивості **Caption** можна ставити символ **&** перед буквою, яка використовуватиметься у поєднанні з клавішею **Alt** для прискореного доступу до кнопки. Також можна перейти до кнопки клавішею **Tab**, а потім натискувати **Enter**.



Таблиця 10.14 - Властивості командних кнопок

Властивість	Опис
<i>BackColor</i>	Колір фону кнопки.
<i>Caption</i>	Текст, який виводиться на кнопці.
<i>Enabled</i>	Значення False робить кнопку недоступною.
<i>Font</i>	Визначає тип і вигляд шрифту на кнопці.
<i>ForeColor</i>	Визначає колір шрифту на кнопці.
<i>(Name)</i>	Ім'я об'єкту, для програми VBA.
<i>Picture</i>	Додає малюнок на кнопку.
<i>PicturePosition</i>	Визначає розташування тексту і малюнка на кнопці.
<i>Visible</i>	Значення False робить кнопку невидимою.

Основною **подією** кнопки є **Click**. Для написання програмного коду, який виконуватиметься при натисненні командної кнопки, досить двічі клацнути на ній лівою кнопкою миші в режимі конструювання проекту.

Найбільш корисним **методом** командної кнопки є **SetFocus**, що дозволяє повернутися до кнопки (передати їй фокус). Наприклад, наступна команда дозволяє повернутися до кнопки за умовчанням після введення даних в текстове поле: **cmdMyButtum.SetFocus**

**Текстове поле** застосовується для введення або виведення інформації. Основні властивості текстового поля представлені в таблиці. 10.15.

Таблиця 10.15 - Властивості текстового поля

Властивість	Опис
<i>Enabled</i>	Значення False робить поле недоступним.
<i>Font</i>	Визначає тип і вигляд шрифту в текстовому полі.
<i>ForeColor</i>	Визначає колір шрифту в текстовому полі.
<i>(Name)</i>	Ім'я об'єкту, для програми VBA.
<i>MaxLength</i>	Визначає кількість символів, що вводяться, в текстове поле.
<i>PasswordChar</i>	Визначає символ, що відображується при введенні в текстове поле.
<i>Text</i>	Визначає вміст текстового поля.

Наприклад, для очищення вмісту текстового поля в ході виконання програми необхідно ввести в необхідному місці програмного коду команду:



`txtResult.Text="" "`

Основною подією текстового поля є **Change**, що відбувається при введенні або видаленні символів. Наприклад, команду `cmdMyButtum.SetFocus` можна помістити в процедуру події **Change** текстового поля.

**Надпис** застосовується як самостійно для виведення довідкової інформації, так і у вигляді "підказок" для текстового поля, списку або іншого елемента. Головна її відмінність від текстового поля в тому, що користувач не може змінити текст надпису (хоча його можна змінити як властивість під час виконання програми). Основні властивості надпису представлені в таблиці 10.16.

Таблиця 10.16 - Властивості надпису

Властивість	Опис
<i>Caption</i>	Визначає текст, що міститься в надписі.
<i>Font</i>	Визначає тип і вигляд шрифту надпису.
<i>ForeColor</i>	Визначає колір шрифту надпису.
<i>(Name)</i>	Ім'я об'єкту, для програми VBA.
<i>Picture</i>	Додає малюнок в надпис.
<i>PicturePosition</i>	Визначає розташування тексту і малюнка надпису.

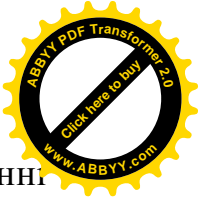
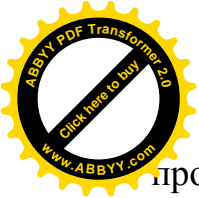
**Список** дозволяє працювати з переліком з декількох варіантів. Користувач може проглянути вміст списку і вибрати один з варіантів для подальшої обробки. Пряме редагування вмісту списку неможливе. Якщо в списку поміщаються не всі рядки, то автоматично додається вертикальна смуга прокрутки. Основні властивості списку представлені в таблиці 10.17.

Таблиця 10.17 - Властивості списку

Властивість	Опис
<i>(Name)</i>	Ім'я об'єкту, для програми VBA.
<i>ListIndex</i>	Повертає номер поточного виділеного рядка списку -1.
<i>Text</i>	Вміст поточного виділеного рядка списку.

Для списку найчастіше використовуються події **Click** і **DbClick** (подвійне клацання лівою кнопкою миші на одному з рядків списку). У другому випадку користувач одночасно виділяє рядок і починає її обробку.

Робота із списком починається з його заповнення методом **AddItem**, який може викликатися кілька разів підряд. Часто метод **AddItem** поміщається в



процедуру **UserForm\_Initialize()**, аби список заповнювався при завантаженні форми. Метод **RemoveItem** видаляє рядки із списку. Метод **Clear** очищає відразу весь список. Наступний приклад показує, як працюють списки, при цьому передбачається, що в проекті створена форма з двома списками (List1 і List2). Подвійне клацання на будь-якому рядку одного списку переміщає її в інший список. Рядок включається в інший список до того, як вона буде видалена з поточного.

```
Private Sub UserForm_Initialize()
```

```
    List1.AddItem "Стол"
```

```
    List1.AddItem "Стул"
```

```
    List1.AddItem "Диван"
```

```
    List1.AddItem "Кресло"
```

```
    List1.AddItem "Кровать"
```

```
End Sub
```

```
Private Sub List1_DblClick()
```

```
    List2.AddItem List1.Text
```

```
    List1.RemoveItem
```

```
    List1.ListIndex
```

```
End Sub
```

```
Private Sub List2_dblClick()
```

```
    List1.AddItem List2.Text
```

```
    List2.RemoveItem
```

```
    List2.ListIndex
```

```
End Sub
```

**Перемикачі** дозволяють вибрати один варіант з групи. Зазвичай вони групуються в рамках (див. далі), проте їх можна розташовувати прямо на формі, якщо використовується лише одна група перемикачів. Основні властивості перемикача представлені в таблиці 10.18.

Таблиця 10.118 - Властивості перемикача

<b>Властивість</b>	<b>Опис</b>
<i><b>Caption</b></i>	Задає текст, що визначає призначення перемикача.
<i><b>(Name)</b></i>	Ім'я об'єкту, для програми VBA.
<i><b>Value</b></i>	Значення True вказує, що перемикач вибраний.

Найбільш важливою є властивість **Value**, значення **True** (перемикач знаходиться у встановленому стані) якого в режимі конструювання задається

лише в одного перемикача в групі. У режимі виконання ця властивість найчастіше перевіряється в процедурі події Click кнопки, яка натиснута після установки потрібного перемикача, що дозволяє перевірити перед викликом наступної процедури деяку умову. Проте певні дії можна виконувати відразу ж після вибору перемикача в процедурі його події **Click**.

**Прапорець** частково аналогічний перемикачу, але на відміну від нього може використовуватися як окремий самостійний елемент. Навіть об'єднані в групу прапорці працюють незалежно один від одного. Основні властивості прапорців такі ж, як і в перемикача (див. таблицю 5.10). Проте властивість Value може набувати три значення (прапорець знаходиться у встановленому стані, знятому або невизначеному).

Найбільш часто використовуваною подією прапорців є **Click**, в процедурі якого можна перевіряти стан прапорця по властивості Value. Наступний приклад ілюструє роботу прапорців, при цьому передбачається, що в проекті створена форма з двома прапорцями (ChkBold і ChkItalic) і текстовим полем TxtExam (рис.10.10). Після введення символів в текстове поле, за допомогою прапорців можна робити текст напівжирним або курсивом. Властивості **FontBold** і **FontItalic** текстового поля встановлюють способи зображення тексту.

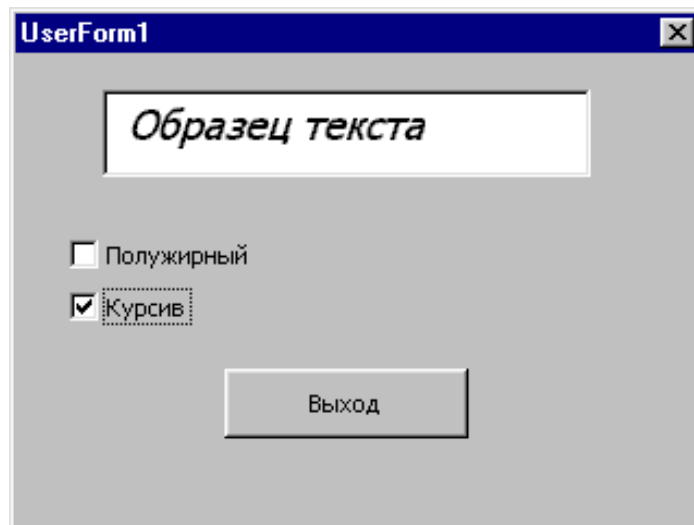


Рис. 10.10 - Використання прапорців

```

Private Sub Chkbold_Click()
  If ChkBold.Value = True Then
    TxtExam.FontBold = True
  Else
    TxtExam.FontBold = False
  End If
End Sub

```



```
Private Sub ChkItalic_Click()  
  If ChkBold.Value = True Then  
    TxtExam.FontItalic = True  
  Else  
    TxtExam.FontItalic = False  
  End If  
End Sub
```

**Рамка** використовується для групування перемикачів або прапорців, і поміщається на форму раніше елементів, що знаходяться усередині неї. Перемикачі рамки, що знаходяться усередині, працюють як самостійна група і не впливають на стан перемикачів в інших рамках. Основною властивістю рамки є **Caption**, яке задає текст, що визначає призначення елементів в рамці.

**Малюнок** використовується для простого виведення зображення на формі. Він може відображувати растрові файли (.BMP), значки (.ICO), метафайли (WMF), а також файли у форматі JPEG (.JPG) і GIF (.GIF). Основні властивості малюнка представлені в таблиці 10.19.

Таблиця 10.19 - Властивості малюнка

Властивість	Опис
<i>Autosize</i>	Значення True підганяє розмір елемента під розмір вмісту.
<i>(Name)</i>	Ім'я об'єкту, для програми VBA.
<i>Picture</i>	Задає файл для малюнка.

Події і методи рамок і малюнків практично не використовуються.





## 11 ЕКСПЕРТНІ І НАВЧАЛЬНІ СИСТЕМИ

**Експертна система (ЕС)** — це система, що використовує знання для забезпечення високоефективного вирішення задач у вузькій професійній області. Експертні знання в ЕС виділені у відособлену базу знань і отримані від експерта - людини, що за роки навчання і практики навчилася надзвичайно ефективно вирішувати задачі, що належать до такої області. Інструментальними засобами побудови ЕС служать мова програмування і підтримуючий пакет програм, які використовуються при створенні ЕС.

ЕС являють собою реальний практичний додаток штучного інтелекту, що підкреслено ще одним визначенням: ЕС - це заснована на знаннях певної комплексної предметної області інформаційна система, що виконує роль експерта-консультанта для кінцевих користувачів.

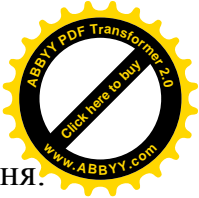
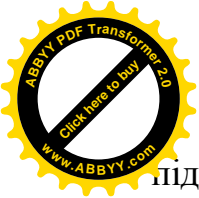
**Ресурсними компонентами ЕС** є апаратні, програмні і людські ресурси. *Апаратні ресурси* складаються з автономних мікрокомп'ютерних систем, а також мікрокомп'ютерних робочих станцій і терміналів, приєднаних до мінікомп'ютерів чи великих ЕОМ за допомогою телекомунікаційної мережі. Складні ЕС іноді розробляються на потужних комп'ютерах спеціального призначення, безпосередньо спроектованих для програмних пакетів розробки експертних систем або мов програмування ЛІСП чи ПРОЛОГ.

*Програмні ресурси* - це механізм виведення, а також інші програми для роботи зі знаннями і для зв'язку з кінцевими користувачами. Програми одержання знань не є частиною експертної системи, а є програмними засобами тільки для розробки бази знань.

Зручними засобами розробки ЕС є програми-оболонки експертних систем без її ядра (основного змісту), тобто без її баз знань і фактів. Загальновизнано також, що кінцевим користувачам і експертам великі практичні можливості створення ЕС надають засоби електронних таблиць (EXCEL, LOTUS 1-2-3 та ін.). Електронні таблиці називають також обмеженими генераторами підтримки прийняття рішень, оскільки вони надають користувачу кілька основних аналітичних інструментів („що-якщо", кореляційно-регресійний і деякі інші види статистичного аналізу, оптимізацію, побудову й аналіз трендів).

*Людські ресурси.* Коли створюється велика ЕС, то база знань і процес експертизи звичайно проектується інженером знань з фактів і правил, наданих експертом. ЕС дає рекомендації кінцевому користувачу. Експерти і кінцеві користувачі можуть бути і самі собі інженерами із знань, якщо вміють використовувати програмні оболонки ЕС чи інтелектуальні можливості електронних таблиць.

За сферами використання ЕС їх можна поділити на **виробничі й управлінські**. Виробничі ЕС дають експертний висновок щодо управління виробничими процесами, управлінські - допомагають менеджерам приймати рішення. Штучна експертиза постійна, несуперечлива, легко передається і уточнюється. Вона зв'язує комп'ютери з багатством людського досвіду і



підвищує цінність людських знань, надаючи їм широкого застосування. Порівняння людської і штучної компетентності представлено в таблиці 11.1:

Таблиця 11.1 - Порівняння людської і штучної компетентності

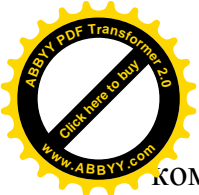
Людська компетентність	Штучна компетентність
Нестійка	Постійна
Важко передається	Легко передається
Важко документується	Легко документується ^
Непередбачувана	Стійка
Дорога	Прийнятна за витратами
Творча	Запрограмована
Пристосовувана	Має потребу в підказці
Використовує чуттєве сприйняття	Використовує символічне введення
Широка за охопленням	Вузьконапрявлена
Використовує загальнодоступні знання (здоровий глузд)	Використовує спеціалізовані знання

На відміну від звичайних програм, ЕС повинна мати набір таких властивостей: компетентність, символічне судження, глибина, самосвідомість.

*Компетентність* означає, що ЕС повинна досягати експертного рівня рішень - бути вмілою, тобто розмірковувати, виходячи з фундаментальних принципів для знаходження правильного рішення навіть у випадку деяких некоректних даних. Останню властивість називають також робастністю, тобто правильний логічний висновок може бути отриманий на основі знання фундаментальних принципів у разі недостачі або некоректності деяких фактів.

*Символьні судження.* Ця вимога означає, що експерти обходяться без розв'язання систем рівнянь або складних математичних формулювань, використовуючи знання, виражені звичайними символами рядків (наприклад, „платоспроможність”, „фінансова стійкість”, „рентабельність”).

Результат експертизи завжди виражений звичайними пропозиціями в термінології професійної області знання. Якщо комп'ютер має сучасні засоби мультимедіа, то ці пропозиції можуть бути навіть озвучені (промовлені)



комп'ютером. Символьні рядки поєднують у символічні структури (блоки) за їхніми логічними взаємозв'язками, що дозволяє переформулювати задачу, якщо це необхідно.

Вимога *глибини* означає, що ЕС повинна працювати в предметній області, що містить важкі задачі, а також використовувати складні правила. ЕС, що не володіє глибиною, вироджується в штучну задачу.

Наявність *самосвідомості* означає, що ЕС повинна бути здатна пояснювати свої висновки і дії.

Відомі 10 способів застосування ЕС:

1. *Інтерпретація* - опис ситуації за інформацією, що надходить від датчиків.
2. *Прогноз* - визначення ймовірних наслідків ситуацій. Приклади: прогноз поведінки виробничого агрегату, прогноз попиту на паливо, прогноз товарообігу, прогноз цін на товари і т.п.

Системи прогнозування іноді використовують імітаційне моделювання - програми, що відображають причинно-наслідкові зв'язки на основі яких за значеннями даних, що вводяться, генеруються різні ситуації.

Для таких цілей у економічному аналізі поряд із спеціальними ринковими програмами можна використовувати електронні таблиці, що відтворюють в електронній моделі арифметичні і логічні взаємозв'язки показників.

Наприклад, можна ввести в клітинку „Чисельність робітників” передбачуване число і відразу ж у вікні виводів одержати висновок „Несприятлива ситуація. Ріст продуктивності праці в порівнянні з минулим роком сповільниться”. Зміна значення іншого показника (наприклад, обсягу випуску продукції) може змінити попередній висновок і т.п.

3. *Діагностика* - виявлення причин неправильного функціонування системи за результатами спостережень.
4. *Проектування* - побудова конфігурації об'єктів при заданих обмеженнях.
5. *Планування* - визначення послідовності дій.
6. *Спостереження* - порівняння результатів спостережень з очікуваними результатами.
7. *Налагодження* - складання рецептів виправлення неправильного функціонування системи.
8. *Ремонт* - виконання послідовності запропонованих виправлень.
9. *Навчання* - діагностика, налагодження і виправлення поведінки того, кого навчають.



## 10. Управління - управління поведінкою системи як єдиного цілого.

Експертні системи належать до систем підтримки прийняття рішень (СППР), заснованих на знаннях.

Традиційні СППР універсальні і застосовуються для вирішення унікальних проблем у різних предметних областях, а ЕС дають відповіді на питання у вузькій предметній області і роблять висновки, які могла б зробити людина-професіонал високої кваліфікації.

Інтеграція традиційної СППР із ЕС утворить більш складний вид - так звану **експертну систему підтримки прийняття рішень** (ЕСППР). Така система, виходячи з загальних вимог, що ставляться до ЕС, повинна пояснювати свої поради кінцевому користувачу, крім того, надавати йому універсальні засоби вільного моделювання. Зазначені розбіжності між СППР і ЕС показано у таблиці 11.2. На рисунку 11.1 наведена концептуальна схема ЕСППР.

Таблиця 11.2 - Основні відмінності традиційних СППР і ЕС

Порівнювані ознаки	СППР	ЕС
Призначення	Сприяння людині в	Дублювання функцій експерта і
Хто приймає	Людина і/або система	Система
Основна орієнтація	Прийняття рішень	Обмін досвідом (людина-машина-людина) і вироблення рекомендацій
Особливості запиту	Людина запитує машину	Машина запитує людину
Метод маніпулювання	Числовий	Символьний
Предметна область	Комплексна, широка	Вузька, спеціалізована
Тип задач	Унікальні	Повторювані
Зміст бази даних	Фактичні дані (факти)	Факти і процедури їхнього використання
Здатність до міркувань	Немає	Є, обмежена
Здатність до пояснення	Обмежена	Є



Рис.11.1 - Структурна схема ЕСПІР (інтегрованої СПІР і ЕС)

ЕС добре вирішують вузькі специфічні проблеми у конкретній сфері знань, але програють у вирішенні задач, що вимагають широкого кругозору. ЕС ефективна для вирішення аналітичних задач.

Наприклад, ЕС допомагає фінансовому консультанту з інвестицій, який видає рекомендації для клієнтів. Однак ЕС не може оцінити нюанси поточної політики, економіки, соціального розвитку або поведінки споживача. Ці важливі фактори повинні аналізуватися консультантом-людиною.

Придбаваючи або розробляючи ЕС, необхідно осмислити і порівняти переваги експертної системи і витрати на неї.

Можливі випадки, коли людина-експерт вирішує задачу за кілька хвилин, а створення ЕС вимагає декількох сотень правил і кілька місяців проектування.

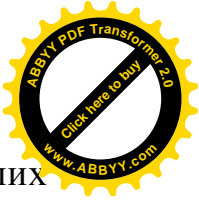
Звичайно великі ЕС розробляються інженерами знань методом прототипування, тобто поступовим наближенням від чорнового варіанта до кінцевої мети.

Інженер знань - це професіонал, що працює з експертами в пошуку знань (фактів і евристик), які вони обробляють. Інженер знань будує базу знань (а у разі необхідності, і всю ЕС) і повинен уміти працювати з експертами в багатьох предметних областях. За кордоном оболонки ЕС порівняно недорогі. Вони допомагають кінцевому користувачу розробляти власні експертні системи. Деякі оболонки використовують формат електронної таблиці, полегшуючи розробку правил „ЯКЩО, ТО”

### 11.1 Приклади експертних систем і систем підтримки прийняття рішень.

1. Першою сферою розробки ЕС була медицина. У середині 70-х років у Стенфордському університеті була розроблена ЕС „MYCIN", що





діагностує і визначає спосіб лікування менінгіту (й інших бактеріальних інфекцій) у перші 48 годин після зараження. ЕС „MYCIN” спроектована шляхом опитування великої кількості лікарів про їхні способи діагностики і лікування і містить близько 500 правил.

2. Компанія „Американ експрес” знизилася збитки від кредитних карток, використовуючи ЕС „Помічник того, що видає кредит”, яка дає поради про надання або відмовлення в кредиті. Індивідуальне рішення приймається не більш ніж за 90 секунд, тобто ризик неправильного рішення є значним. „Помічник того, що видає кредит” містить знання досвідчених співробітників, допомагаючи розпізнати нетипові запити і відкинути їх.
3. ЕС „ПланПауер” допомагає окремим особам у персональному фінансовому плануванні податків, нерухомості, регулюванні портфеля активів. ЕС враховує множину змінних факторів для рекомендації фінансового об'єкта або послуги. Вона враховує цілі і фінансову ситуацію клієнта, дозволяючи консультанту вести аналіз „що, якщо” за факторами процентних ставок, темпами інфляції та інш.
4. Розроблений в Інституті кібернетики ім. В. М. Глушкова Національної академії наук України інтелектуальний пакет Ppredictor дозволяє будь-якій організації швидко покращити процес прийняття рішень при плануванні продажів, у ціноутворенні, плануванні виробництва, фінансовому плануванні. Ppredictor функціонує в середовищі MS Excel, а його користувацький інтерфейс уніфікований з MS Office. Ppredictor - ефективний, практичний, простий в освоєнні й експлуатації інструмент експрес-прогнозування і складного аналізу часових рядів. Після інсталяції він автоматично інтегрується в середовище MS Excel і забезпечує роботу в чотирьох режимах:
  - для експрес-прогнозування: Ppredictor автоматично будує чотири кращі прогнози;
  - для новачків: Ppredictor забезпечує зрозуміле покрокове прогнозування з автоматичним встановленням оптимальних параметрів;
  - для досвідчених користувачів: Ppredictor полегшує одержання якісних прогнозів з покроковим ручним налаштуванням необхідних параметрів;
  - для професіоналів: Ppredictor дозволяє вставити функцію методу прогнозування прямо в Excel-таблицю. Цей режим дозволяє самостійно підібрати метод прогнозування, здійснюючи налаштування і управління прогнозуванням командними кнопками. Якість моделі можна оцінити візуально (за накладенням графіка прогнозу на вихідний ряд) і за іншими статистичними оцінками (на вибір користувача-професіонала).





## 12 ПЕРСПЕКТИВИ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

### 12.1 Сучасний стан інформаційних технологій

Сучасний стан інформаційних технологій характеризується наступними положеннями.

1. Наявність великої кількості промислово функціонуючих баз даних великого обсягу, що містять інформацію практично по усіх видах діяльності суспільства.
2. Створення технологій, що забезпечують інтерактивний доступ масового користувача до цих інформаційних ресурсів. Технічною основою даного положення з'явилися державні і приватні системи зв'язку і передачі даних загального призначення і спеціалізовані, об'єднані в національні, регіональні і глобальні інформаційно-обчислювальні мережі.
3. Розширення функціональних можливостей інформаційних систем, що забезпечують рівнобіжну одночасну обробку баз даних з різноманітною структурою даних, мультиоб'єктних документів, гіперсередовищ, у тому числі реалізуючі технології створення і ведення гіпертекстових баз даних. Створення багатфункціональних програмно-орієнтованих інформаційних систем різного призначення на основі могутніх персональних комп'ютерів і локальних обчислювальних мереж.
4. Включення в інформаційні системи елементів інтелектуалізації інтерфейсу користувача, експертних систем, систем машинного перекладу й інших технологічних засобів.

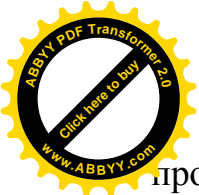
### 12.2 Тенденції розвитку інформаційних технологій

Виділяють 5 основних тенденцій у розвитку інформаційних технологій.

*Ускладнення інформаційних продуктів (послуг).* Інформаційний продукт у виді програмних засобів, баз даних і служб експертного забезпечення здобуває стратегічне значення.

*Здатність до взаємодії.* З ростом значимості інформаційного продукту можливість провести ідеальний обмін цим продуктом між комп'ютером і людиною або між інформаційними системами здобуває значення ведучої технологічної проблеми. Серед інших це проблема сумісності технічних і програмних засобів. Усі проблеми обробки передачі інформаційного продукту повинні знаходитися в повній відповідності по сумісності і швидкодії.

*Ліквідація проміжних ланок.* Розвиток здатності до взаємодії веде до удосконалювання процесу обміну інформаційним продуктом, а отже, при взаємовідношенні постачальників і споживачів у цій області ліквідуються



проміжні ланки. Не потрібні посередники, якщо істи можливість розміщати замовлення безпосередньо за допомогою інформаційних технологій.

*Глобалізація.* Фірми можуть за допомогою інформаційних технологій вести справи де завгодно, одержуючи вичерпну інформацію. Глобалізація ринку інформаційного продукту дозволяє одержати переваги за рахунок розподілу витрат на більш широкий географічний регіон.

*Конвергенція.* Зникають розходження між виробами і послугами, інформаційним продуктом і засобами, що використовуються в побуті і для ділових цілей, розходження між передачею звукових, цифрових і відеосигналів. Стосовно до бізнесу ці тенденції приводять до наступного.

- Здійснення розподілених персональних обчислень, коли на кожному робочому місці досить ресурсів для обробки інформації в місцях її виникнення;
- Створення розвинутих систем комунікації, коли підприємство включається у світовий інформаційний потік;
- Створення і розвиток систем електронної торгівлі;

Усунення проміжних ланок у системі інтеграції організація – зовнішнє середовище.

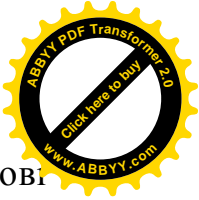
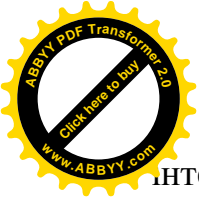
### 12.3 Системи штучного інтелекту

Перспективними інформаційними технологіями є системи штучного інтелекту (ШІ), які використовуються в даний час для підтримки прийняття рішень. Вони мають властивості розпізнавання паттернів (нейронні мережі і генетичні алгоритми), навчання й адаптації (нейронні мережі і генетичні алгоритми), тлумачення-пояснення (експертні системи, нечіткі системи, генетичні алгоритми), гнучкості-узагальнення (нейронні мережі, нечіткі системи, генетичні алгоритми).

Крім того, сучасні інтелектуальні системи в стані використовувати метадані, (метазнання) і виявляти нові відношення (закономірності) у масивах даних. Недоступні поки для систем ШІ „здатності” розробляти концептуальні моделі і оцінювати значення виявлених закономірностей. Тому ці компоненти і залишаються за людиною, також як і функція постановки проблем - формулювання задач.

Перераховані складові інтелекту людини, не підвладні сучасним системам ШІ, деякою мірою характеризують творчий процес, що відіграє значну роль у прийнятті управлінських рішень.

Найважливішими операціями мислення людини, як відомо, є *аналіз* (виділення в об'єкті елементів і зв'язків між ними), *синтез* (об'єднання компонентів, що складають об'єкт), *узагальнення* (об'єднання елементів у групи на основі загальних ознак), *порівняння* (ідентифікація подібності і розбіжності в об'єктах), *категоризація* (кластеризація - у термінології, прийнятій в штучному інтелекті). Досягнутий сьогодні рівень розробок у теорії і практиці штучного



інтелекту такий, що окремі технології здатні моделювати перераховані складові людського мислення.

Так, можна припустити, що аналіз і синтез стимулюються динамічними структурними діаграмами й експертними системами, які продукують спадні і висхідні висновки.

Операції узагальнення, порівняння і категоризації в тому чи іншому ступені моделюються штучними нейронними мережами, нечіткими системами і генетичними алгоритмами.

Ключовими характеристиками інтелектуальних систем, що визначають їхнє застосування в різних галузях діяльності, є такі:

- здатність до навчання;
- здатність до адаптації;
- гнучкість;
- „прозорість” тлумачення (пояснення);
- здатність відкривати нове.

Необхідно підкреслити, що всіма перерахованими можливостями і інтелектуальні системи володіють завдяки своїй внутрішній структурі і властивостям, що моделюють природні процеси.

У той же час не всі інтелектуальні технології поєднують у собі одночасно всі п'ять перерахованих характеристик. Кожна з них за сукупністю властивостей не є універсальною і має свої сильні і слабкі сторони.

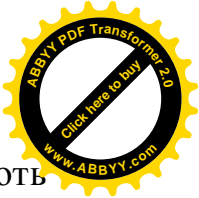
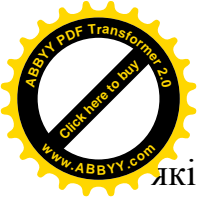
Ця властивість інтелектуальних систем є **найбільш** важливою і виражається в тому, що вони можуть після появи вхідних сигналів самоналаштуватися, забезпечуючи потім реакцію (вихід) з необхідною точністю.

У бізнесі ця здатність інтелектуальних систем проявляється в тому, що вони можуть виробляти рішення безпосередньо з даних, виводячи модель предметної області після проходження через сотні і тисячі операцій. Як правило, такі операційні знання властиві співробітникам, що мають багаторічний досвід роботи в організації.

До інтелектуальних систем, що самонавчаються, належать нейронні мережі, СВР-технології і генетичні алгоритми. У перших „поколіннях” інтелектуальних систем, до яких відносяться, насамперед, статичні експертні системи, знання, необхідні для виконання задачі, вводилися в систему „вручну” людиною-експертом.

Процес визначення знань для експертних систем вимагає затрат часу, є дорогим і потенційно ненадійним. Експерти вважають важкою задачею виразити декларативні (інтуїтивні) знання, а іноді просто відмовляються брати участь у довгій процедурі видобування знань.

Необхідно також враховувати, що залучення для формування бази знань деяких професіоналів, наприклад, таких, як фінансові брокери і страхові агенти,



які займаються оцінкою ризиків, вимагає значних коштів, які підвищують вартість створюваної системи. Отже, здатність навчатися безпосереднє на основі даних є особливо важливою.

Крім того, людині-професіоналу властиві деякі обмеження, наприклад, можливі пробіли в знаннях і суб'єктивність оцінок істинності знань. Часто залучені експерти мають різні точки зору щодо виконання поставлених завдань. На відміну від експертів, інтелектуальні системи, що самонавчаються, не залежать від суб'єктивних факторів, є більш послідовними, і в цьому їхня перевага.

Адаптація - властивість системи швидко налаштовувати свої параметри під умови змінного навколишнього середовища. Як відзначалося вище, бізнес піддається постійним змінам.

Причинами змін є, наприклад, нестабільність у політичній ситуації і макроекономіці, поява нових конкурентів і законів, що регламентують функціонування організацій.

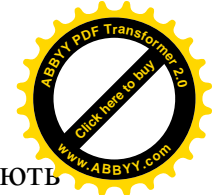
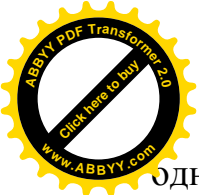
Інтелектуальні системи повинні в ідеалі мати здатність адаптуватися до такого роду змін. Причому у деяких випадках здатність адаптуватися до швидких змін є найбільш значимою властивістю інтелектуальних систем, оскільки характеристики, що регулюють певний ринок, можуть радикально змінитися за порівняно короткий термін.

Наприклад, відомий неоднозначний вплив збільшення відсоткових ставок на посилення або ослаблення валюти. Тому інтелектуальна система, що використовується для прогнозів на фінансових ринках, повинна вміти адаптуватися до складних трендів і давати успішні рекомендації, постійно навчаючись на досвіді ринку.

Слід зазначити, що адаптованість лежить в основі навчання інтелектуальних систем. Однак властивість навчання характеризує фазу підготовки системи до роботи адаптованість прийнято відносити до етапу функціонування попередньо навченої системи.

Під **гнучкістю** розуміється здатність інтелектуальної системи „бачити” образ криз шум і спотворення та здійснювати узагальнення на основі неповних, нечітких і неточних даних. Ця властивість інтелектуальних систем дозволяє перебороти вимогу суворості точності, пропоновану звичайним комп'ютером, і відкриває шлях до системи, що може мати справу з тим недосконалим світом, у якому ми живемо. Таким чином, окремі інтелектуальні системи, попередньо навчившись на частині інформації про деякий клас об'єктів, здатні „виводити” узагальнену модель, яку можна далі застосовувати для прогнозування поведінки інших об'єктів, що належать до цього класу.

Під **прозорістю тлумачення** поясненням розуміється здатність інтелектуальних систем надавати видобуті з даних знання в зрозумілому для експерта або ОПР вигляді. Інтелектуальні системи володіють потенційною можливістю автоматизувати процедуру прийняття рішень, але прийняті системою рішення повинні бути зрозумілі людині. Для досягнення необхідного рівня обґрунтованості управлінських рішень сама процедура ухвалення рішення повинна бути прозорою і відкритою для обговорення й аналізу. І якщо



Одні види інтелектуальних систем, наприклад, експертні системи, забезпечують прозорість тлумачення, то інші, наприклад, нейронні мережі не дають пояснення своїм рішенням.

Важливо також розуміти прийнятий інтелектуальною системою процес висування аргументів з метою підвищення надійності її роботи. Якщо інтелектуальна система перестає приймати правильні рішення, її можна „виправити” тільки в тому випадку, якщо процес аргументування рішення зрозумілий людині, тому що в цьому випадку легко встановити, що виявилось причиною такої „поведінки” системи. У випадках, коли приймаються інвестиційні рішення, пов'язані з величезними грошовими сумами або з заощадженнями клієнтів, ступінь аргументованості рішень відіграє надзвичайно велику роль.

Прозорість інтелектуальних систем важлива для забезпечення взаємодії між системою і людиною-експертом. Це є доказом того, що за певних обставин, наприклад, у разі непередбаченої конкуренції чи загостренні політичної ситуації, покращити результати роботи системи можуть так звані експертні ревізії моделей прийняття рішень. Такі експертні ревізії можна з відомими допущеннями трактувати як спільну „мозкову атаку” людей та інтелектуальної системи. Оскільки тільки людина здатна розуміти значення подій і оперативно зв'язуватися з зовнішнім світом, для підвищення ефективності рішень, прийнятих спільно експертом і системою, важливо, щоб система забезпечила доступ до свого знання, причому у формі, зрозумілій людині.

Під відкриттям нового розуміється здатність інтелектуальних систем виявляти раніше невідомі, приховані зв'язки і відношення у великих масивах числової і текстової інформації, прогнозувати появу нових бізнес-процесів. Технологія обробки даних Data mining є прикладом витягу раніше невідомої і потенційно корисної інформації з даних. Однак необхідно, як правило, проводити додаткові дослідження, щоб підтвердити, що виявлені відношення дійсно є значимими.

Крім перерахованих основних можливостей деякі інтелектуальні системи в окремих випадках можуть додатково мати такі **властивості**:

- універсальність;
- нелінійність;
- паралелізм;
- стійкість;
- креативність.

Під *універсальністю* розуміється здатність системи вирішувати широке коло задач (апроксимувати будь-яку функцію) і бути вільною від яких-небудь припущень щодо джерела даних. В основі універсальності лежить повнота функціонального базису, тобто можливість породжувати безліч функцій (будь-яку функцію).



Під *нелінійністю* розуміється здатність апроксимувати як завгодно складні нелінійні функції з будь-якою наперед заданою точністю. В основі нелінійності лежить властивість системи реалізовувати нелінійну непараметричну регресію.

Під *паралелізмом* розуміється рівнобіжна процедура обробки інформації. В основі паралелізму лежить відповідна архітектура процесорних елементів системи.

Під *стійкістю* розуміється здатність системи продовжувати виконання задачі, зберігаючи належну якість рішень, в умовах, коли ушкоджена частина її структури. В основі стійкості лежить розподілена пам'ять, тобто така властивість структури системи, при якій кожна „порція“ вхідних даних міститься в будь-якому елементі структури, і навпаки, кожен елемент структури зберігає інформацію про всі „порції“ вхідних даних.

Під *креативністю* розуміється здатність інтелектуальної системи породжувати нові (які не зустрічалися під час навчання) варіанти вирішення проблеми. Прозорість інтелектуальних систем важлива для забезпечення взаємодії між системою і людиною-експертом. Це є доказом того, що за певних обставин, наприклад, у разі непередбаченої конкуренції чи загостренні політичної ситуації, покращити результати роботи системи можуть так звані експертні ревізії моделей прийняття рішень.





## ЛІТЕРАТУРА

1. Матюшок В.М. Информатика для экономистов Учебник. – Москва, ИНФРА-М, 2006. – 880 с.
2. Экономическая информатика: учебник / Под ред. В.П. Косарева. – М.: Финансы и статистика, 2004.
3. Экономическая информатика. Учебник для вузов/ Под ред. проф. Евдокимова. СПб.: Питер, 1997.
4. Карлберг Конрад. Бизнес-анализ с помощью Excel / Пер. с англ. – К.: Диалектика, 2004. – 448 с.
5. Практикум по экономической информатике: Учебное пособие в 3-х ч. – М.: Перспектива, 2002.
6. Вейскас Д. Эффективная работа с Microsoft Access 7.0 для Windows 95 / Перев. с англ. – СПб.: Питер, 1997. – 848с.
7. Хэлворсон М, Янг М. Эффективная работа с Microsoft Office 97. – СПб.: Питер, 1997. – 1056 с.
8. Гетц К., Джилберт М. Программирование в Microsoft Office.. Руководство по Visual Basic for Applications // ВНУ, 2000, 384с.
9. Камлиш К. Internet для занятых /Перев. с англ. - СПб.: Питер, 1997. – 320 с.
- 10.Хоннекут Д. Использование Internet. 2-е Издание.: Пер. с англ. - К.: Диалектика, 1997, 300 с.
- 11.Годин В.В., Корнеев И.К. Управление информационными ресурсами. М.: Инфра-М, 2000. – 334 с.
- 12.Вовчак І.С. Інформаційні системи та комп'ютерні технології в менеджменті. – Тернопіль. Карт-бланк, 2001.
- 13.Титоренко Г.А. Информационные технологии управления. М.: Юнити, 2002.–280 с.
- 14.Батюк А.Є. та інш. Інформаційні системи в менеджменті. Львів: Інтеллект-Захід, 2004. – 520 с.
- 15.Гельман В.Я. Решение математических задач средствами Excel / Практикум. – СПб.: Питер, 2003. – 237с.
- 16.Смирнова Г.Н. и др. Проектирование экономических информационных систем: Учебник/ Под ред. Ю.Ф. Тельнова. – М.: Финансы и статистика, 2002. – 512 с.

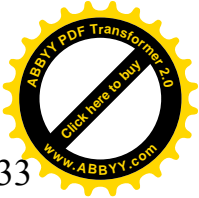


17. Устинова Г.М. Информационные системы менеджмента: Основные аналитические технологии в поддержке принятия решений / Учеб. пособие. - СПб: Издательство "ДиасофтЮП", 2000.
18. Павлыш В.Н., Анохина И.Ю., Кононенко И.Н., Зензеров В.И. Начальный курс информатики для пользователей персональных компьютеров. / Учебно-метод. пособие. – Донецк: ДонНТУ, 2006. – 235 с.
19. Солодова Л.А., Едемская Е.Н., Кляус П.Т., Кляус О.А. Информационные системы и технологии: Обработка экономических информационных массивов средствами СУБД ACCESS. / Методическое пособие. – Донецк: ДонНТУ, 2002. – 48 с.
20. Основи комп'ютерних інформаційних технологій. Операційні системи MS-DOS і WINDOWS- 98: Методичний посібник / Уклад. Л.А. Солодова, Є.М. Єдемська та інші. – Донецьк: ДонНТУ, 2002. – 60 с.

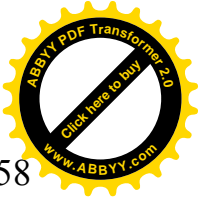
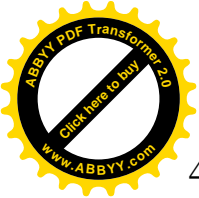


## ЗМІСТ

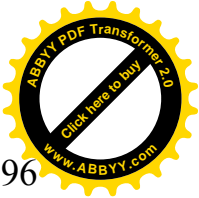
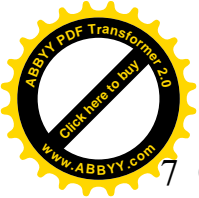
ВСТУП.....	3
1 ПРЕДМЕТ, МЕТОДИ І ЗАВДАННЯ ДИСЦИПЛІНИ.....	5
1.1 Інформація і керування.....	7
1.2 Інформаційні технології.....	8
1.3 Приклади інформаційних технологій і процедур обробки економічної інформації.....	11
2 ТЕОРЕТИЧНІ ОСНОВИ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ.....	13
2.1 Загальні відомості про персональний комп'ютер.....	13
2.1.1 Системний блок.....	13
2.1.2 Пам'ять.....	14
2.1.3 Гнучкі магнітні диски.....	14
2.1.4 Жорсткий диск (Вінчестер).....	14
2.1.5 Дисплей (Монітор).....	15
2.1.6 Клавіатура.....	15
2.1.7 Приводи CD-ROM.....	15
2.1.8 Принтер.....	15
2.1.9 Плоттер.....	15
2.1.10 Миша.....	15
2.1.11 Сканер.....	15
2.1.12 Модем.....	15
2.2 Загальна характеристика програмного забезпечення.....	16
2.3 Представлення інформації в ЕОМ.....	18
2.3.1 Переведення числа з однієї системи числення в іншу.....	19
2.3.2 Запис нечислової інформації в двійковій системі числення.....	22
3 СИСТЕМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ.....	24
3.1 Файли.....	25
3.2 Папки.....	25
3.3 Запуск WINDOWS – XP.....	26
3.4 Робота з мишею.....	27
3.5 Основні елементи "Панели задач".....	28
3.6 Головне меню.....	28
3.7 Вихід з системи WINDOWS – XP.....	29
3.8 Відкриття папки.....	30
3.9 Папка "Мой комп'ютер".....	30
3.10 Папка "Корзина".....	30
3.11 Ярлики.....	31
3.12 Меню об'єкта.....	31
3.13 Робота з вікнами.....	31
3.13.1 Елементи вікна.....	31
3.13.2 Типи вікон.....	33
3.13.3 Зміна розмірів вікон і переміщення вікон.....	33



3.13.4	Перемикання між вікнами.....	33
3.13.5	Упорядкування вікон на екрані.....	34
3.13.6	Системне меню й рядок меню.....	34
3.13.7	Пункт "Вид" рядка меню вікна папки.....	36
3.13.8	Робота з діалоговими вікнами.....	37
3.14	Робота з папками й файлами.....	38
3.14.1	Вибір папок, файлів і ярликів у вікні папки (вибір об'єктів).....	38
3.14.2	Властивості папок і файлів.....	39
3.14.3	Створення нових папок.....	39
3.14.4	Переміщення й копіювання папок, файлів і ярликів (об'єктів).....	39
3.14.5	Перейменування папок, файлів і ярликів (об'єктів).....	41
3.14.6	Видалення папок, файлів і ярликів (об'єктів).....	41
3.14.7	Відновлення видалених файлів.....	41
3.14.8	Очищення папки "Корзина".....	41
3.14.9	Атрибути папок, файлів (об'єктів).....	41
3.14.10	"Проводник WINDOWS".....	42
3.14.11	Запуск програм на виконання.....	44
3.14.12	Створення ярликів.....	44
3.15	Форматування дисків.....	45
3.16	Обмін даними.....	46
3.16.1	Обмін даними за допомогою "Буфера обмена".....	46
3.16.2	Впровадження об'єкта.....	46
3.16.3	Створення зв'язку.....	47
3.17	Настроювання елементів системи.....	47
3.17.1	Настройка "Панели задач".....	47
3.17.2	Підключення індикатора годинника.....	48
3.17.3	Підключення індикатора мови.....	48
3.17.4	Настроювання "Главного меню".....	49
3.17.5	Очищення пункту "Документи" "Главного меню".....	49
3.17.6	Підключення багатомовної підтримки.....	49
3.17.7	Підключення мови.....	49
3.17.8	Видалення мови з списку підключених мов.....	50
3.17.9	Установка принтера.....	50
3.17.10	Настроювання параметрів принтера.....	50
3.17.11	Настроювання екрана.....	52
3.17.12	Настроювання форми дати, часу, чисел і грошових одиниць.....	52
4	МЕРЕЖНІ ТЕХНОЛОГІЇ.....	53
4.1	Еволюція комп'ютерної технології.....	53
4.2	Особливості локальних, глобальних, міських та корпоративних мереж.....	55
4.3	Вимоги до сучасних комп'ютерних мереж.....	55
4.4	Основні програмні та апаратні компоненти мережі.....	56
4.5	Лінії зв'язку.....	57
4.6	Загальні принципи побудови комп'ютерної мережі.....	58

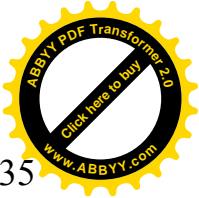


4.7	Стандарти створення комп'ютерних мереж.....	58
4.7.1	Модель OSI.....	59
4.7.2	Технологія Ethernet.....	59
4.7.2.1	Комунікаційне обладнання Ethernet.....	60
4.7.3	Подальший розвиток технології Ethernet.....	62
4.8	Загальні відомості про глобальну комп'ютерну мережу Internet.....	62
4.8.1	Основні послуги, які надає користувачам Internet.....	63
4.8.1.1	Передача файлів за протоколом FTP.....	64
4.8.1.2	Доступ до ресурсів хост-комп'ютерів за допомогою Telnet.....	64
4.8.1.3	Електронна пошта.....	65
4.8.1.4	Телеконференції UseNet.....	65
4.8.1.5	Конференції Internet Relay Chat.....	66
4.8.1.6	World Wide Web (WWW).....	66
5	ЗАСТОСУВАННЯ ІНТЕРНЕТУ В ЕКОНОМІЦІ.....	67
5.1	Сектори електронної комерції.....	68
5.2	Основні види бізнесу в ІНТЕРНЕТІ.....	73
5.2.1	Електронна вітрина.....	73
5.2.2	Електронний магазин.....	73
5.2.3	Електронний аукціон.....	74
5.3	Електронні платіжні системи.....	75
5.3.1	Оплата пластиковою картою.....	75
5.3.2	Дебетові схеми (некарткові платіжні системи).....	77
5.3.3	Електронні гроші і смарт-карти.....	77
6	ОРГАНІЗАЦІЯ КОМП'ЮТЕРНОЇ БЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ.....	79
6.1	Інформаційна політика.....	79
6.2	Політика безпеки.....	80
6.3	Сервіси безпеки та механізми її порушень.....	82
6.4	Шифрування даних.....	85
6.4.1	Симетрична криптографія.....	86
6.4.2	Асиметрична криптографія.....	87
6.4.3	Цифровий підпис.....	88
6.5	Комп'ютерні віруси.....	90
6.5.1	Зародження комп'ютерних вірусів.....	91
6.5.2	Шляхи проникнення вірусів у комп'ютер і механізм розподілу вірусних програм.....	91
6.5.3	Ознаки появи вірусів.....	92
6.5.4	Методи боротьби з вірусами.....	92
6.6	Класифікація антивірусів.....	93
6.6.1	Сканери.....	93
6.6.2	CRC-сканери.....	94
6.6.3	Блокуючі антивіруси.....	94
6.6.4	Імунізатори.....	95

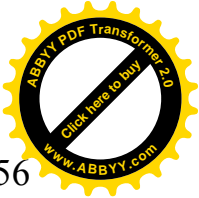


7	ОСНОВИ ВЕБ-ДИЗАЙНУ.....	96
7.1	Коротка характеристика мови HTML.....	96
7.2	Визначення загальних параметрів.....	98
7.3	Форматування тексту.....	100
7.4	Використання тегів заголовків.....	104
7.5	Форматування HTML-документу.....	106
7.6	Використання коментарів та спеціальних символів.....	108
7.7	Рисунки.....	109
7.8	Звук.....	113
7.9	Гіперпосилання.....	114
7.10	Списки.....	115
7.10.1	Маркіровані списки.....	115
7.10.2	Нумеровані списки.....	117
7.10.3	Списки визначень.....	117
7.11	Таблиці.....	119
7.12	Фрейми.....	120
8	ПРОГРАМНІ ЗАСОБИ РОБОТИ ЗІ СТРУКТУРОВАНИМИ ДОКУМЕНТАМИ.....	123
8.1	Текстовий редактор WORD.....	123
8.1.1	Завантаження редактора Word.....	123
8.1.2	Збереження документа.....	123
8.1.3	Збереження файлу з іншим ім'ям або в іншій папці.....	123
8.1.4	Відкриття документа.....	124
8.1.5	Настроювання параметрів сторінки.....	124
8.1.6	Друкування файлу.....	124
8.1.7	Режим попереднього перегляду.....	125
8.1.8	Робота з панелями інструментів.....	125
8.1.8.1	Підключення окремих кнопок до панелі інструментів.....	125
8.1.8.2	Видалення кнопки з панелі інструментів.....	126
8.1.9	Виділення тексту.....	126
8.1.10	Копіювання фрагмента.....	127
8.1.11	Переміщення фрагмента.....	127
8.1.12	Видалення фрагмента.....	127
8.1.13	Форматування тексту (символів).....	127
8.1.14	Створення нестандартного підкреслення символів.....	128
8.1.15	Вставка спеціальних символів.....	128
8.1.16	Вставка нижніх і верхніх індексів.....	128
8.1.17	Форматування абзаців.....	129
8.1.18	Створення рамки навколо фрагмента тексту.....	130
8.1.19	Створення газетних стовпчиків.....	130
8.1.20	Створення списків.....	131
8.1.21	Створення багаторівневого списку.....	133
8.1.22	Нумерація сторінок.....	134
8.1.23	Видалення номерів сторінок.....	134





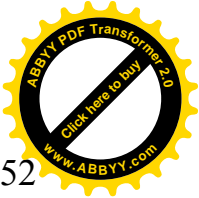
8.1.24	Установка переносу слів.....	135
8.1.25	Робота з таблицями.....	135
8.1.25.1	Створення таблиці.....	135
8.1.25.2	Розміри комірок.....	135
8.1.25.3	Виділення рядків, стовпців і комірок.....	136
8.1.25.4	Зміна орієнтації тексту в комірці.....	136
8.1.25.5	Заливання комірок кольором.....	137
8.1.25.6	Зміна виду границь комірок.....	137
8.1.25.7	Об'єднання комірок таблиці.....	138
8.1.25.8	Розбивка комірок таблиці.....	138
8.1.25.9	Видалення рядків, стовпців, комірок.....	139
8.1.25.10	Вставка рядка в будь-якому місці таблиці.....	139
8.1.25.11	Додавання рядка в кінець таблиці.....	139
8.1.25.12	Додавання стовпця в таблицю.....	139
8.1.25.13	Додавання стовпця в правий край таблиці.....	139
8.1.25.14	Переміщення і копіювання рядків, стовпців і комірок.....	139
8.1.25.15	Сортування даних у таблиці.....	140
8.1.26	Вставка об'єкта WordArt.....	141
8.1.27	Вставка готових малюнків.....	142
8.1.28	Зміна розмірів графічних об'єктів.....	142
8.1.29	Переміщення графічних об'єктів.....	142
8.1.30	Зміна параметрів графічних об'єктів.....	142
8.1.31	Обрізка малюнка.....	142
8.1.32	Обтікання малюнка текстом.....	143
8.1.33	Малювання об'єктів.....	144
8.1.34	Групування фігур.....	145
8.1.35	Напис усередині фігури.....	146
8.1.36	Побудова формул.....	146
8.1.37	Редагування формул.....	148
8.2	Електронні таблиці EXCEL.....	149
8.2.1	Завантаження електронних таблиць EXCEL.....	149
8.2.2	Вікно робочої книги.....	149
8.2.3	Перейменування листа.....	150
8.2.4	Вставка нового листа.....	150
8.2.5	Видалення листа.....	150
8.2.6	Переміщення листа.....	150
8.2.7	Переміщення по листах книги.....	151
8.2.8	Переміщення по листу книги.....	151
8.2.9	Рядок формул.....	151
8.2.10	Виділення комірок.....	152
8.2.11	Форматування комірок.....	153
8.2.11.1	Вкладка "ЧИСЛО".....	153
8.2.11.2	Вкладка "ВЫРАВНИВАНИЕ".....	153
8.2.11.3	Вкладка "ШРИФТ".....	155
8.2.11.4	Вкладка "ГРАНИЦА".....	155



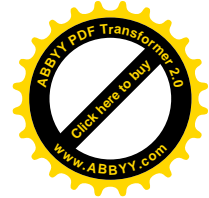
8.2.11.5 Вкладка "ВИД" .....	156
8.2.12 Зміна ширини стовпців .....	157
8.2.13 Зміна висоти рядків.....	157
8.2.14 Вставка стовпців (рядків).....	157
8.2.15 Видалення рядків, стовпців.....	157
8.2.16 Очищення комірок.....	157
8.2.17 Копіювання даних за допомогою Буфера обміну.....	157
8.2.18 Введення даних.....	158
8.2.18.1 Введення чисел.....	158
8.2.18.2 Введення довгого тексту.....	158
8.2.18.3 Введення дати і часу.....	158
8.2.19 Редагування даних.....	159
8.2.20 Сортування даних.....	159
8.2.21 Введення формул.....	160
8.2.22 Відображення формул в комірках листа.....	161
8.2.23 Відносні, абсолютні і змішані посилання.....	161
8.2.23.1 Зміна типу посилання.....	161
8.2.23.2 Різниця абсолютного і відносного посилань.....	162
8.2.24 Копіювання формули.....	162
8.2.25 Введення даних за допомогою списків.....	163
8.2.26 Введення даних за допомогою рядів даних.....	163
8.2.27 Посилання на комірки іншого листа тієї ж книги.....	163
8.2.28 Копіювання таблиці на інший лист за допомогою посилань на комірки.....	164
8.2.29 Копіювання таблиці за допомогою команди "Спеціальна вставка".....	164
8.2.30 Використовування Майстра функцій.....	165
8.2.31 Автопідсумовування.....	167
8.2.32 Математичні функції.....	167
8.2.33 Статистичні функції.....	168
8.2.34 Логічні функції.....	169
8.2.35 Діаграми.....	170
8.2.35.1 Основні елементи діаграм.....	170
8.2.35.1 Побудова діаграм.....	170
8.2.35.3 Коректування елементів діаграм.....	176
8.2.36 Додавання лінії тренда.....	177
8.2.37 Прогноз даних за допомогою лінії тренда.....	178
8.2.38 Формули масивів.....	179
8.2.39 Функції роботи з матрицями.....	180
8.2.40 Розв'язування системи лінійних алгебраїчних рівнянь(СЛАУ).....	180
8.2.40.1 Метод зворотної матриці.....	181
8.2.40.2 Метод Крамера.....	181
8.2.41 Макроси.....	184
8.2.42 Рішення задач оптимізації.....	185



8.2.43 Використовування функції ПРОСМОТР для зв'язку двох таблиць.....	191
8.2.44 Операції з датами.....	192
8.2.45 Функції категорії "Дата и время".....	193
8.2.46 Використовування фільтрів.....	194
8.2.46.1 Автофільтр.....	194
8.2.46.2 Розширений фільтр.....	197
8.2.47 Розрахунок проміжних підсумків в таблиці.....	201
8.2.48 Створення зведених таблиць.....	206
8.2.49 Перестановка полів зведеної таблиці.....	209
8.2.50 Зміна функції в зведеній таблиці.....	210
8.2.51 Оновлення даних в зведеній таблиці.....	211
<b>9 ПРОГРАМНІ ЗАСОБИ РОБОТИ З БАЗАМИ ТА СХОВИЩАМИ ДАНИХ.....</b>	<b>214</b>
9.1 Використання СУБД ACCESS для обробки даних.....	214
9.1.1 Правила нормалізації таблиці.....	214
9.1.2 Типи даних у СУБД ACCESS.....	216
9.1.3 Зв'язки таблиць БД.....	218
9.2 Створення бази даних.....	218
9.3 Створення таблиць в базі даних.....	218
9.3.1 Створення структури таблиці.....	219
9.3.2 Використання Майстра Підстановок.....	219
9.3.3 Установка і видалення зв'язку таблиць бази даних.....	221
9.3.4 Введення даних у таблицю.....	223
9.4 Використання форм.....	223
9.4.1 Створення простої форми за допомогою майстра.....	223
9.4.2 Створення підпорядкованих форм.....	224
9.5 Вибірка даних за запитом.....	228
9.5.1 Запити на вибірку.....	228
9.5.2 Створення запиту з полем, що обчислюється.....	231
9.5.3 Сортування даних у запиті.....	233
9.5.4 Вибірка даних за умовою.....	234
9.5.5 Параметричний запит.....	236
9.5.6 Підсумковий запит.....	238
9.6 Створення простих звітів.....	240
9.7 Вбудовані функції.....	243
<b>10 ОСНОВИ ОФІСНОГО ПРОГРАМУВАННЯ.....</b>	<b>245</b>
10.1 Інтегроване середовище розробки додатків на мові VBA.....	245
10.2 Особливості програмування на мові VBA.....	248
10.2.1 Об'єкти, властивості і методи VBA.....	248
10.2.2 Опис даних.....	250
10.2.2.1 Опис простих змінних.....	251
10.2.2.2 Опис констант.....	251
10.2.2.3 Опис масивів.....	251



10.2.3	Вирази.....	252
10.2.3.1	Види операцій.....	252
10.2.3.2	Пріоритет виконання операцій.....	252
10.3	Основні оператори мови VBA.....	255
10.3.1	Правила запису операторів.....	255
10.3.2	Оператор присвоювання.....	255
10.3.3	Оператори вводу-виводу.....	255
10.3.3.1	Оператор і функція MsgBox .....	255
10.3.3.2	Функція InputBox.....	256
10.3.4	Умовний оператор IF.....	256
10.3.5	Оператори циклу.....	258
10.4	Структура програми. Модулі, процедури та функції.....	260
10.5	Створення екранних форм і включення їх в проекти.....	264
10.5.1	Створення форм. Властивості, події і методи форм.....	264
10.5.2	Вибір і використання елементів, що управляють.....	267
11	ЕКСПЕРТНІ І НАВЧАЛЬНІ СИСТЕМИ.....	274
11.1	Приклади експертних систем і систем підтримки прийняття рішень.....	278
12	ПЕРСПЕКТИВИ РОЗВИТКУ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ.....	280
12.1	Сучасний стан інформаційних технологій.....	280
12.2	Тенденції розвитку інформаційних технологій.....	280
12.3	Системи штучного інтелекту.....	281
	ЛІТЕРАТУРА.....	286



## **КУРС ЛЕКЦІЙ З ДИСЦИПЛІНИ “ІНФОРМАТИКА”**

(для студентів економічних спеціальностей).

Укладачі:

Бельков Дмитро Валерійович, к.т.н., доц.

Єдемська Євгенія Миколаївна, ст. викладач