



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ
ТЕХНІЧНИЙ УНІВЕРСИТЕТ

"ОСОБЛИВОСТІ ОБРОБКИ ТАБЛИЧНИХ ДАНИХ EXCEL В IDE НА МОВІ VBA"

Методичні вказівки і завдання

(для студентів всіх спеціальностей)



Донецьк 2008



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**"ОСОБЛИВОСТІ ОБРОБКИ ТАБЛИЧНИХ ДАНИХ
EXCEL В IDE НА МОВІ VBA"**

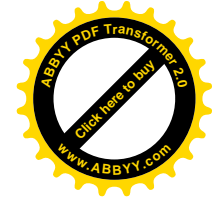
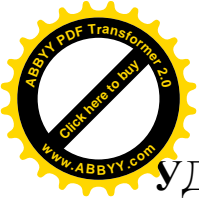
Методичні вказівки і завдання

(для студентів всіх спеціальностей)

*Розглянуто на засіданні кафедри
обчислювальної математики
і програмування.
Протокол № 4 від 17 листопада 2008 р.*

*Затверджено на навчально-видавничій
раді ДонНТУ.
Протокол №6 від 15 грудня 2008 р.*

Донецьк ДонНТУ 2008



УДК 681.3.06(071)

О 23

"Особливості обробки табличних даних EXCEL в IDE на мові VBA":

Методичні вказівки і завдання / Автори: Зензеров В.І., Славінська Л.В., Єдемська Є.М. – Донецьк: ДонНТУ, 2008. – 84 с.

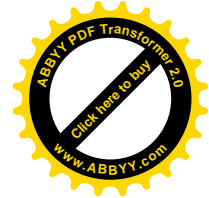
Методичні вказівки присвячені питанням розробки програмних модулів в середовищі Visual Basic for Application для обробки інформації представленої в табличній формі в MS Excel. Містять приклад і варіанти завдань для виконання лабораторної роботи, що сприяє практичному засвоєнню матеріалу.

Матеріал корисний для студентів всіх спеціальностей, що вивчають курс "Інформатика і КТ", а також для викладачів, аспірантів і інших користувачів персональних комп'ютерів.

Автори:

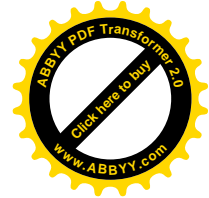
Зензеров В.І., ст. викладач
Славінська Л.В., ст. викладач
Єдемська Є.М., ст. викладач

Відповідальний за випуск: д.т.н., проф. Павлиш В.М.



ЗМІСТ

Вступ.....	4
1. Інтегроване середовище розробки додатків на мові VBA.....	5
2. Особливості програмування на мові VBA.....	9
2.1. Об'єкти, властивості і методи VBA.....	9
2.2. Опис даних.....	12
2.3. Вирази.....	14
3. Основні оператори мови VBA.....	19
3.1. Правила запису операторів.....	19
3.2. Оператор присвоювання.....	19
3.3. Оператори вводу-виводу.....	19
3.4. Умовний оператор IF.....	21
3.5. Оператори циклу.....	23
4. Структура програми. Модулі, процедури і функції.....	25
5. Створення екранних форм і включення їх в проекти.....	31
5.1. Створення форм. Властивості, події і методи форм.....	31
5.2. Вибір і використання елементів, що управляють.....	34
6. Приклад обробки табличних даних в MS EXCEL з використанням VBA.....	42
7. Варіанти завдання до лабораторної роботи.....	52
Список літератури.....	83



ВСТУП

Мрія користувачів персонального комп'ютера мати універсальний, потужний, гнучкий і зручний засіб, що дозволяє швидко і ефективно вирішувати найрізноманітніші економічні і фінансові завдання, втілилася в MS EXCEL. Visual Basic for Application (VBA) суттєво збагатив і посилив MS EXCEL. VBA – це поєднання одного з найпростіших мов програмування і всіх обчислювальних можливостей MS EXCEL. За допомогою VBA ви зможете легко і швидко створювати всілякі додатки, навіть не будучи фахівцем в області програмування. VBA містить графічне середовище, що дозволяє наглядно конструювати екранні форми з елементів, що управляють. VBA у поєднанні з можливостями MS EXCEL дозволяє вирішувати задачі, про рішення яких лише засобами MS EXCEL раніше навіть і не говорили.

Даний методичний посібник допоможе вам грамотно і ефективно використовувати комп'ютер в своїй роботі, вивчити особливості програмування на мові Visual Basic for Application для обробки даних, представлених у вигляді таблиць в MS EXCEL, і, таким чином, розширити можливості стандартних застосувань (зокрема, популярного пакету MS Office). Розширення можливостей стандартних компонентів пакету Microsoft Office для вирішення конкретних задач досягається створенням власних додатків шляхом програмування на мові VBA, яка доступна будь-якому кваліфікованому користувачеві персональних комп'ютерів.

Викладений в посібнику методичний матеріал із прикладами і приведені варіанти завдань передбачають, що користувач знайомий з основами програмування в середовищі VBA і бажає розширити свої знання в області використання VBA для рішення економічних задач.

1. ІНТЕГРОВАНЕ СЕРЕДОВИЩЕ РОЗРОБКИ ДОДАТКІВ НА МОБІ 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. 1.1. 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.

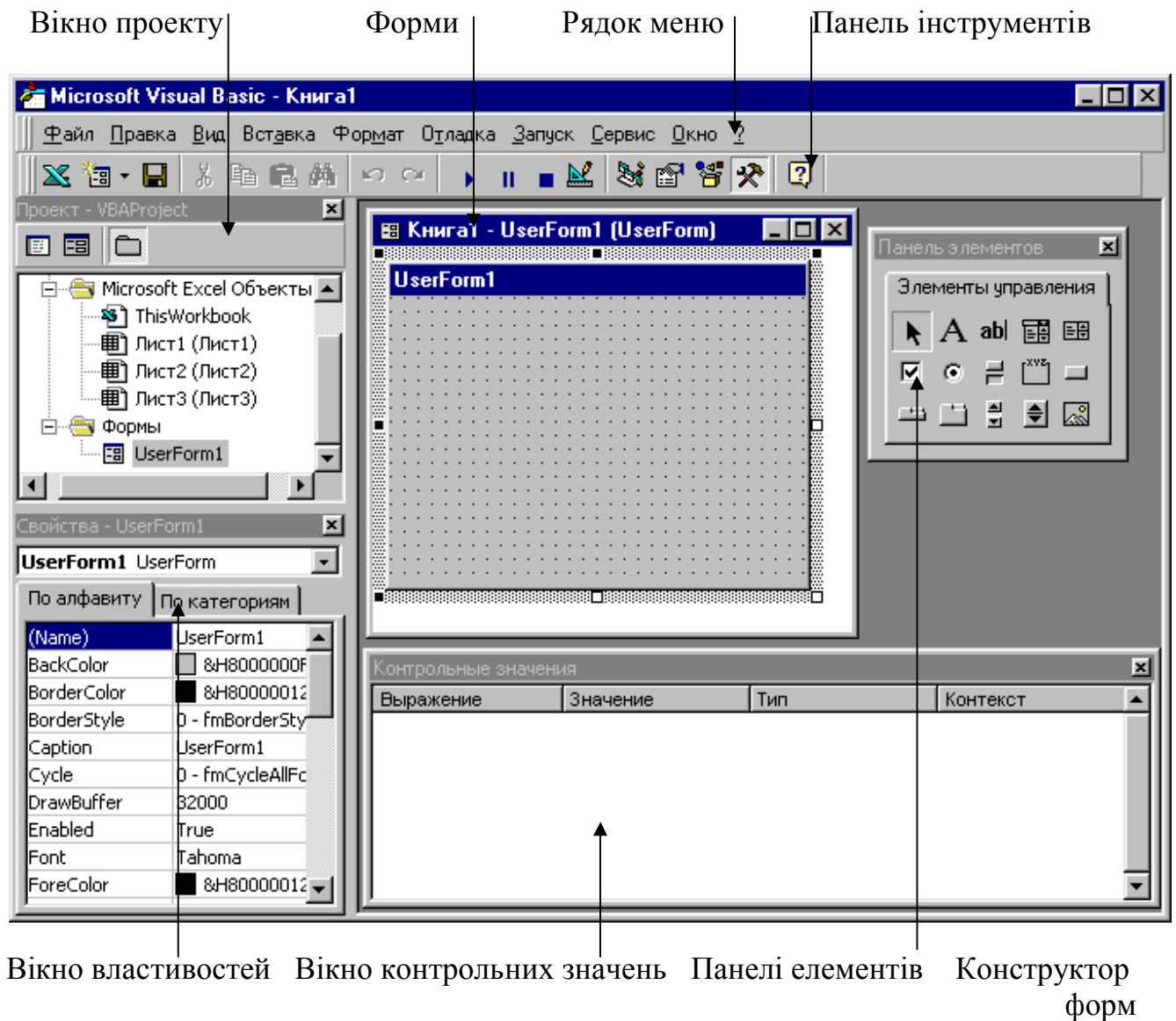


Рис. 1.1. Загальний вигляд IDE VBA



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 ній 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.

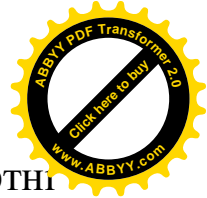
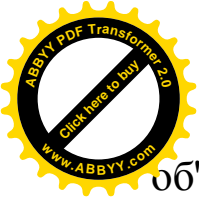
Меню **"Сервіс"** дозволяє включити додаткові елементи, запустити макроси і настроїти параметри редактора.

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. 1.2). Якщо вона відсутня, необхідно виконати команду **"Вид" → "Панели инструментов" → "Стандарт"**.

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. 1.3). Це стандартні об'єкти відкритого додатка Microsoft Office (документ редактора MS Word, книга і її листи MS Excel), форми, модулі і класи.

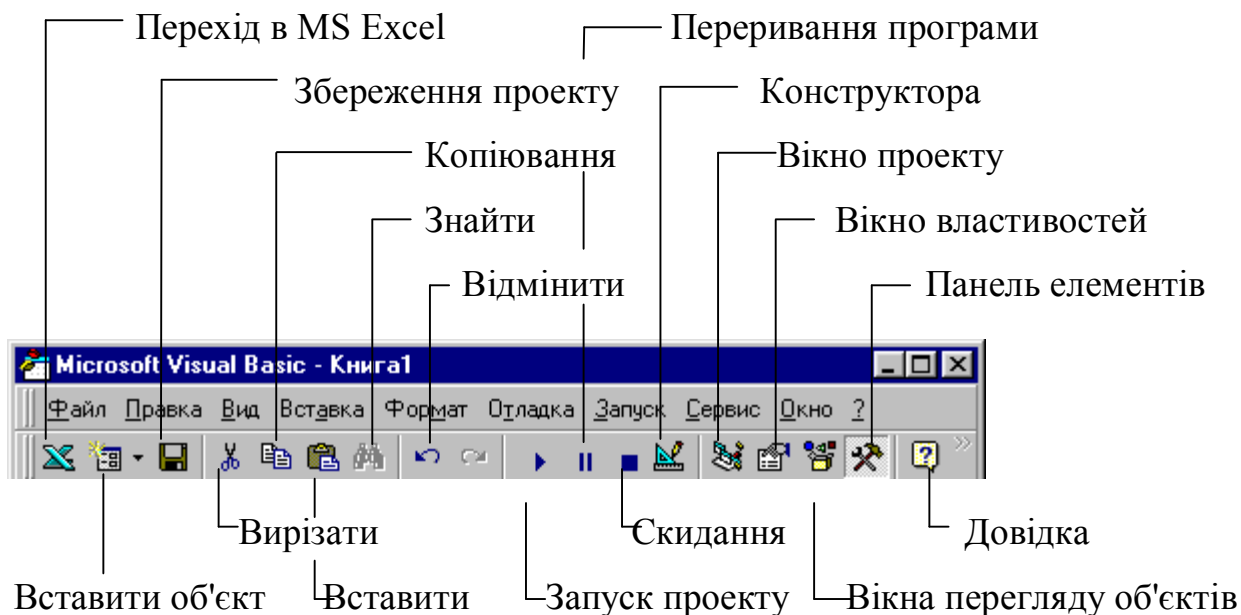


Рис. 1.2. Панель інструментів VBA

Вікно властивостей відображує різні атрибути виділеного об'єкту (рис. 1.4). 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.

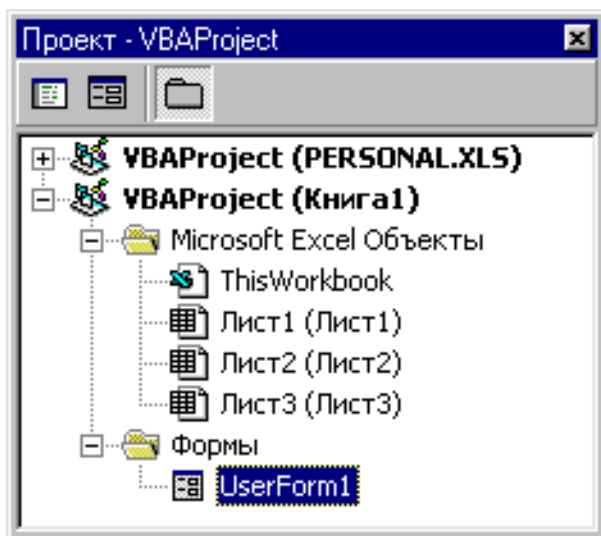


Рис. 1.3. Вікно проекту

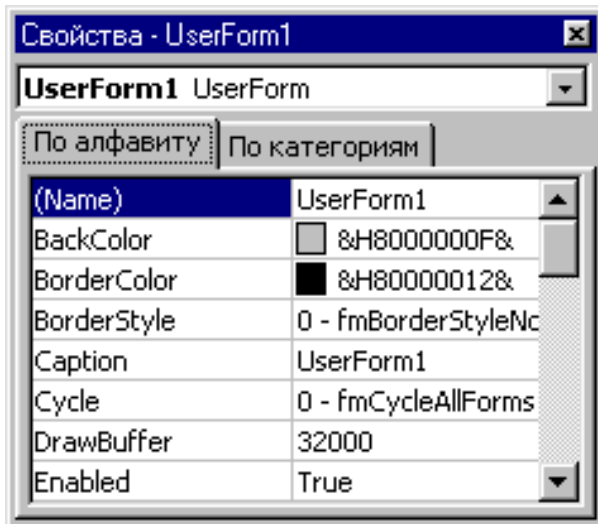
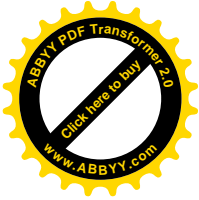
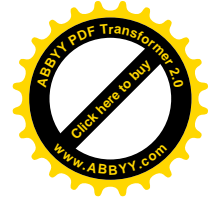


Рис. 1.4. Вікно властивостей


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).



2. ОСОБЛИВОСТІ ПРОГРАМУВАННЯ НА МОВІ VBA

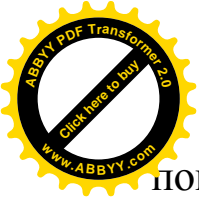
Процес розробки програми на мові VBA – **проекту**, може складатися з декількох етапів, залежно від кінцевого результату. Якщо необхідно отримати програму, яка проводитиме певні обчислення або дії, що розширюють математичні можливості стандартного застосування Microsoft Office, то досить створити **програмний 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".

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:

- | | |
|----------------------------|--|
| Range (“Адреса”) | - діапазон комірок (може включати лише одну комірку). |
| Cells (i, j) | - комірка, що знаходиться на пересіченні і-й рядка і j-го стовпця робочого аркуша MS Excel (i і j – цілі числа). |
| Rows (№ рядки) | - рядок із заданим номером. |
| Columns (№ стовпця) | - стовпець із заданим номером |
| Sheets (“Ім'я”) | - аркуш з вказаним ім'ям. |
| Sheets (№ аркуша) | - аркуш з вказаним номером. |
| WorkSheet | - 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 комірок 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:

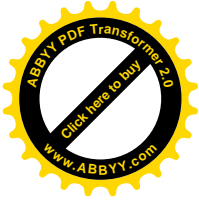
Змінна = Об'єкт. Властивість

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:

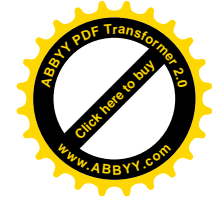
ActiveCell.Value = "Да".

2.2. Опис даних

Всі об'єкти, якими оперує мова програмування VBA, відносяться до певного типу.

Тип даних визначає:

- область можливих значень змінної;

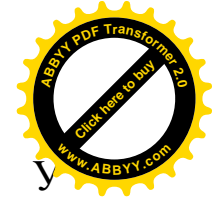


- структуру організації даних;
- операції, визначені над даними цього типу.

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 2.1.

Таблиця 2.1. Скалярні типи VBA

Ім'я типа	Українська назва типа	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**.

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.

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*

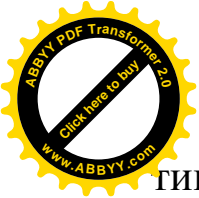
Наприклад.

Const Pi As Double = 3.141593

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



типа. 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 - одомірний масив, що складається з 5 елементів цілого типа, Y - двомірний масив, в якого 10 рядків і 20 стовпців з елементами числового типа подвійної точності.

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.

2.3.1. Види операцій

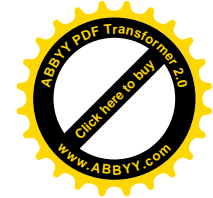
Операції бувають арифметичні, відношення і логічні:

– **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").

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 2.2.

Таблиця 2.2. 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 2.3.

Таблиця 2.3. 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 2.4.

Таблиця 2.4. 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.
$\text{Sign } X$	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.
$\text{tg } X$	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:

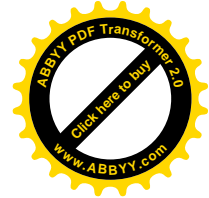
$$X \geq 1 \text{ And } X \leq 5$$

Результатом виразу буде *True*.

Щоб отримати перелік всіх математичних функцій, досить набрати ім'я будь-якої відомої математичної функції (наприклад, *SIN*), а потім натискувати клавішу **F1** і нижче за опис вибраної функції вибрати посилання на **"Математические функции"**. У отриманому переліку можна отримати довідку про призначення будь-якій зі вбудованих математичних функцій і її аргументі.

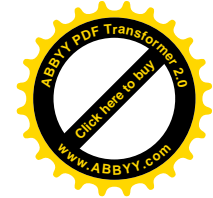
Щоб отримати перелік всіх похідних математичних функцій і правила їх формування, досить набрати ім'я будь-якої відомої математичної функції (наприклад, *SIN*), а потім натискувати клавішу **F1** і нижче за опис вибраної функції вибрати посилання на **"Производные математические функции"**.

Нижче, в таблиці 2.5, приведений список функцій, яки можуть бути побудовані за допомогою вбудованих математичних функцій.



Таблиця 2.5. Похідні математичні функції

Математичний запис	Назва функції	Комбінація вбудованих функцій
$\sec X$	Секанс	$1/\cos(X)$
$\csc X$	Косеканс	$1/\sin(X)$
$\operatorname{ctg} X$	Котангенс	$1/\tan(X)$
$\arcsin X$	Арксинус	$\operatorname{Atn}(X/\sqrt{-X^2+1})$
$\arccos X$	Арккосинус	$\operatorname{Atn}(-X/\sqrt{-X^2+1})+2*\operatorname{Atn}(1)$
$\operatorname{arcsec} X$	Арксеканс	$\operatorname{Atn}(X/\sqrt{X^2-1})+\operatorname{Sgn}(X-1)*2*\operatorname{Atn}(1)$
$\operatorname{arccsc} X$	Арккосеканс	$\operatorname{Atn}(X/\sqrt{X^2-1})+(\operatorname{Sgn}(X)-1)*2*\operatorname{Atn}(1)$
$\operatorname{arcctg} X$	Арккотангенс	$\operatorname{Atn}(X)+2*\operatorname{Atn}(1)$
$\operatorname{sh} X$	Гіперболічний синус	$(\exp(X) - \exp(-X))/2$
$\operatorname{ch} X$	Гіперболічний косинус	$(\exp(X) + \exp(-X))/2$
$\operatorname{th} X$	Гіперболічний тангенс	$(\exp(X) - \exp(-X))/(\exp(X) + \exp(-X))$
$\operatorname{sch} X$	Гіперболічний секанс	$2/(\exp(X) + \exp(-X))$
$\operatorname{csch} X$	Гіперболічний косеканс	$2/(\exp(X) - \exp(-X))$
$\operatorname{cth} X$	Гіперболічний котангенс	$(\exp(X) + \exp(-X))/(\exp(X) - \exp(-X))$
$\operatorname{arsh} X$	Гіперболічний арксинус	$\operatorname{Log}(X + \sqrt{X^2+1})$
$\operatorname{arch} X$	Гіперболічний арккосинус	$\operatorname{Log}(X + \sqrt{X^2-1})$
$\operatorname{arth} X$	Гіперболічний арктангенс	$\operatorname{Log}((1+X)/(1-X)) / 2$
$\operatorname{arsch} X$	Гіперболічний арксеканс	$\operatorname{Log}((\sqrt{-X^2+1}+1)/X)$
$\operatorname{arsch} X$	Гіперболічний арккосеканс	$\operatorname{Log}((\operatorname{Sgn}(X)*\sqrt{X^2+1}+1)/X)$
$\operatorname{arcth} X$	Гіперболічний арккотангенс	$\operatorname{Log}((X+1)/(X-1)) / 2$
$\log_n X$	Логарифм по підставі N	$\operatorname{Log}(X) / \operatorname{Log}(N)$



3. ОСНОВНІ ОПЕРАТОРИ МОВИ VBA

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.

3.2. Оператор присвоювання

Оператор присвоювання має наступний синтаксис:

ІМ'Я_ЗМІННОЇ = ВИРАЗ

3.3. Оператори вводу-виводу

3.3.1. Оператор і функція MsgBox

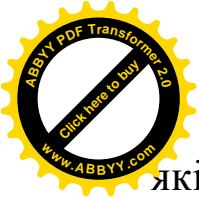
Оператор **MsgBox** здійснює виведення інформації в діалоговому вікні і встановлює режим чекання натиснення кнопки користувачем. Він має наступний синтаксис:

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. 3.1 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 3.1. 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 "Прекратить", "Повторить" та "Игнорировать"	VbAbortRetryIgnore
Кнопки "Повторить" та "Отмена"	VbRetryCancel
Інформаційний знак	VbInformation
Знак 	VbCritical
Знак питання	VbQuestion
Знак оклику	VbExclamation

3.3.2. Функція InputBox

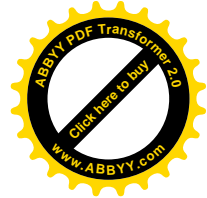
Функція **InputBox** здійснює введення значень змінних за допомогою вікна введення і має наступний синтаксис:

$ІМ'Я_ЗМІННОЇ = \text{InputBox}(ПОВІДОМЛЕННЯ[, ЗАГОЛОВОК])$

Аргументи:

ПОВІДОМЛЕННЯ - обов'язковий аргумент. Задає у вікні інформаційне повідомлення, що зазвичай пояснює сенс величини, що вводиться

ЗАГОЛОВОК - задає заголовок вікна.



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

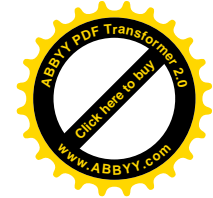
ОПЕРАТОР_1 виконується, якщо УМОВА істинна, інакше виконується ОПЕРАТОР_2. При цьому оператор **If...Then...Else** записується в один рядок.

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. 3.2.

Таблиця 3.2. 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. 3.3.



Таблиця 3.3. Логічні операції

	Назва	Вираз	A	B	Результат
Not	Логічне заперечення	Not A	False True		True False
And	Логічне "И"	A And B	True True False False	True False True False	True False False False
Or	Логічне "ИЛИ"	A Or B	True True False False	True False True False	True True True False

У умовному операторові 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 операторові 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
```



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:

цикли з передумовою	–	Do While ... Loop, Do Until ... Loop;
цикли з постумовою	–	Do ... Loop While, Do ... Loop Until.

* **Примітка.** Не рекомендується примусово змінювати значення параметра циклу, його початкового і кінцевого значення в тілі циклу **For...Next**.



Нижче приведені 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.

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. 4.1.

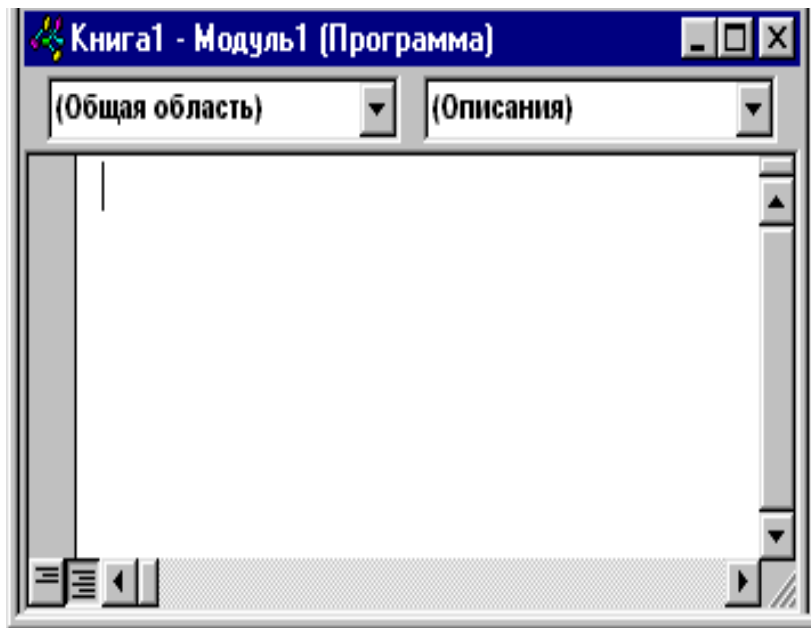


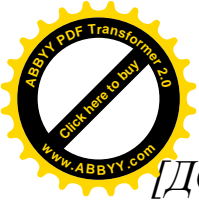
Рис. 4.1. Вікно модуля

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])
ТИЛО_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 присвоєння, 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. 4.2), і введемо в нього текст наступної програми:

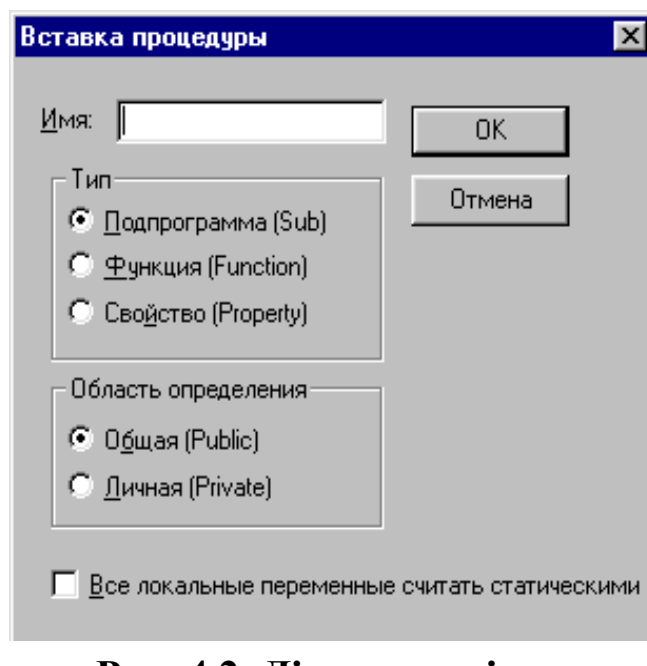


Рис. 4.2. Діалогове вікно
“Вставка процедури”

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. 4.3).

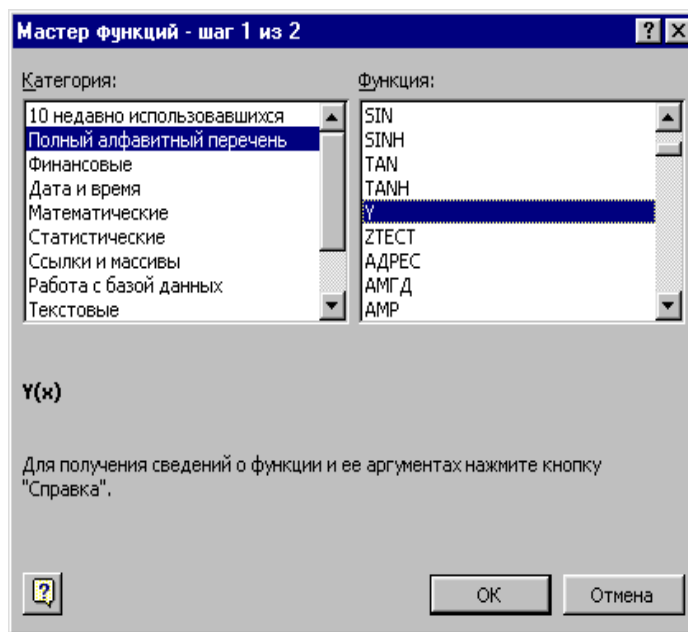


Рис. 4.3. Виклик функції Y()

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).

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 – глобальная константа

Const PI As Double = 3.14159

‘ x - глобальная переменная

Dim x As Double

‘ Функция Disc вычисляет площадь круга

Public Function Disc(R As Double) As Double

x = 2

Disc = PI * R ^ 2

End Function

‘ Функция Rec вычисляет площадь треугольника

Public Function Rec(a As Double, b As Double, c As Double) As Double

‘ p – локальная переменная

Dim p As Double

p = (a + b + c) / 2

Rec = Sqr (p * (p - a) * (p - b) * (p - c))

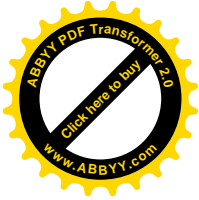
End Function

‘ 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)

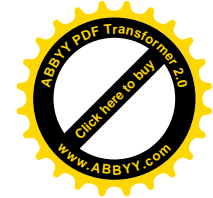
End Sub

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.



5. СТВОРЕННЯ ЕКРАННИХ ФОРМ І ВКЛЮЧЕННЯ ЇХ В ПРОЕКТИ

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. 5.1).

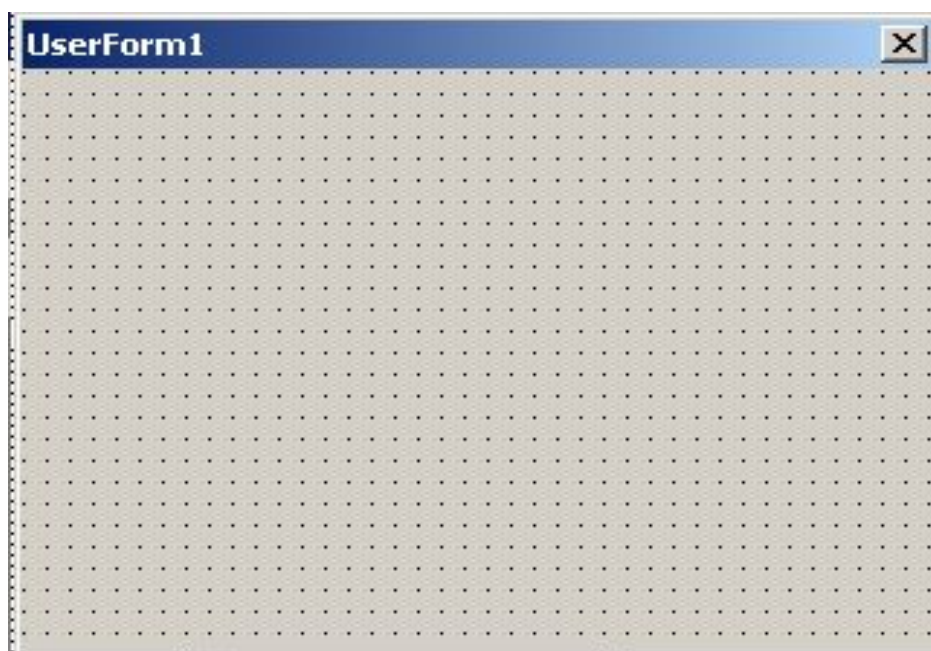
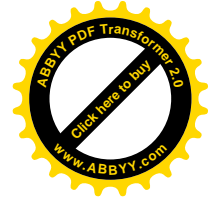


Рис. 5.1. Об'єкт форма

Як і будь-який інший об'єкт VBA форма має набір **властивостей**, основні з яких приведені в таблиці 5.1. Для здобуття довідки по будь-якій властивості досить виділити його у вікні властивостей і натискувати F1.



Таблиця 5.1. 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. 5.2 приведені найбільш часто використовувані події.

Таблиця 5.2. 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 форме уменьшает её размеры"  
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 5.3 і 5.4 приведені найбільш часто використовувані методи і інструкції для роботи формами.

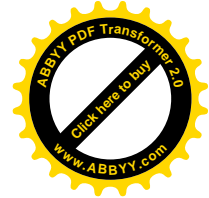
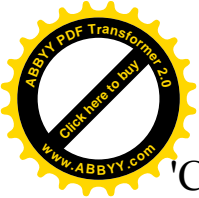
Таблиця 5.3. Основні методи форм

Метод	Опис
<i>Hide</i>	Приховує об'єкт UserForm, але не вивантажує його.
<i>Show</i>	Виводить на екран об'єкт UserForm.

Таблиця 5.4. 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
```

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 елементов" (рис. 5.2).

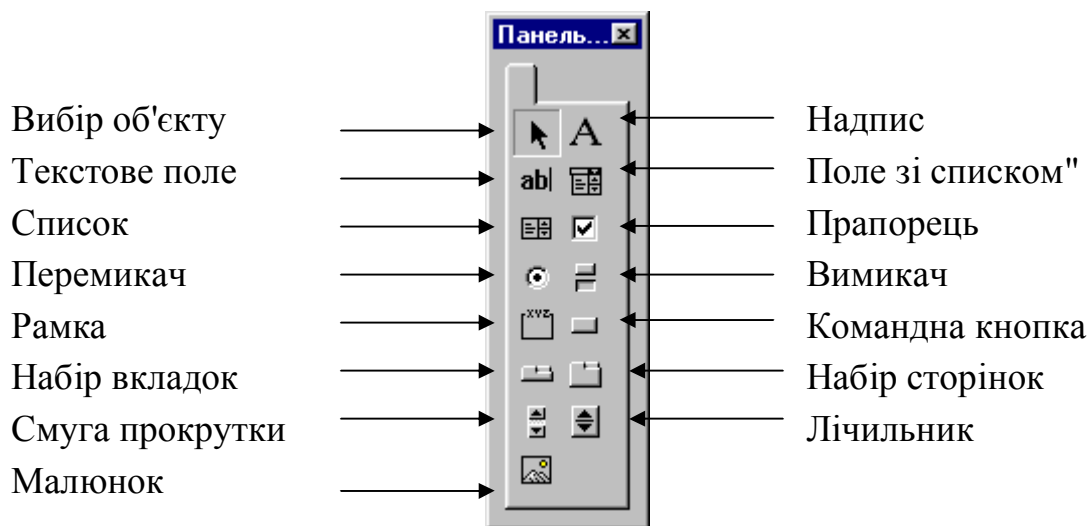
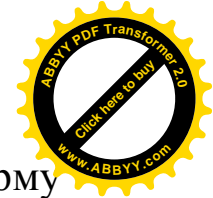


Рис. 5.2. Панель елементів VBA



За допомогою кнопок цієї панелі можна помістити на форму необхідний елемент управління. Для цього потрібно клацнути на значку елемента управління, далі при натиснутій лівій кнопці миші визначити розмір і місце розташування елемента 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.

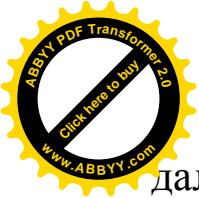
Кожен елемент (об'єкт), що управляє, характеризується набором властивостей (які можна змінювати в режимах конструювання або виконання), подій і методів.

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 елемента, а останні символи - призначення. У таблиці 5.5. представлені поєднання перших трьох символів для найчастіше використовуваних елементів.

Таблиця 5.5. Рекомендоване поєднання перших трьох символів імен

Об'єкт	Перші 3 символи імені	Приклад імені
Форма	frm	frmMyForm
Надпис	lbl	lblInfo
Текстове поле	txt	txtInput
Командна кнопка	cmd	cmdExit
Прапорець	chk	chkSound
Перемикач	opt	optLevel
Список	lsb	lsbTypes
Рамка	fra	fraChoices
Смуга прокрутки	vcb	vcbSpeed
Малюнок	pic	picChema

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. 5.6. 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**.

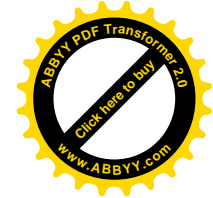
Таблиця 5.6. Властивості командних кнопок

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. 5.7.



Таблиця 5.7. 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. 5.8.

Таблиця 5.8. 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. 5.9.

Таблиця 5.9. 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"
```

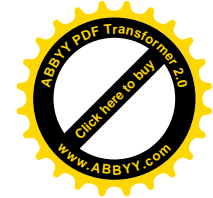
```
    List1.AddItem "Стул"
```

```
    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. 5.10.

Таблиця 5.10. Властивості перемикача

Властивість	Опис
<i>Caption</i>	Задає текст, що визначає призначення перемикача.
<i>(Name)</i>	Ім'я об'єкту, для програми VBA.
<i>Value</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.5.3). 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.

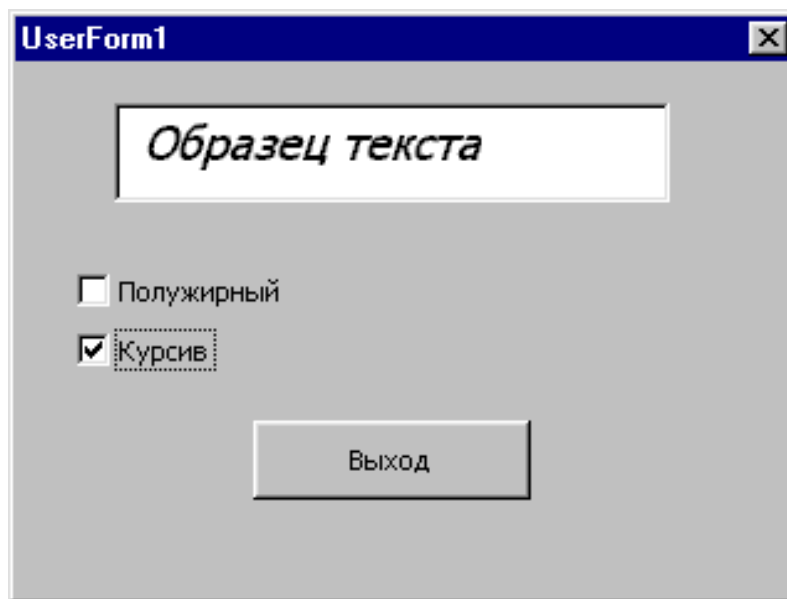
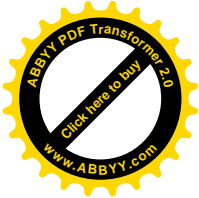
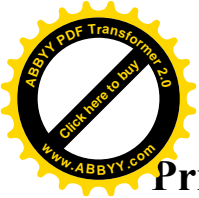


Рис. 5.3. Використання прапорців

```

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
```

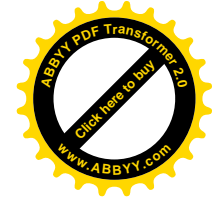
Рамка використовується для групування перемикачів або прапорців, і поміщається на форму раніше елементів, що знаходяться усередині неї. Перемикачі рамки, що знаходяться усередині, працюють як самостійна група і не впливають на стан перемикачів в інших рамках. Основною властивістю рамки є **Caption**, 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 5.11.

Таблиця 5.11. 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.



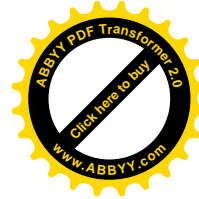
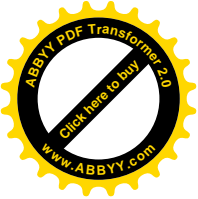
6. ПРИКЛАД ОБРОБКИ ТАБЛИЧНИХ ДАНИХ В MS EXCEL З ВИКОРИСТАННЯМ VBA

Завдання. Створити проект, форми користувача і програмні модулі для обробки сесії студентів:

- 1 Створити таблицю наступної структури:
 - індекс групи;
 - прізвище студента;
 - оцінки по п'яти іспитах;
 - стипендія;
- 2 Додати в таблицю довільну кількість рядків;
- 3 Розрахувати стипендію студентам за результатами сесії, причому відмінникам стипендію збільшити на 100%, тим, хто здав без трійок – збільшити на 30%, а за наявності хоч би однієї двійки стипендію не нараховувати;
- 4 Створити на 2-ому аркуші книги відомість для видачі стипендії студентам;
- 5 Сортувати дані в таблиці по вибраній колонці;
- 6 Змінити показники студента із заданим прізвищем;
- 7 Видалити з таблиці дані про студентів, що мають більше двох двійок.

Приклад таблиці:

Индекс группы	ФИО студента	Экзамен 1	Экзамен 2	Экзамен 3	Экзамен 4	Экзамен 5	Стипендия
УТР02Б	Петров	5	4	5	4	2	0
УТР02А	Иванов	4	5	3	3	3	100
УТР02А	Климова	5	4	5	4	3	100
УТР02А	Морозова	3	3	3	3	3	100
УТР02А	Абрамова	4	4	5	3	3	100
УТР02А	Сидоров	4	4	3	3	4	100
УТР02Б	Жуковская	4	3	5	4	4	100
УТР02А	Ющенко	3	3	3	3	3	100
УТР02Б	Жукова	5	5	5	5	5	200



Проект рішення
зачаці
frmMenu

frmVvod →

Ввод данных

Индекс группы: УТРО2А

Фамилия студента: Жукова

Оценки: Эк1: 4, Эк2: 5, Эк3: 3, Эк4: 4, Эк5: 5

Количество строк: 9

Ввод Очистить Выход

frmRas →

Расчет стипендии

Введите размер обычной стипендии

Расчет Выход

43

frmSort →

Сортировка данных

Выберите критерий сортировки:

- Индекс группы
- ФИО
- Оценка эк. 1
- Оценка эк. 2
- Оценка эк. 3
- Оценка эк. 4
- Оценка эк. 5
- Стипендия

Начать сортировку Отмена

Меню

Ввод

Добавить

Расчет

Ведомость

Сортировка

Корректур.

Удаление

Выход

frmL2 ←

Ведомость

Формировать

Лист1

Лист2

Очистить

Выход

Корректировка данных

Задайте ФИО студента (выбор или ввод)

выбор из списка

ввод

ФИО

Петров
Иванов
Юшимова
Морозова
Абрамова
Сидоров
Жуковская
Юценко
Жукова
Петров

Поиск

Отмена

frmKor ←

Корректировка данных

Индекс группы: УТРО2Б

Фамилия студента: Жуковская

Оценки: Эк1: 4, Эк2: 3, Эк3: 5, Эк4: 4, Эк5: 4

Ввод Очистить Выход

frmKor1 ←

Удаление!!!

Будут удалены студенты, имеющие более двух двоек по экзаменам!

Удалить!!! Отмена

frmDel ←

Програмный модуль «МЕНЮ»:

```
Private Sub CmdMenu1_Click()  
    Range("A2:Z100").Clear  
    Load frmVvod  
    frmVvod.Show  
End Sub
```

```
Private Sub CmdMenu2_Click()  
    Load frmVvod  
    frmVvod.Show  
End Sub
```

```
Private Sub CmdMenu3_Click()  
    Load frmRas  
    frmRas.Show  
End Sub
```

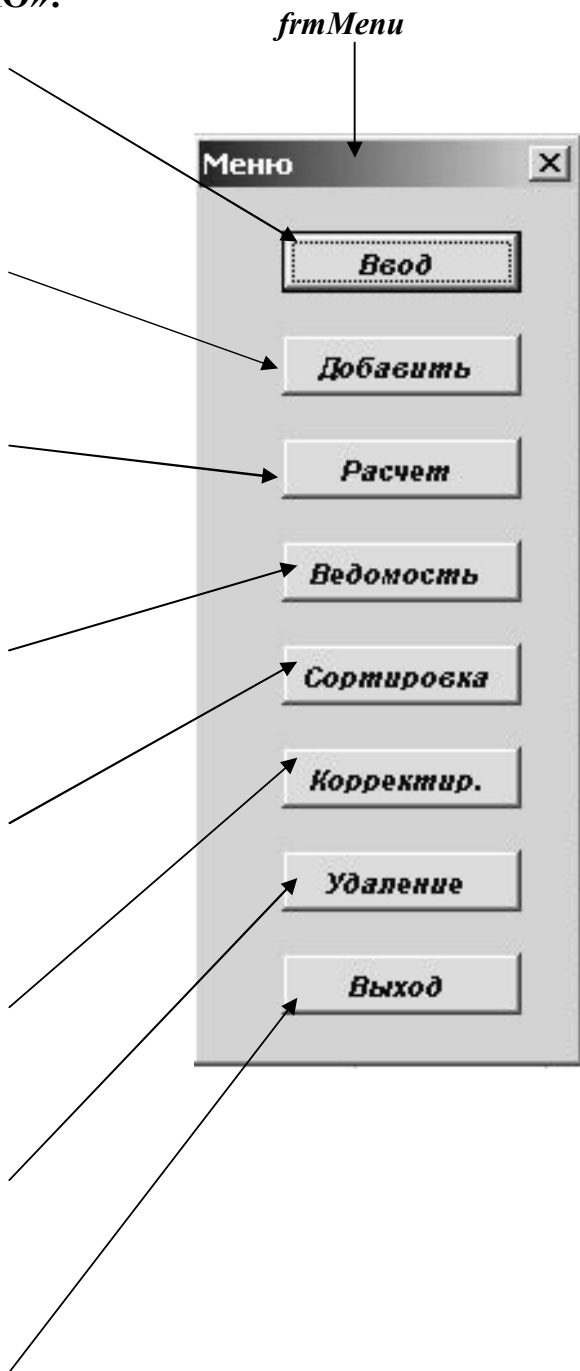
```
Private Sub CmdMenu4_Click()  
    Load frmL2  
    frmL2.Show  
End Sub
```

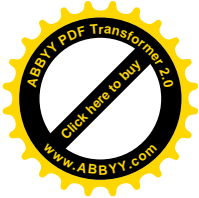
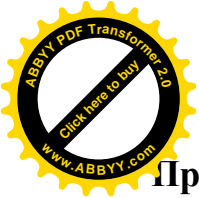
```
Private Sub CmdMenu5_Click()  
    Load frmSort  
    frmSort.Show  
End Sub
```

```
Private Sub CmdMenu6_Click()  
    Load frmKor  
    frmKor.Show  
End Sub
```

```
Private Sub CmdMenu7_Click()  
    Load frmDel  
    frmDel.Show  
End Sub
```

```
Private Sub CmdMenu8_Click()  
    End  
End Sub
```





Програмный модуль «ВВОД» (выкликается также и в режиме «ДОБАВИТЬ»):

```
Dim i As Double
```

```
Private Sub UserForm_Activate()
```

```
    ' Подсчет количества строк в таблице и вывод на форму в поле txtN
```

```
    i = 1
```

```
    Do While Cells(i, 1) > " "
```

```
        i = i + 1
```

```
    Loop
```

```
    txtN.Enabled = True
```

```
    txtN.Text = CStr(i - 2)
```

```
    txtN.Enabled = False
```

```
End Sub
```

```
Private Sub CmdVvod_Click()
```

```
    Cells(i, 1) = txtInd.Text
```

```
    Cells(i, 2) = txtFIO.Text
```

```
    Cells(i, 3) = CInt(txtM1.Text)
```

```
    Cells(i, 4) = CInt(txtM2.Text)
```

```
    Cells(i, 5) = CInt(txtM3.Text)
```

```
    Cells(i, 6) = CInt(txtM4.Text)
```

```
    Cells(i, 7) = CInt(txtM5.Text)
```

```
    txtN.Enabled = True
```

```
    txtN.Text = CStr(i - 1)
```

```
    txtN.Enabled = False
```

```
    i = i + 1
```

```
End Sub
```

```
Private Sub CmdCancel_Click()
```

```
    txtInd.Text = "": txtFIO.Text = "": txtM1.Text = ""
```

```
    txtM2.Text = "": txtM3.Text = "": txtM4.Text = ""
```

```
    txtM5.Text = ""
```

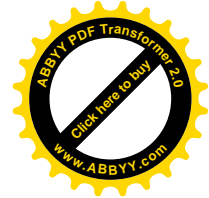
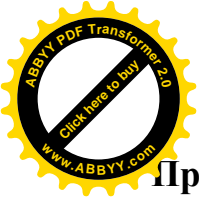
```
End Sub
```

```
Private Sub CmdExit_Click()
```

```
    frmVvod.Hide
```

```
End Sub
```

frmVvod



Програмный модуль «РАСЧЕТ»:

frmRas

Private Sub CmdRas_Click()*'Ввод обычной стипендии*

St = CSng(txtSt)

i = 2

Do While Cells(i, 1) > " "

' Подсчет двоек по экзаменам

k2 = 0: k4 = 0: k5 = 0

For j = 1 To 5

If Cells(i, j + 2) = 2 Then k2 = k2 + 1

If Cells(i, j + 2) = 4 Then k4 = k4 + 1

If Cells(i, j + 2) = 5 Then k5 = k5 + 1

Next j

'Расчет стипендии

If k5 = 5 Then

Stip = St * 2

Else

If k5 + k4 = 5 Then

Stip = St * 1.3

Else

If k2 > 0 Then

Stip = 0

Else

Stip = St

End If

End If

End If

Cells(i, 8) = Stip

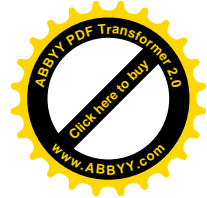
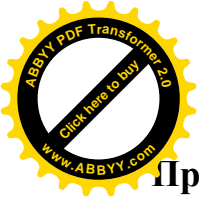
i = i + 1

Loop

End Sub**Private Sub CmdE_Click()**

frmRas.Hide

End Sub



Програмний модуль «ВЕДОМОСТЬ»:

Private Sub CmdF_Click()

```

Sheets(1).Activate
'Формирование шапки таблицы на 2-м листе
Sheets(2).Cells(1, 1) = "Ведомость начисления стипендии"
Sheets(2).Cells(2, 1) = "ФИО"
Sheets(2).Cells(2, 2) = "Сумма"
'i - номер строки на 1-м листе; k - на 2-м листе
i = 2: k = 3
Do While Cells(i, 2) > ""
    If Cells(i, 8) > 0 Then
        Sheets(2).Cells(k, 1) = Cells(i, 2)
        Sheets(2).Cells(k, 2) = Cells(i, 8)
        k = k + 1
    End If
    i = i + 1
Loop
Sheets(2).Activate

```

End Sub

Private Sub CmdL1_Click()

```

Sheets(1).Activate

```

End Sub

Private Sub CmdL2_Click()

```

Sheets(2).Activate

```

End Sub

Private Sub CmdO_Click()

```

Sheets(2).Activate
Range("A1:Z100").Clear
Sheets(1).Activate

```

End Sub

Private Sub CmdVuh_Click()

```

Sheets(1).Activate
frmL2.Hide

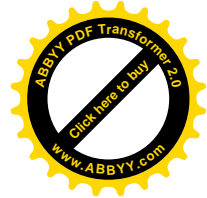
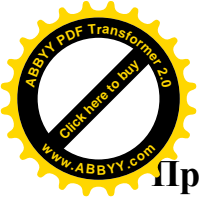
```

End Sub



Вид сформованої таблиці на аркуші 2:

	А	В	С
1	Ведомость	начисления стипендии	
2	ФИО	Сумма	
3	Абрамова	350	
4	Жукова	700	
5	Жуковская	350	
6	Климова	350	
7	Петров	455	
8	Сидоров	350	



Програмный модуль «СОРТИРОВКА»:

Private Sub UserForm_Activate()

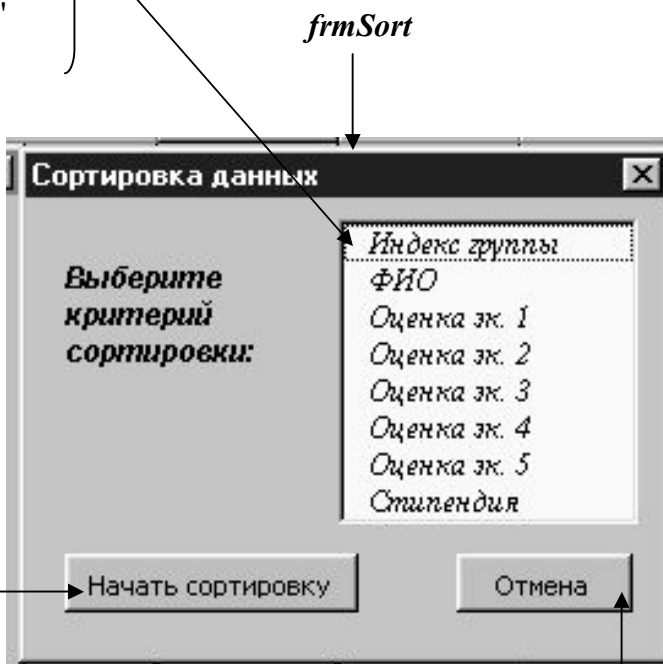
```
    'Заполнение списка ListBox1  
    ListBox1.AddItem "Индекс группы"  
    ListBox1.AddItem "ФИО"  
    ListBox1.AddItem "Оценка эк. 1"  
    ListBox1.AddItem "Оценка эк. 2"  
    ListBox1.AddItem "Оценка эк. 3"  
    ListBox1.AddItem "Оценка эк. 4"  
    ListBox1.AddItem "Оценка эк. 5"  
    ListBox1.AddItem "Стипендия"  
End Sub
```

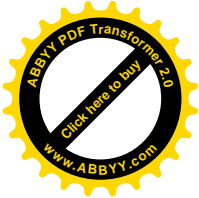
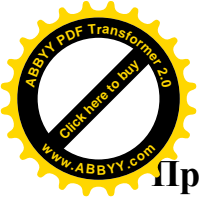
Private Sub CmdSort_Click()

```
    'Определение количества  
    'строк в таблице  
    N = 1  
    Do While Cells(N, 1) > " "  
        N = N + 1  
    Loop  
    N = N - 1  
    'Номер выбранного критерия  
    k = ListBox1.ListIndex + 1  
    'Сортировка  
    i = 2  
    Do While i <= N  
        x = Cells(i, k)  
        kx = i: i = i + 1  
        Do While i <= N  
            y = Cells(i, k)  
            ky = i: i = i + 1  
            If y < x Then  
                For j = 1 To 8  
                    r = Cells(kx, j)  
                    Cells(kx, j) = Cells(ky, j)  
                    Cells(ky, j) = r  
                Next j  
                x = y  
            End If  
        Loop  
        i = kx + 1  
    Loop  
    MsgBox "Сортировка по " & ListBox1.Text & " завершена!", , "Сортировка"  
End Sub
```

Private Sub CmdVuh_Click()

```
    frmSort.Hide  
End Sub
```





Програмный модуль «КОРРЕКТИРОВКА»: Модуль KOR

*' Заполнение списка ListBox1
' из колонки таблицы ФИО*

```
Private Sub UserForm_Activate()
N = 2
Do While Cells(N, 2) > " "
ListBox1.AddItem Cells(N, 2)
N = N + 1
Loop
ListBox1.ListIndex = 0
' Очистка текстового поля txtFIO
txtFIO = ""
End Sub
```

*' Заполнение списка ListBox1
' из колонки таблицы неповторяющимися ФИО*

```
Private Sub UserForm_Activate()
N = 2
Do While Cells(N, 2) > " "
M = ListBox1.ListCount
pr = 1
For j = 1 To M
ListBox1.ListIndex = j - 1
If Cells(N, 2) = ListBox1.Text Then pr = 0
Next j
If pr = 1 Then ListBox1.AddItem Cells(N, 2)
N = N + 1
Loop
ListBox1.ListIndex = 0
' Очистка текстового поля txtFIO
txtFIO = ""
End Sub
```

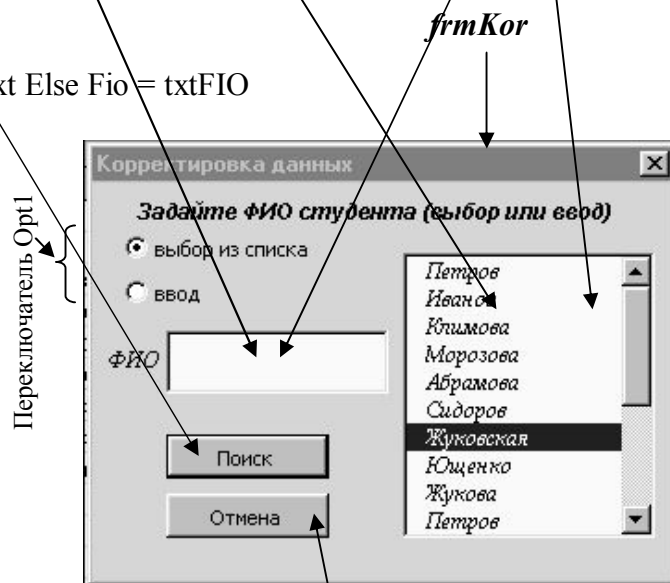
Private Sub CmdSeek_Click()

*' Выбор фамилии из списка ListBox1
' или введенной в поле txtFIO
' Поиск фамилии в таблице*

```
N = 2: Pr = 0
Do While Cells(N, 2) > " "
If Cells(N, 2) = Fio Then
Cells(N, 1).Select
Pr = 1
Exit Do
End If
N = N + 1
Loop
If Pr = 1 Then
' Загрузка формы для корректировки  
' если фамилия найдена frmKor1
Load frmKor1
frmKor1.Show
Else
' Вывод сообщения если фамилия не найдена
MsgBox "ФИО не найдена", vbCritical, "Корректировка"
End If
End Sub
```

Private Sub CmdO_Click()

```
frmKor.Hide
End Sub
```



Програмный модуль «КОРРЕКТИРОВКА» (продовження):

Модуль KOR1

*' i – номер найденной для корректировки строки в таблице
Dim i As Double*

Private Sub UserForm_Activate()

*' Вывод данных о студенте
' с найденной фамилией на форму frmKor1*

```
i = ActiveCell.Row
txtInd.Text = Cells(i, 1)
txtFIO.Text = Cells(i, 2)
txtM1.Text = CStr(Cells(i, 3))
txtM2.Text = CStr(Cells(i, 4))
txtM3.Text = CStr(Cells(i, 5))
txtM4.Text = CStr(Cells(i, 6))
txtM5.Text = CStr(Cells(i, 7))
```

End Sub

Private Sub CmdVvod_Click()

' Ввод новых данных после корректировки

```
Cells(i, 1) = txtInd.Text
Cells(i, 2) = txtFIO.Text
Cells(i, 3) = CInt(txtM1.Text)
Cells(i, 4) = CInt(txtM2.Text)
Cells(i, 5) = CInt(txtM3.Text)
Cells(i, 6) = CInt(txtM4.Text)
Cells(i, 7) = CInt(txtM5.Text)
```

End Sub

Private Sub CmdCancel_Click()

' Очистка текстовых полей

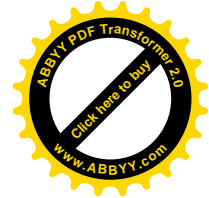
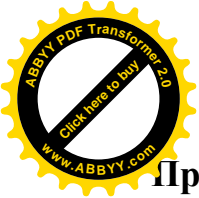
```
txtInd.Text = "": txtFIO.Text = "": txtM1.Text = ""
txtM2.Text = "": txtM3.Text = "": txtM4.Text = ""
txtM5.Text = ""
```

End Sub

Private Sub CmdExit_Click()

```
frmKor1.Hide
End Sub
```

frmKor1



Програмный модуль «УДАЛЕНИЕ»:

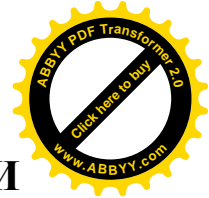
Private Sub CmdU_Click()

```
' N – количество строк в таблице  
N = 1  
Do While Cells(N, 1) > " "  
N = N + 1  
Loop  
N = N - 1  
' Удаление (u – количество удаленных строк)  
i = 2: u = 0  
Do While i <= N  
kx = i  
' Подсчет количества двоек K2  
k2 = 0  
For j = 1 To 5  
If Cells(i, j + 2) = 2 Then k2 = k2 + 1  
Next j  
i = i + 1  
If k2 >= 2 Then  
u = u + 1  
Do While i <= N  
For j = 1 To 8  
Cells(i - 1, j) = Cells(i, j)  
Next j  
i = i + 1  
Loop  
' Удаление последней строки  
Rows(N).Delete  
i = kx: N = N - 1  
End If  
Loop  
MsgBox "Удалено " & CStr(u) & " строк!", , "Удаление"  
End Sub
```

Private Sub CmdV_Click()

```
frmDel.Hide  
End Sub
```





7. ВАРІАНТИ ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Завдання

Створити проект, форми користувача і програмні модулі для обробки даних, організованих в таблиці Excel. У всіх завданнях обробляються дані таблиці, створеною в п.1.

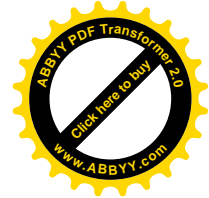
Варіант 1

1. Створити таблицю, що містить відомості про місячну заробітну плату робітників заводу.
Структура таблиці:
 - табельний номер;
 - прізвище робітника;
 - найменування цеху;
 - розмір заробітної плати в місяць.
2. Передбачити режим додавання в таблицю довільної кількості рядків.
3. Обчислити:
 - загальну суму виплат за місяць по заданому цеху,
 - середньомісячний заробіток цього цеху.
4. Створити на 2 аркуші книги відомість для нарахування заробітної плати робітників заданого цеху.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити розмір заробітної плати у робітника із заданим прізвищем на X% (значення X задано).
7. Видалити з таблиці 2 записи. Номери записів задати.



Варіант 2

1. Створити таблицю, що містить відомості про кількість виробів, зібраних складальниками цеху за тиждень.
Структура таблиці:
 - прізвище складальника;
 - кількість виробів, зібраних їм щодня протягом шестиденного тижня, тобто – в понеділок, вівторок, і так далі
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - загальну кількість деталей, зібраних складальниками цеху;
 - прізвище складальника, що зібрав найбільше число виробів, і визначити день, коли він досяг найвищої продуктивності праці.
4. Створити на 2 аркуші книги таблицю, що містить наступну інформацію: прізвище складальника і загальну кількість деталей, зібрану їм за тиждень.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити в X раз (значення X задано) кількість деталей, зібраних в заданий день тижня.
7. Видалити з таблиці запис про складальника із заданим прізвищем.



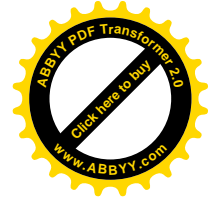
Варіант 3

1. Створити таблицю, що містить відомості про кількість виробів категорій А, В, С, зібраних робітником за місяць.
Структура таблиці:
 - прізвище складальника;
 - найменування цеху;
 - кількість виробів (по категоріях), зібраних робітником за місяць;
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Розрахувати:
 - загальну кількість виробів кожної категорії;
 - загальну кількість виробів, зібраних всіма робітниками заданого цеху;
 - середній розмір заробітної плати працівників заданого цеху.
4. Створити на 2 аркуші книги відомість для нарахування заробітної плати робітників заданого цеху.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Зменшити кількість виробів заданої категорії на $X\%$ (значення X - задано).
7. Видалити з таблиці записи, в яких кількість виробів заданої категорії менша Y штук (значення Y - задано).



Варіант 4

1. Створити таблицю, що містить відомості про телефонних абонентів.
Структура таблиці:
 - прізвище абонента;
 - номер телефону;
 - рік установки телефону;
 - наявність заборгованості в оплаті.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - загальну кількість телефонів, встановлених починаючи з заданого року по нинішній рік.
 - по заданому прізвищу абонента номер його телефону.
4. Створити на 2 аркуші книги таблицю, що містить прізвища боржників і їх телефони.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити номер телефону у абонента із заданим прізвищем
7. Видалити з таблиці запис, що містить заданий номер телефону.

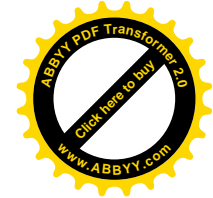


Варіант 5

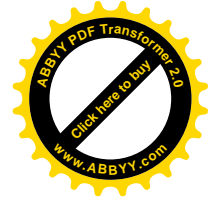
1. Створити таблицю, що містить відомості про асортимент іграшок в магазині.
Структура таблиці:
 - назва іграшки;
 - ціна;
 - кількість;
 - вікові межі, наприклад, 2-5, тобто від 2 до 5 років.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити вартість найдорожчої іграшки і її найменування. По значеннях A , B і X , що вводяться, видавати назву іграшки, яка за вартістю не перевищує X і підходить дитині від A до B років.
4. Створити на 2 аркуші книги таблицю, що містить наступну інформацію: найменування іграшок, які підходять дітям від X до Y років і їх ціни.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити ціну іграшок для заданої вікової групи на $Z\%$ (значення Z – задано).
7. Видалити з таблиці запис про іграшку із заданою назвою.



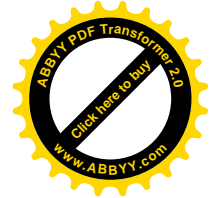
Варіант 6



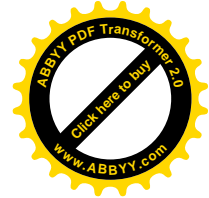
1. Створити таблицю, що містить відомості про здачу студентами першого курсу сесії.
Структура таблиці:
 - група;
 - прізвище студента;
 - оцінки по 5 іспитам
 - ознака участі в суспільній роботі: "Да"/"Нем".
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити загальне число активістів в списку.
4. Створити на 2 аркуші книги таблицю, що містить відомість нарахування стипендії студентам заданої групи. Розрахувати розмір стипендії за наступним принципом. Студентові, який отримав всі оцінки «5» і який бере активну участь в суспільній роботі, призначається підвищена стипендія – доплата 50%. Студентові, який отримав «4» і «5», призначається звичайна стипендія – її розмір необхідно задати. Студентові, який отримав одну оцінку «3», але який активно займається суспільною роботою, також призначається звичайна стипендія. Решті студентів стипендія не призначається. Передбачити введення розміру звичайної стипендії.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити ознаку участі в суспільному житті студентів з оцінками «55555» на протилежний.
7. Видалити з таблиці записи про студентів, що мають хоча би одну двійку.



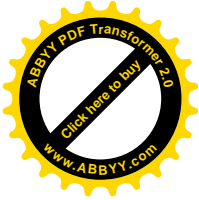
1. Створити таблицю, що містить інформацію про особисту колекцію книголюбів.
Структура таблиці:
 - шифр книги;
 - назва книги;
 - прізвище автора;
 - рік видання;
 - місце розташування (номер стелажу, номер шафи, номер полиці);
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - загальну кількість книг в колекції, а також число книг заданого року видання.
 - по заданому авторові і назві книги інформацію про місцезнаходження книги.
4. Створити на 2 аркуші книги таблицю, що містить інформацію про книги заданого автора, що знаходяться в колекції.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити місце розташування книг найпізнішого року видання.
7. Видалити з таблиці записи про книги, видані до заданого року.



1. Створити таблицю, що містить відомості про здачу сесії студентами.
Структура таблиці:
 - група;
 - прізвище студента;
 - результати іспитів по 5 предметам (наприклад, 55445);
 - результати заліків по 5 предметам (наприклад, 3ЗНЗЗ, де «З» – залік, а «Н» - незалік).
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Обчислити середній бал, отриманий кожним студентом заданої групи, і середній бал цієї групи по кожному предмету. Визначити загальну кількість заборгованостей (по іспитах і заліках в сумі) кожного студента заданої групи і загальне число студентів-боржників тієї ж групи.
4. Створити на 2 аркуші книги таблицю, що містить відомості про неуспішних студентів: групу, прізвище і кількість заборгованостей.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити результат іспиту із заданого предмету студентів із заданим прізвищем.
7. Видалити з таблиці записи про студентів, що мають три заборгованості.



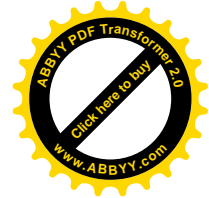
1. Створити таблицю, що містить відомості про рейси Аерофлоту.
Структура таблиці:
 - номер рейсу;
 - пункт призначення;
 - дата вильоту;
 - час вильоту;
 - час прибуття;
 - кількість вільних місць в літаку.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - номери рейсів і час відправлення літаків в задане місто;
 - по заданому місту і даті відправлення кількість вільних місць на рейси;
 - загальну кількість рейсів за добу в задане місто.
4. Створити на 2 аркуші книги таблицю, що містить номери рейсів, дату і час відправлення літаків в задане місто.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити час вильоту і час прибуття в заданого рейсу.
7. Видалити інформацію про нічні рейси (вважати нічними рейси з часом вильоту в інтервалі 23:00– 3:00).



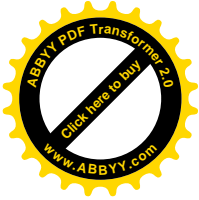
1. Створити таблицю, що містить відомості про асортимент взуття в магазині.
Структура таблиці:
 - артикул взуття (артикул починається з букви Д – для дамського взуття, М – для чоловічої, П – для дитячого взуття, наприклад, Д0321);;
 - найменування;
 - розмір;
 - кількість пар взуття;
 - вартість.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - вартість взуття заданого артикулу, і які розміри є в наявності;
 - загальну кількість пар дитячого взуття, наявного в магазині, і її сумарну вартість.
4. Створити на 2 аркуші книги таблицю, що містить інформацію про всі моделі взуття заданого виду.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити вартість взуття із заданим найменуванням на $X\%$ (значення X задано).
7. Видалити записи про чоловіче взуття з вартістю нижче заданої.



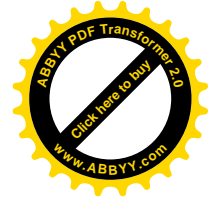
1. Створити 2 таблиці що містять відомості про десятьох нападаючих хокейних команд «Динамо» і «Шахтар».
Структура таблиць
 - прізвище хокеїста;
 - число закинутих ним шайб;
 - число зроблених ним голевих передач;
 - зароблений штрафний час.
2. Передбачити можливість додавання в таблиці довільної кількості рядків.
3. Обчислити:
 - загальне число шайб, забитих хокеїстами кожної команди,
 - сумарний штрафний час кожної команди
4. Створити на 2 аркуші книги таблицю, що містить прізвища шести кращих гравців з обох команд (по 3 гравця з кожної команди), і суму очків того гравця (голи + передачі).
5. Сортувати дані в таблицях по заданому стовпцю.
6. Для хокеїстів «Динамо» зменшити кількість забитих шайб на задане число, якщо це число більше 0.
7. Видалити інформацію про нападаючих обох команд, які заробили штрафний час вище заданого.



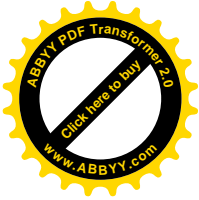
1. Створити таблицю, що містить відомості про те, яку з п'яти дисциплін бажає слухати студент.
Структура таблиці:
 - прізвище студента;
 - група;
 - найменування п'яти дисциплін (вибирає дисципліна помічається символом «1», інакше – пропуск;
 - середній бал успішності.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - кількість слухачів кожної дисципліни.
 - число слухачів заданої дисципліни, в яких середній бал перевищує заданий.
4. Створити на 2 аркуші книги таблицю, що містить прізвище, групу і середній бал всіх слухачів заданої дисципліни. Якщо число їх перевищить задане, то відібрати студентів, що мають вищий середній бал успішності.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити відомості для студента із заданим прізвищем.
7. Видалити інформацію про студентів з середнім балом нижче заданого.



1. Створити таблицю, що містить відомості про відправлення поїздів.
Структура таблиці:
 - номер поїзда;
 - станція призначення;
 - дата відправлення;
 - час відправлення;
 - час прибуття;
 - наявність (кількість) квитків у вагони кожного виду окремо: купе і плацкарт.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - кількість вільних місць в купейні вагони поїзда із заданими номером і датою;
 - кількість поїздів, наступних до заданої станції призначення.
4. Створити на 2 аркуші книги таблицю, що містить інформацію про поїзди, які вирушають до заданої станції в заданому інтервалі часу (часовий інтервал задати двома значеннями, наприклад 13:00 і 18:30).
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити відомості про наявність квитків на потяг із заданим номером.
7. Видалити записи про поїзди, що прямують до заданої станції.

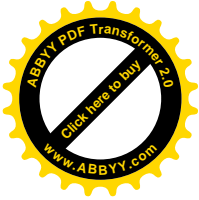


1. Створити таблицю, що містить відомості про співробітників інституту.
Структура таблиці:
 - прізвище;
 - стать;
 - назва відділу;
 - дата народження;
 - дата вступу на роботу;
 - посада;
 - оклад.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Розрахувати:
 - стаж роботи всіх співробітників;
 - середній стаж роботи співробітників заданого відділу;
 - кількість співробітників з окладом нижче заданого.
4. Створити на 2 аркуші книги таблицю, що містить список співробітників пенсійного віку (на сьогоднішній день), вказавши стаж роботи. Врахувати різницю в пенсійному віці жінок і чоловіків.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити оклад на задане число відсотків особам із стажем роботи вище заданого.
7. Видалити з таблиці відомості про співробітників пенсійного віку.

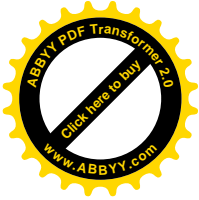


Варіант 15

1. Створити таблицю, що містить відомості про пацієнтів клініки.
Структура таблиці:
 - прізвище пацієнта;
 - стать;
 - вік;
 - місце мешкання (місто);
 - діагноз.
2. Розробити режим додавання в таблицю довільної кількості рядків.
3. Визначити:
 - кількість іногородніх, прибулих в клініку;
 - кількість пацієнтів із заданим діагнозом;
 - кількість пацієнтів пенсійного віку.
4. Створити на 2 аркуші книги таблицю, що містить список пацієнтів старше заданого віку із заданим діагнозом.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити інформацію у пацієнта із заданим прізвищем.
7. Видалити інформацію про пацієнтів пенсійного і дитячого віку.



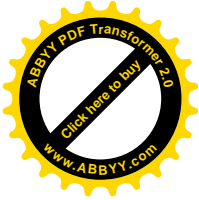
1. Створити таблицю, що містить відомості про багаж, зданий пасажиром в камеру схову.
Структура таблиці:
 - прізвище пасажира;
 - номер комірки;
 - шифр багажу;
 - кількість речей;
 - вага багажу.
2. Розробити режим додавання в таблицю довільної кількості рядків.
3. Визначити:
 - загальну середню вагу однієї речі;
 - багаж, в якого середня вага однієї речі відрізняється не більше ніж на X кг від загальної середньої ваги однієї речі (значення X задано);
 - число власників багажу з кількістю речей більшої заданої.
4. Створити на 2 аркуші книги таблицю, що містить інформацію про багаж (і про його власника), вага якого перевищує заданий.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити інформацію про багаж пасажира із заданим прізвищем.
7. Видалити відомості про багаж з максимальною вагою.



1. Створити таблицю, що містить відомості про роботу обмінного пункту за день (продаж і купівля одного виду валюти).
Структура таблиці:
 - найменування банку;
 - курс продажу;
 - курс купівлі.
 - кількість проданих у.о.;
 - кількість куплених у.о.;
2. Розробити режим додавання в таблицю довільної кількості рядків.
3. Визначити:
 - максимальну різницю між курсом продажу і купівлі у.о.;
 - сумарна кількість у.о., проданих всіма обмінними пунктами за день, і загальну суму, на яку вони продані.
4. Створити на 2 аркуші книги таблицю, що містить найменування фірми, кількість проданих у.о. і загальну суму продажів.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити курс покупки валюти в заданого банку.
7. Видалити інформацію про банки з максимальною різницею між курсом продажу і курсом купівлі.



Варіант 18



1. Створити таблицю, що містить відомості про замовлення на ремонт устаткування.
Структура таблиці:
 - номер замовлення;
 - дата оформлення замовлення;
 - прізвище замовника;
 - найменування устаткування;
 - складність ремонту (низька, середня, висока);
 - вартість;
 - дата закінчення ремонту.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - сумарну вартість всіх замовлень;
 - кількість замовлень на ремонт заданого вигляду складності;
 - мінімальна вартість ремонту.
4. Створити на 2 аркуші книги таблицю, що містить відомості про тривалість ремонту замовлень, оформлених протягом заданого сезону (наприклад, весною): номер замовлення, прізвище замовника, найменування устаткування, тривалість замовлення.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити вартість ремонту на $X\%$ (X задано) в устаткування із заданим найменуванням.
7. Видалити інформацію про замовлення з вартістю ремонту нижче середньої.

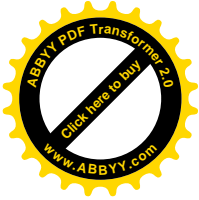


Варіант 19

1. Створити таблицю, що містить відомості про міжміські розмови.
Структура таблиці:
 - прізвище абонента;
 - номер телефону;
 - дата розмови;
 - код міста;
 - тривалість розмови в хвилинах;
 - вартість 1 хвилини.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - максимальну вартість розмови;
 - сумарну вартість всіх розмов;
 - загальна кількість розмов в місто із заданим кодом.
4. Створити на 2 аркуші книги таблицю, що містить відомості про вартість розмов, які відбулися в інтервалі між двома заданими датами.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити вартість всіх розмов на $X\%$ (значення X задано).
7. Видалити відомості про розмови абонента із заданим прізвищем.



1. Створити таблицю, що містить відомості про продаж товарів.
Структура таблиці:
 - найменування товару;
 - шифр товару;
 - оптова ціна;
 - роздрібна ціна;
 - дата продажу;
 - кількість проданого товару.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. По заданому шифру товару видавати інформацію про нього.
Визначити:
 - сумарний прибуток від продажу всіх товарів;
 - найменування товарів, що продаються за найвищою і найнижчою ціною.
4. Створити на 2 аркуші книги таблицю, що містить відомості про продаж товарів в інтервалі між двома заданими датами. Таблиця повинна містити найменування товару і сумарну виручку.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити роздрібну ціну на $X\%$ в товару із заданим найменуванням (значення X задано).
7. Видалити відомості про товари, продані протягом заданого сезону (наприклад, взимку).



1. Створити таблицю, що містить відомості про випуск виробів на заводі.
Структура таблиці:
 - найменування виробу;
 - код виробу;
 - номер цеху-виготівника;
 - дата виготовлення;
 - кількість;
 - собівартість;
 - ціна.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - сумарну різницю між собівартістю і ціною всіх виготовлених виробів;
 - загальна кількість виробів, виготовлених раніше заданої дати;
 - ціну виробу по заданому найменуванню.
4. Створити на 2 аркуші книги таблицю, що містить відомості про товари, виготовлені в заданому цеху і собівартість яких складає половину ціни.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити собівартість і ціну в заданого виробу.
7. Видалити відомості про товари з цінами нижчими за середню ціну.



1. Створити таблицю, що містить відомості про почасову оплату праці робітників.
Структура таблиці:
 - прізвище робітника;
 - розряд;
 - номер цеху;
 - дата;
 - вартість однієї години;
 - кількість відпрацьованого часу.
2. Розробити режим додавання в таблицю довільної кількості рядків.
3. Визначити:
 - середню кількість часу, відпрацьованого за день;
 - максимальну вартість однієї години;
 - по заданій даті і прізвищу робітника кількість відпрацьованого часу.
4. Створити на 2 аркуші книги таблицю, що містить наступні відомості про працівників заданого цеху: прізвище працівника; сумарна вартість відпрацьованого ним часу.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити вартість однієї години у працівників заданого цеху на $X\%$ (значення X задано).
7. Видалити інформацію про працівників, у яких кількість відпрацьованого часу за день менше середнього.



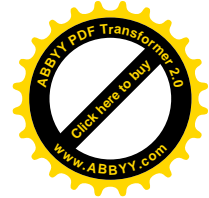
1. Створити таблицю, що містить відомості про замовлення на пошиття одягу.
Структура таблиці:
 - прізвище клієнта;
 - номер замовлення;
 - найменування виробу;
 - дата прийому замовлення;
 - дата виконання замовлення;
 - прізвище майстра;
 - вартість.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - кількість замовлень, виконаних майстром із заданим прізвищем;
 - вартість найбільш дорогого замовлення;
 - середню вартість замовлень.
4. Створити на 2 аркуші книги таблицю, що містить прізвище клієнта, номер замовлення і тривалість його виконання.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити відомості про замовлення, оформлені пізніше заданої дати.
7. Видалити інформацію про клієнта із заданим прізвищем.



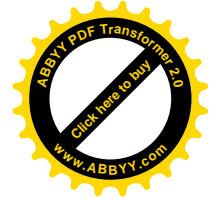
1. Створити таблицю, що містить відомості про товари, що експортуються.
Структура таблиці:
 - шифр товару;
 - найменування товару;
 - країна, що імпортує товар;
 - розмір партії, що поставляється, в штуках;
 - вартість 1 штуки в у.о..
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - сумарний об'єм товарів, що імпортуються заданою країною;
 - сумарну вартість партії товару по заданому шифру
 - мінімальну вартість товару.
4. Створити на 2 аркуші книги таблицю, що містить відомості про вартість товарів, що імпортуються заданою країною. Таблиця повинна містити найменування товару і сумарну вартість партії.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити на $X\%$ вартість заданого товару (значення X задано).
7. Видалити інформацію про товар, що має мінімальну вартість.



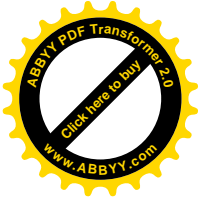
1. Створити таблицю, що містить відомості про виготовлення деталей.
Структура таблиці:
 - номер цеху;
 - прізвище робітника;
 - кількість виготовлених деталей;
 - кількість бракованих деталей;
 - оплата за виготовлення однієї якісної деталі.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - загальна кількість бракованих деталей, виготовлених всіма майстрами заданого цеху;
 - суму штрафу за кожну браковану деталь (штраф складає 20% від оплати за якісну);
 - прізвище майстра, що виготовив максимальну кількість якісних деталей.
4. Створити на 2 аркуші книги таблицю, що містить відомості про оплату праці робітників. Таблиця повинна містити прізвище робітника, номер цеху і суму до виплати з урахуванням штрафу і податку (податок складає $X\%$ від вартості оплати).
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити оплату співробітникам, що працюють без браку на $Y\%$ (значення Y задане).
7. Видалити відомості про працівників, в яких відсоток браку складає більше 50% від кількості виготовлених.



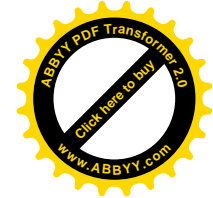
1. Створити таблицю, що містить інформацію про вартість лікування хворих в стаціонарній клініці.
Структура таблиці:
 - назва хвороби;
 - номер відділення;
 - тривалість лікування;
 - вартість 1 дня лікування;
 - витрати на ліки.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - витрати на ліки по заданій назві хвороби;
 - середню вартість 1 дня лікування;
 - розрахувати сумарну вартість лікування кожної хвороби, включаючи витрати на ліки.
4. Створити на 2 аркуші книги таблицю, що містить відомості про вартість лікування в заданому відділенні.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити вартість 1 дня лікування заданої хвороби на $X\%$ (значення X задано).
7. Видалити відомості про лікування хвороб з мінімальною вартістю одного дня.



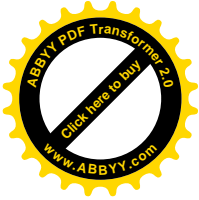
1. Створити таблицю, що містить відомості про продаж книг.
Структура таблиці:
 - назва книги;
 - прізвище автора;
 - номер магазину;
 - ціна;
 - продано;
 - залишок.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - кількість проданих книг в заданому магазині;
 - сумарну вартість всіх непроданих книг;
 - середню ціну однієї книги.
4. Створити на 2 аркуші книги таблицю, що містить сумарну вартість книг, проданих кожним магазином
5. Сортувати дані в таблиці по заданому стовпцю.
6. Зменшити ціну на книги, в яких залишок перевищує кількість проданих більш ніж в 2 рази.
7. Видалити відомості про книги з ціною нижче заданою.



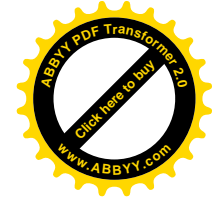
1. Створити таблицю, що містить відомості про облік використання матеріалів.
Структура таблиці:
 - шифр деталі;
 - кількість виготовлених деталей;
 - кількість браку;
 - шифр матеріалу;
 - витрата матеріалу на 1 деталь.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - для всіх деталей сумарна витрата матеріалу на брак;
 - кількість якісних деталей;
 - деталь, на яку витрачається більше всього матеріалу.
4. Створити на 2 аркуші книги таблицю, що містить шифр деталі і відсоток браку.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити відомості про деталь із заданим шифром.
7. Видалити відомості про деталі, на виготовлення яких витрачається кількість матеріалу, що перевищує задане значення.



1. Створити таблицю, що містить відомості про продаж ліків в аптеках
Структура таблиці:
 - номер аптеки;
 - назва ліків;
 - категорія ліків (антибіотик, жарознижуюче, вітамін, снодійне і так далі);
 - ціна 1 упаковки;
 - кількість проданих упаковок;
 - залишок;
 - дата закінчення терміну придатності.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - сумарна кількість упаковок ліків, проданих всіма аптеками і що належать заданій категорії;
 - вартість всіх непроданих упаковок;
 - середню вартість ліків.
4. Створити на 2 аркуші книги таблицю, що містить інформацію про прострочені ліки: номер аптеки, назву, категорію і дату закінчення терміну.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Змінити ціну в ліків, що належать заданій категорії.
7. Видалити відомості про прострочені ліки.

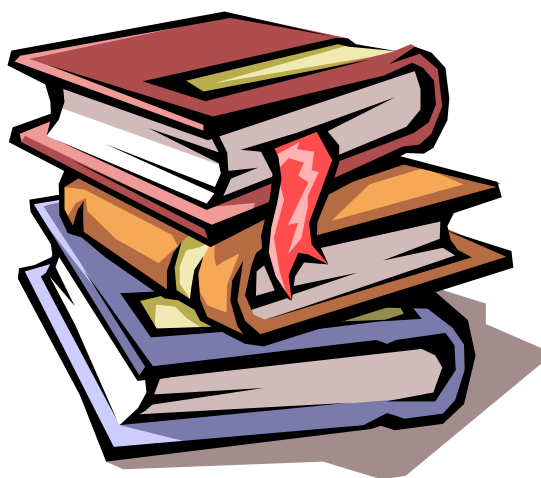


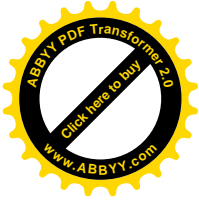
1. Створити таблицю, що містить відомості про співробітників фірми.
Структура таблиці:
 - прізвище співробітника;
 - посада;
 - дата народження;
 - дата вступу на роботу;
 - номер відділу;
 - оклад.
2. Передбачити можливість додавання в таблицю довільної кількості рядків.
3. Визначити:
 - вік співробітників (кількість повних років) під час вступу на роботу і на поточний час;
 - кількість співробітників заданого відділу молодших за X роки (значення X задано);
 - мінімальний розмір окладу.
4. Створити на 2 аркуші книги таблицю, що містить відомості про 5-ть самих високооплачуваних співробітників фірми.
5. Сортувати дані в таблиці по заданому стовпцю.
6. Збільшити на $Y\%$ оклади співробітникам, що пропрацювали не менше Z років (значення X і Z задані).
7. Видалити інформацію про співробітників старше заданого віку.



СПИСОК ЛИТЕРАТУРИ

1. Браун С. VISUAL BASIC 6: Учебный Курс. – Питер, 1999. – 688с.
2. Visual Basic 6: Полное Руководство. – СПб: ВHV - Санкт - Петербург, 1999. – 992с.
3. Брюс Мак-Кинни Крепкий орешек Visual Basic. // Русская редакция, 1999. – 632с.
4. М. Ченнел Разработка приложений на Microsoft Visual Basic 6.0 // Трейдинг Лимитед, 2000. – 400с.
5. А. Гарнаев Самоучитель VBA. Технология создания пользовательских приложений // ВHV - Санкт - Петербург, 1999. – 512 с.
6. А. Гарнаев Использование MS Excel и VBA в экономике и финансах. – СПб: БХВ – Санкт-Петербург, 2000. – 336 с.
7. Кен Гетц, Майк Джилберт Программирование в Microsoft Office. Руководство по Visual Basic for Applications. – СПб: ВHV, 2000. – 384с.
8. Microsoft Press Руководство программиста по Visual Basic для MS Office 97 // Русская редакция, 1997. – 544с.
9. Deborah Kurata Doing Objects in Microsoft Visual Basic 6 // 1999. – 642 p.





"ОСОБЛИВОСТІ ОБРОБКИ ТАБЛИЧНИХ ДАНИХ

EXCEL В IDE НА МОВІ VBA"

Методичні вказівки і завдання

(для студентів всіх спеціальностей)

Автори: *Зензеров Володимир Іванович, ст. викладач*
Славінська Людмила Василівна, ст. викладач
Єдемська Євгенія Миколаївна, ст. викладач