



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ



ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

**"Основы программирования
в среде Visual Basic for Application"**

**Методические указания и задания
к выполнению лабораторных работ
(для студентов всех специальностей)**

Донецк 2008



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**"ОСНОВЫ ПРОГРАММИРОВАНИЯ
В СРЕДЕ VISUAL BASIC FOR APPLICATION"**

**Методические указания и задания
к выполнению лабораторных работ**

(для студентов всех специальностей)

*Рассмотрено на заседании кафедры
вычислительной математики
и программирования
Протокол № 9 от 4 апреля 2008 г.*

*Утверждено на учебно-издательском
совете ДонНТУ
Протокол №4 от 19 мая 2008 г.*

Донецк ДонНТУ 2008



УДК 681.3.06(071)

О 23

"Основы программирования в среде Visual Basic for Application".

Методические указания и задания к выполнению лабораторных работ

(для студентов всех специальностей)./ Авторы: Зензеров В.И., Славинская Л.В.,
Едемская Е.Н. – Донецк: ДонНТУ, 2008. – 88 с.

Методические указания посвящены вопросам практической работы со средой VBA. Содержат краткое описание теоретических основ алгоритмического языка, описание среды VBA, а также задания для выполнения лабораторных работ по темам: «Знакомство со средой VBA», «Создание экранных форм: работа с элементами управления», «Создание программных модулей VBA для обработки табличных данных MS Excel». Материал может быть полезен для студентов всех специальностей, изучающих курсы «Основы информационных технологий», «Информатика и компьютерная техника», а также для преподавателей, аспирантов и других пользователей персональных компьютеров.

Авторы:

Зензеров В.И., ст. преподаватель,
Славинская Л.В., ст. преподаватель,
Едемская Е.Н., ст. преподаватель

Отв. за выпуск:

д.т.н., проф. Павлыш В.Н.



ОГЛАВЛЕНИЕ

Введение.....	4
1. Работа в интегрированной среде разработки приложений на языке VBA.....	5
2. Основы программирования на языке VBA.....	9
3. Основные операторы языка VBA.....	19
4. Структура программы. Модули, процедуры и функции.....	30
5. Создание форм в VBA и включение их в проекты	36
6. Лабораторный практикум.....	47
Список литературы.....	87



ВВЕДЕНИЕ



Мечта пользователей персонального компьютера иметь универсальное, мощное, гибкое и удобное средство, позволяющее быстро и эффективно решать самые разнообразные экономические и финансовые задачи, воплотилась в MS EXCEL. Visual Basic for Application (VBA) существенно обогатил и усилил MS EXCEL. VBA – это сочетание одного из самых простых языков программирования и всех вычислительных возможностей MS EXCEL. С помощью VBA вы сможете легко и быстро создавать разнообразные приложения, даже не являясь специалистом в области программирования. VBA содержит графическую среду, позволяющую наглядно конструировать экранные формы из управляющих элементов. VBA в сочетании с возможностями MS EXCEL позволяет решать задачи, о решении которых только средствами MS EXCEL ранее даже и не говорили.

Настоящее методическое пособие поможет вам грамотно и эффективно использовать компьютер в своей работе, изучить особенности программирования на языке Visual Basic for Application для обработки данных, представленных в виде таблиц в MS EXCEL, и, таким образом, расширить возможности стандартных приложений (в частности, популярного пакета MS Office). Расширение возможностей стандартных компонентов пакета Microsoft Office для решения конкретных задач достигается созданием собственных приложений путем программирования на языке VBA, который доступен любому квалифицированному пользователю персональных компьютеров.

Изложенный в пособии методический материал с примерами и приведенные варианты заданий предполагают, что пользователь знаком с основами программирования в среде VBA и желает расширить свои знания в области применения VBA для решения экономических задач.



Главное меню – это строка текста, расположенная в верхней части окна Visual Basic, и состоит из нескольких пунктов.

Меню **Файл** предназначено для работы с файлами, из которых образуются приложения. В нем можно создавать, сохранять и печатать проекты. Меню **Правка** выполняет стандартные операции с буфером обмена – вырезание, копирование и вставка. Они применяются не только к фрагментам программы, но и к управляющим элементам. В меню **Вид** включаются режимы просмотра различных компонентов и инструментов. Просматривать можно формы и программные модули. Меню **Вставка** позволяет добавлять процедуры, формы, модули и модули класса. Команды меню **Формат** определяют расположение и размеры элементов и форм. При помощи команд меню **Отладка** можно запустить и остановить приложение, расставить точки прерывания и выбрать просматриваемые объекты, а также выполнить другие операции, помогающие следить за работой приложения. Команды меню **Запуск** запускают и останавливают приложение, прерывают и возобновляют выполнение программы, что особенно удобно в процессе отладки. Меню **Сервис** позволяет включить дополнительные элементы, запустить макросы и настроить параметры редактора. Меню **Окно** позволяет выстроить окна IDE (каскадное или мозаичное расположение), упорядочить значки свернутых форм, а также создает список, позволяющий быстро перейти к одному из открытых окон IDE. Меню **?** – помощь пользователю. Для быстрого вызова главного меню необходимо нажать клавишу F10.

Панель инструментов находится под главным меню (рис. 1.2). Если она отсутствует, необходимо выполнить команду **Вид → Панели инструментов → Стандарт**.

Окно проекта напоминает собой окно Проводника Windows, и предназначено для быстрого просмотра составляющих проекта, который объединяет в себе все объекты, составляющие приложение (рис. 1.3). Это стандартные объекты открытого приложения Microsoft Office (документ редактора MS Word, книга и её листы MS Excel), формы, модули и классы.

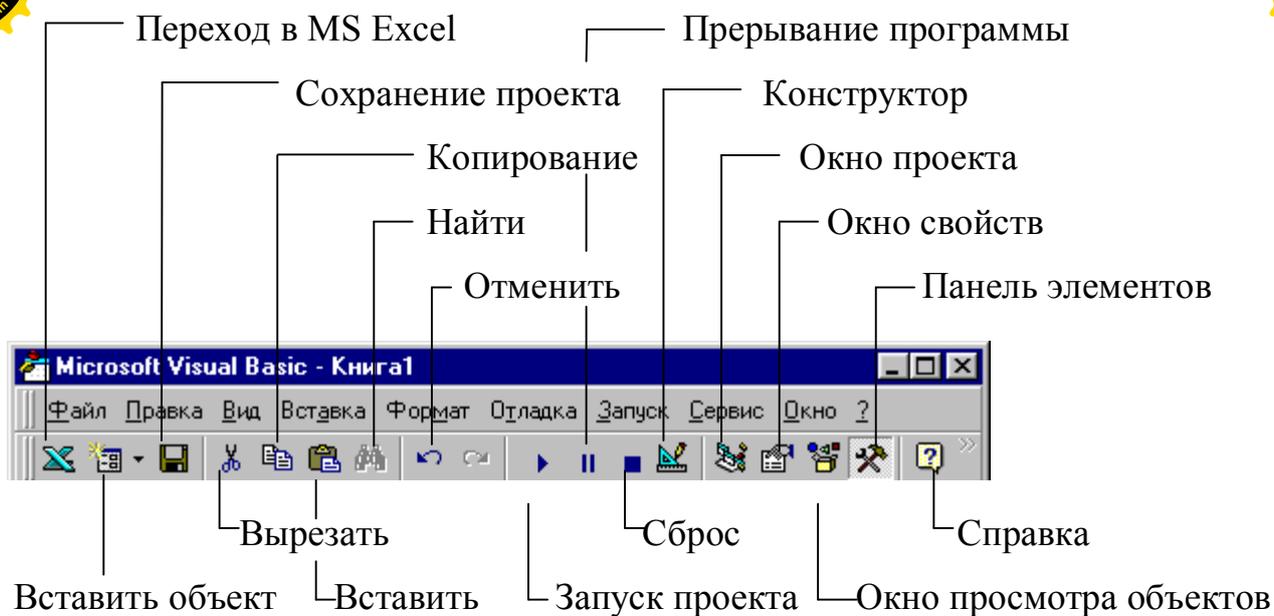
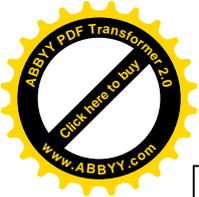


Рис. 1.2. Панель инструментов VBA

Окно свойств отображает различные атрибуты выделенного объекта (рис. 1.4). Все объекты (формы, управляющие элементы и т.д.) имеют атрибуты, которые изменяют не только внешний вид объекта, но и его поведение. Все эти атрибуты называются свойствами. Следовательно, каждый объект обладает набором свойств.

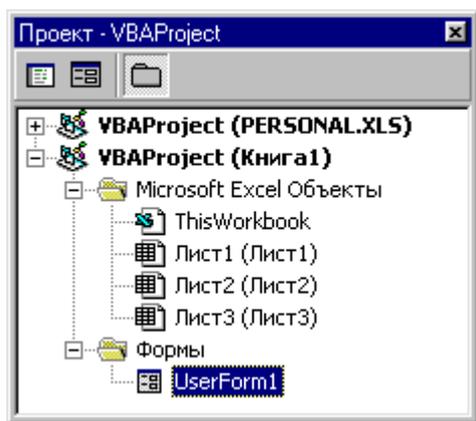


Рис. 1.3. Окно проекта

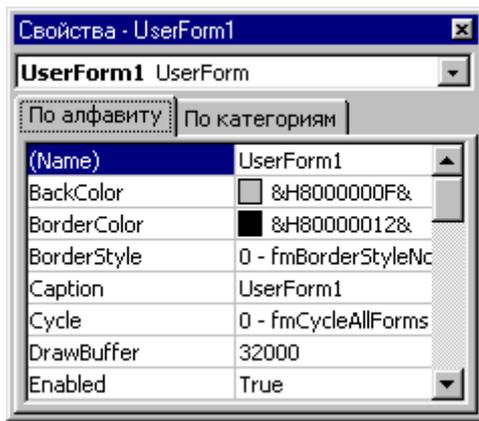


Рис. 1.4. Окно свойств

Окно контрольных значений позволяет просматривать значения контрольных переменных в процессе проверки правильности работы (отладки) проекта, что позволяет находить ошибки в логике работы программ.

Конструктор форм расположен в центре экрана редактора VBA. Здесь выводится либо изображение формы, что позволяет производить визуальное конструирование макета формы и расположенных на ней элементов, либо окно программы (более подробно об этом будет рассказано ниже).



2. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ VBA

Процесс разработки программы на языке VBA – **проекта**, может состоять из нескольких этапов, в зависимости от конечного результата. Если необходимо получить программу, которая будет производить определенные вычисления или действия, расширяющие математические возможности стандартного приложения Microsoft Office, то достаточно создать **программный модуль**. Для применения этой программы можно поместить в рабочей области приложения кнопку, нажатие которой будет вызывать выполнение программы. Для этого в приложении необходимо включить панель инструментов с помощью команды **Вид → Панели инструментов → Элементы управления**, а затем создать кнопку с соответствующим программным кодом. Либо выполнять программу с помощью команды **Сервис → Макрос → Макросы**.

Разработка “полноценной” программы (для выполнения которой требуется отдельное окно, с различными элементами управления) будет включать два этапа. Первый этап – этап визуального программирования, на котором создается окно (**форма**) программы, где располагаются необходимые элементы управления. Второй – этап программирования, на котором создаются части программы (**процедуры**), выполняющиеся в ответ на определенные события. Событием является, например, щелчок левой кнопкой мыши на командной кнопке (событие Click), нажатие клавиши на клавиатуре (событие KeyPress) и т.д. Использовать такое приложение можно нажатием кнопки  – «Запуск проекта».

2.1. Объекты, свойства и методы VBA

Одним из основных понятий VBA является объект. **Объект** – это то, чем вы управляете с помощью программы на языке VBA, например, форма, кнопка, рабочий лист или диапазон ячеек MS Excel. Каждый объект обладает некоторыми **свойствами**. Например, форма может быть видимой или невидимой в данный момент на экране. Другой пример свойства объекта – шрифт для ото-



Выражения информации в ячейке (объекте) рабочего листа.

Объект содержит также список методов, которые к нему применимы.

Методы – это то, что вы можете делать с объектом. Например, показать форму на экране или убрать её можно с помощью методов Show и Hide.

Таким образом, **объект** – это программный элемент, который имеет свое отображение на экране, содержит некоторые переменные, определяющие его **свойства**, и некоторые **методы** для управления объектом. Например, в MS Excel имеется много встроенных объектов:

Range (“Адрес”)	- диапазон ячеек (может включать только одну ячейку).
Cells (i, j)	- ячейка, находящаяся на пересечении i-й строки и j-го столбца рабочего листа MS Excel (i и j – целые числа).
Rows (№ строки)	- строка с заданным номером.
Columns (№ столбца)	- столбец с заданным номером
Sheets (“Имя”)	- лист с указанным именем.
Sheets (№ листа)	- лист с указанным номером.
WorkSheet	- рабочий лист.

Установка значений свойств – это один из способов управления объектами. Синтаксис установки значения свойства объекта следующий:

Объект. Свойство = Выражение

Основным свойством объектов **Cells** и **Range**, является **Value** (значение), которое, однако, можно не указывать. Например:

Range(“A5:A10”). **Value** = 0 или **Range**(“A5:A10”) = 0 – в диапазон ячеек A5:A10 заносится значение 0.

Cells(2, 4). **Value** = n или **Cells**(2, 4) = n – в ячейку, находящуюся на пересечении 2-й строки и 4-го столбца (ячейка с адресом “D2”), заносится значение переменной n.

Синтаксис чтения свойств объекта следующий:

Переменная = Объект. Свойство

Например:

Xn = **Cells**(1, 2).**Value** или **Xn** = **Range**(“B1”).**Value** – переменной Xn



присваивается значение из ячейки B1 текущего рабочего листа.

Синтаксис применения методов к объекту:

Объект. Метод

Например:

Sheets(2).Activate – сделать активным лист с №2.

Sheets("Диаграмма").Delete – удалить лист с именем "Диаграмма".

Range("A5:A10").Clear – очистить диапазон ячеек A5:A10.

Range("A2:B10").Select – выделить диапазон ячеек A2:B10.

В MS Excel имеются объекты, которые содержат другие объекты. Например, рабочая книга содержит рабочие листы, рабочий лист содержит диапазон ячеек и т.д. Объектом самого высокого уровня является **Application** (приложение). Если вы изменяете его свойства или вызываете его методы, то результат применяется к текущей работе MS Excel. Например:

Application.Quit - завершение работы с Excel.

Отметим, что точка после имени объекта может использоваться для перехода от одного объекта к другому. Например, следующее выражение очищает вторую строку рабочего листа **Май** в рабочей книге **Отчет**:

Application.Workbooks("Отчет").Worksheets("Май").Rows(2).Delete

Нужно отметить следующее:

- Можно не писать имя объекта **Application**, так как это подразумевается по умолчанию.
- При работе с подобъектом уже активизированного объекта нет необходимости указывать содержащий его объект.
- VBA использует некоторые свойства и методы, которые возвращают объект к которому они относятся (это позволяет быстро указывать нужный объект). Примеры таких свойств: **ActiveCell** (активная ячейка), **ActiveSheet** (активный лист), **ActiveWorkBook** (активная рабочая книга). Так, установить значение активной ячейки можно следующим образом:

ActiveCell.Value = " Да".



2.2. Описание данных

Все объекты, которыми оперирует язык программирования VBA, относятся к определенному типу.

Тип данных определяет:

- область возможных значений переменной;
- структуру организации данных;
- операции, определенные над данными этого типа.

Типы данных подразделяются на простые (скалярные) и сложные (структурированные). У простых типов данных возможные значения данных едины и неделимы. Сложные же типы имеют структуру, в которую входят различные простые типы данных. Скалярные типы данных представлены в таблице 2.1.

Таблица 2.1. Скалярные типы VBA

Имя типа	Русское название типа	Возможные значения
Boolean	Логический	True, False
Byte	Байтовый	0...255
Integer	Целое	-32768...+32767
Long	Длинное целое	-2147483648...+2147483647
Single	Число с плавающей точкой	-3,4E38...-1,4E-45 для отрицательных значений. 1,4E-45...3,4E38 для положительных значений.
Double	Число с плавающей точкой двойной точности	-1,7E308...-4,9E-324 для отрицательных значений. 4,9E-324...1,7E308 для положительных значений.
Currency	Денежный	Десятичные числа с фиксированной позицией запятой. Возможны 15 цифр до запятой и 4 после.
String	Строковый	Есть два вида строк: строки фиксированной длины (до 2^{16} символов) и строки переменной длины (до 2^{31} символов). Данные записываются в кавычках.
Date	Дата	Даты изменяются в диапазоне от 1.01.100г. до 31.12.9999г.
Object	Объект	Ссылка на объект (указатель)
Variant	Вариант	Универсальный тип, значением которого могут быть данные любого из перечисленных выше типов, объекты, значения NULL и значения ошибок ERROR.



Переменные в программе можно описывать или не описывать. В последнем случае ей будет присвоен тип **Variant**. Явно описывать переменную можно как в начале блока, так и в любом месте, где возникла необходимость использовать новую переменную. Лучше все переменные описывать явно и, как правило, в начале блока. Для запрета использования переменных, которые не были описаны явно, в начало программы необходимо вставить оператор **Option Explicit**.

2.2.1. Описание простых переменных

Описание простых переменных имеет следующий синтаксис:

Dim *ИМЯ_ПЕРЕМЕННОЙ* **As** *ИМЯ_ТИПА*

Одним оператором **Dim** можно описать произвольное число переменных, но конструкция **As** должна быть указана для каждой из них, иначе переменным без **As** будет присвоен тип **Variant**.

Например.

Dim X As Byte, Z As Integer, C, Слово As String

Здесь переменная *X* - это переменная байтового типа, переменная *Z* - целого типа, переменная *C* - типа вариант (по умолчанию), переменная *Слово* - строкового типа.

2.2.2. Описание констант

Данные, не изменяющиеся внутри программы можно считать константами. Их можно описать следующим образом:

Const *ИМЯ_КОНСТАНТЫ* **As** *ИМЯ_ТИПА=ПОСТОЯННОЕ_ВЫРАЖЕНИЕ*

Например.

Const Pi As Double = 3.141593

2.2.3. Описание массивов

Для хранения векторов, матриц и т.д. можно использовать массивы.



Массив - это структурированный тип данных, который представляет собой последовательность ячеек памяти, имеющих общее имя и хранящих данные одного типа. Каждый элемент массива определяется индексом (номером). Количество элементов в массиве называется **размерностью массива**. Массив описывается следующей конструкцией:

Dim ИМЯ_МАССИВА(СПИСОК_РАЗМЕРНОСТЕЙ) As ИМЯ_ТИПА

В списке размерностей массива каждое измерение отделяется запятой и определяется заданием нижней и верхней границ изменения индексов.

Например.

Dim X(1 TO 5) As Integer, Y(1 To 10, 1 To 20) As Double

Здесь *X* - одномерный массив, состоящий из 5 элементов целого типа, *Y* - двумерный массив, у которого 10 строк и 20 столбцов с элементами числового типа двойной точности.

2.3. Выражения

Выражения устанавливают порядок выполнения действий над элементами данных. Выражения состоят из операндов и знаков операций. Операндами являются константы, переменные, указатели функций, выражения, взятые в скобки.

2.3.1. Виды операций

Операции бывают арифметические, отношения и логические:

– **арифметические операции:**

- ^ возведение в степень,
- * умножение,
- / деление,
- \ деление нацело,
- mod остаток от деления,
- + плюс,
- минус;



– **операции отношения:**

- < меньше,
- > больше,
- <= меньше или равно,
- >= больше или равно,
- = равно,
- <> не равно;

– **логические операции:**

- Not логическое отрицание,
- And логическое "И",
- Or логическое "ИЛИ".

Результатом логической операции может быть одно из двух значений: True ("истина") или False ("ложь").

2.3.2. Приоритет выполнения операций

Если выражение содержит несколько операций, то приоритет их выполнения следующий:

1. Сначала выполняются арифметические операции в таком порядке, как они представлены в таблице 2.2.

Таблица 2.2. Приоритет арифметических операций

Описание операции	Обозначение в VBA
Унарный минус (изменение знака)	- (в начале выражения)
Возведение в степень	^
Умножение и деление	*, /
Деление нацело и остаток от деления	\, Mod
Сложение и вычитание	+, -

2. Далее операции отношения (они имеют одинаковый приоритет).
3. Последними выполняются логические операции в таком порядке, как они представлены в таблице 2.3.



Таблица 2.3. Приоритет логических операций

Описание операции	Обозначение в VBA
Логическое отрицание	Not
Логическое "И"	And
Логическое "ИЛИ"	Or

Если выражение содержит несколько операций одинакового приоритета, то порядок их выполнения слева направо. Чтобы изменить порядок действий в выражении используются круглые скобки.

Выражения бывают арифметические, отношения и логические.

Арифметические выражения записываются с помощью операндов числовых типов и арифметических операций, а результатом является числовое значение. В арифметическом выражении можно использовать стандартные математические функции, которые приведены в таблице 2.4.

Таблица 2.4. Стандартные математические функции VBA

Математическая запись	Имя функции в VBA	Описание
$ X $	Abs(число)	Возвращает значение, тип которого совпадает с типом переданного аргумента, равное абсолютному значению указанного числа.
$\arctg X$	Atn(число)	Возвращает значение типа Double, содержащее арктангенс числа.
$\cos X$	Cos(число)	Возвращает значение типа Double, содержащее косинус угла.
$]X[$	Int(число)	Возвращает значение типа, совпадающего с типом аргумента, которое содержит целую часть числа.
$\ln X$	Log(число)	Возвращает значение типа Double, содержащее натуральный логарифм числа.
e^X	Exp(число)	Возвращает значение типа Double, содержащее результат возведения числа e (основание натуральных логарифмов) в указанную степень.
$\text{Sign } X$	Sgn(число)	Возвращает значение типа Variant (Integer), соответствующее знаку указанного числа.
$\sin X$	Sin(число)	Возвращает значение типа Double, содержащее синус угла.
\sqrt{X}	Sqr(число)	Возвращает значение типа Double, содержащее квадратный корень указанного числа.
$\text{tg } X$	Tan(число)	Возвращает значение типа Double, содержащее тангенс угла.



Выражения отношения определяют истинность или ложность результата при сравнении двух операндов. Сравнить можно данные любого одинакового типа. Результат операции отношения только логический: True - "истина" или False - "ложь".

Логические выражения. Результатом логического выражения является логическое значение True или False. Простейшими видами логических выражений являются: логическая константа, логическая переменная, логическая функция, выражение отношения. Логические операции выполняются только над операндами логического типа.

Пример. Записать $1 \leq X \leq 5$ и определить значение выражения при $X=3$.

Выражение в VBA будет выглядеть так:

$X >= 1$ And $X <= 5$

Результатом выражения будет *True*.

Чтобы получить перечень всех математических функций, достаточно набрать имя любой известной математической функции (например, *SIN*), а затем нажать клавишу **F1** и ниже описания выбранной функции выбрать ссылку на **Математические функции**. В полученном перечне можно получить справку о назначении любой из встроенных математических функций и ее аргументе.

Чтобы получить перечень всех производных математических функций и правила их формирования, достаточно набрать имя любой известной математической функции (например, *SIN*), а затем нажать клавишу **F1** и ниже описания выбранной функции выбрать ссылку на **Производные математические функции**.

Ниже, в таблице 2.5, приведен список функций, которые могут быть получены с помощью встроенных математических функций.



Таблица 2.5. Производные математические функции

Математическая запись	Название функции	Комбинация встроенных функций
sc X	Секанс	$1/\text{Cos}(X)$
csc X	Косеканс	$1/\text{Sin}(X)$
ctg X	Котангенс	$1/\text{Tan}(X)$
arcsin X	Арксинус	$\text{Atn}(X/\text{Sqr}(-X*X+1))$
arccos X	Арккосинус	$\text{Atn}(-X/\text{Sqr}(-X*X+1))+2*\text{Atn}(1)$
arcsec X	Арксеканс	$\text{Atn}(X/\text{Sqr}(X*X-1))+\text{Sgn}(X-1)*2*\text{Atn}(1)$
arcsc X	Арккосеканс	$\text{Atn}(X/\text{Sqr}(X*X-1))+(\text{Sgn}(X)-1)*2*\text{Atn}(1)$
arcctg X	Арккотангенс	$\text{Atn}(X)+2*\text{Atn}(1)$
sh X	Гиперболический синус	$(\text{Exp}(X)-\text{Exp}(-X))/2$
ch X	Гиперболический косинус	$(\text{Exp}(X)+\text{Exp}(-X))/2$
th X	Гиперболический тангенс	$(\text{Exp}(X)-\text{Exp}(-X))/(\text{Exp}(X)+\text{Exp}(-X))$
sch X	Гиперболический секанс	$2/(\text{Exp}(X)+\text{Exp}(-X))$
csch X	Гиперболический косеканс	$2/(\text{Exp}(X)-\text{Exp}(-X))$
cth X	Гиперболический котангенс	$(\text{Exp}(X)+\text{Exp}(-X))/(\text{Exp}(X)-\text{Exp}(-X))$
arsh X	Гиперболический арксинус	$\text{Log}(X+\text{Sqr}(X*X+1))$
arch X	Гиперболический арккосинус	$\text{Log}(X+\text{Sqr}(X*X-1))$
arth X	Гиперболический арктангенс	$\text{Log}((1+X)/(1-X))/2$
arsch X	Гиперболический арксеканс	$\text{Log}((\text{Sqr}(-X*X+1)+1)/X)$
arsch X	Гиперболический арккосеканс	$\text{Log}((\text{Sgn}(X)*\text{Sqr}(X*X+1)+1)/X)$
arcth X	Гиперболический арккотангенс	$\text{Log}((X+1)/(X-1))/2$
$\log_n X$	Логарифм по основанию N	$\text{Log}(X)/\text{Log}(N)$



3. ОСНОВНЫЕ ОПЕРАТОРЫ ЯЗЫКА VBA

3.1. Правила записи операторов

При записи операторов необходимо придерживаться следующих правил:

- Каждый новый оператор записывается с новой строки.
- Чтобы записать несколько операторов на одной строке, их разделяют между собой двоеточием (:).
- Если оператор не помещается в одной строке, то необходимо поставить в конце строки пробел и знак подчеркивания (_), а затем продолжить не поместившуюся часть на следующей строке.

3.2. Оператор присваивания

Оператор присваивания используется, если какой-то переменной нужно присвоить новое значение. Он имеет следующий синтаксис:

ИМЯ_ПЕРЕМЕННОЙ = ВЫРАЖЕНИЕ

Сначала вычисляется выражение в правой части, а затем результат присваивается переменной, стоящей в левой части.

Например. Записать с помощью оператора присваивания следующее ма-

тематическое выражение:
$$y = \sqrt[3]{a} + \frac{a^2 + e^{-b}}{\sin^2 a - \ln b}$$

На VBA это выражение можно записать в виде следующего оператора:

$$Y = a^{(1/3)} + (a^2 + Exp(-b)) / (Sin(a)^2 - Log(b))$$

3.3. Операторы ввода-вывода

3.3.1. Оператор и функция MsgBox

Оператор **MsgBox** осуществляет вывод информации в диалоговом окне и устанавливает режим ожидания нажатия кнопки пользователем.



Он имеет следующий синтаксис:

MsgBox *СООБЩЕНИЕ*[,*КНОПКИ*][,*ЗАГОЛОВОК*]

Аргументы:

СООБЩЕНИЕ - обязательный аргумент, задающий в окне выводимое информационное сообщение. Может состоять из нескольких текстовых строк, объединенных знаком **&**. Использование в этом аргументе **Chr(13)** приводит к переходу на новую строку при выводе информации.

КНОПКИ - значение этого аргумента определяет категории появляющихся в окне кнопок. От значения аргумента кнопки зависят также, появляется ли в окне какой-либо значок. Если не указано, какие кнопки необходимо отображать в окне сообщений, то используется значение по умолчанию, соответствующее кнопке ОК. В табл. 3.1 приведены возможные комбинации кнопок и значков в окне сообщений.

ЗАГОЛОВОК - задает заголовок окна.

Функция **MsgBox** возвращает значение типа Integer, указывающее, какая кнопка была нажата в диалоговом окне.

Таблица 3.1. Допустимые значения переменной кнопки

Отображение	Аргумент
Кнопка ОК	VbOKOnly
Кнопки ОК и Отмена	VbOKCancel
Кнопки Да и Нет	VbYesNo
Кнопки Да, Нет и Отмена	VbYesNoCancel
Кнопки Прекратить, Повторить и Игнорировать	VbAbortRetryIgnore
Кнопки Повторить и Отмена.	VbRetryCancel
Информационный знак	VbInformation
Знак 	VbCritical
Знак вопроса	VbQuestion
Знак восклицания	VbExclamation



Например. Вывести сообщение о текущей дате.

Для этого можно использовать следующий оператор:

MsgBox "Сегодня на календаре" & Date , , "Внимание"

В результате будет выведено следующее окно (рис.3.1).

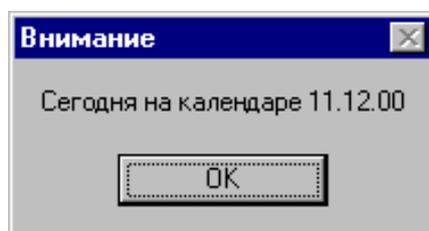


Рис. 3.1

После щелчка по кнопке ОК окно сообщения закроется, и выполнение программы возобновится с оператора, стоящего непосредственно за вызовом MsgBox.

3.3.2. Функция InputBox

Функция **InputBox** осуществляет ввод значений переменных с помощью окна ввода и имеет следующий синтаксис:

ИМЯ_ПЕРЕМЕННОЙ = InputBox(СООБЩЕНИЕ[, ЗАГОЛОВОК])

Аргументы:

СООБЩЕНИЕ - обязательный аргумент. Задаёт в окне информационное сообщение, обычно поясняющее смысл вводимой величины

ЗАГОЛОВОК - задаёт заголовок окна.

Например. Ввести значение переменной N с клавиатуры, предусмотрев значение по умолчанию равное 10.

Для этого можно использовать следующий оператор:

N = InputBox("Введите N", "Ввод исходных данных", 10)

В результате будет выведено следующее окно для ввода значения переменной N (рис.3.2).

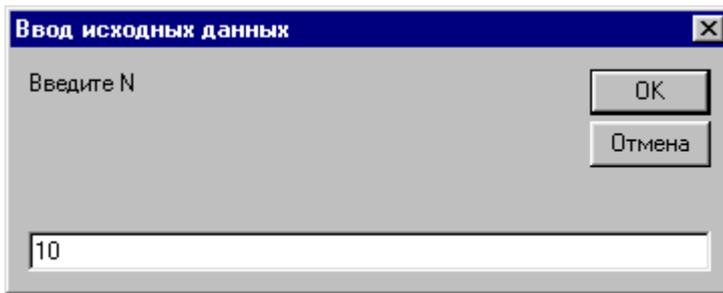


Рис. 3.2

Если значение по умолчанию подходит пользователю, то после щелчка кнопки ОК окно ввода закроется, переменной N присвоится значение 10 и выполнение программы возобновится с оператора, стоящего непосредственно за вызовом **InputBox**.

Если же значение по умолчанию не подходит пользователю, то перед щелчком по кнопке ОК необходимо ввести нужное значение переменной N.

3.4. Условный оператор IF

Для реализации разветвляющегося вычислительного процесса в VBA используется оператор **If...Then...Else**, который представляет собой простейшую форму проверки условий. Он имеет следующий синтаксис:

If УСЛОВИЕ Then ОПЕРАТОР_1 Else ОПЕРАТОР_2

ОПЕРАТОР_1 выполняется, если *УСЛОВИЕ* истинно, в противном случае выполняется *ОПЕРАТОР_2*. При этом оператор If...Then...Else записывается в одну строку.

УСЛОВИЕ – это выражение логического типа. Результат выражения всегда имеет булевский тип. Выражение может быть простым и сложным. При записи простых условий могут использоваться все возможные операции отношения, указанные в табл. 3.2.



Таблица 3.2. Логические отношения

Операция	Название	Выражение	Результат
=	Равно	$A = B$	True, если A равно B
\neq	Не равно	$A \neq B$	True, если A не равно B
>	Больше	$A > B$	True, если A больше B
<	Меньше	$A < B$	True, если A меньше B
\geq	Больше или равно	$A \geq B$	True, если A больше или равно B
\leq	Меньше или равно	$A \leq B$	True, если A меньше или равно B

Сложные условия образуются из простых путем применения логических операций и круглых скобок. Список логических операций приведен в табл. 3.3.

Таблица 3.3. Логические операции

	Название	Выражение	A	B	Результат
Not	Логическое отрицание	Not A	False		True
			True		False
And	Логическое И	A And B	True	True	True
			True	False	False
			False	True	False
			False	False	False
Or	Логическое ИЛИ	A Or B	True	True	True
			True	False	True
			False	True	True
			False	False	False

В условном операторе допустимо использование блока операторов вместо любого из операторов. В этом случае условный оператор имеет вид:

```

If УСЛОВИЕ Then
    БЛОК_ОПЕРАТОРОВ_1
Else
    БЛОК_ОПЕРАТОРОВ_2
End If

```



В условном операторе может проверяться несколько условий. В этом случае условный оператор имеет вид:

```

If УСЛОВИЕ_1 Then
    БЛОК_ОПЕРАТОРОВ_1
ElseIf УСЛОВИЕ_2 Then
    БЛОК_ОПЕРАТОРОВ_2
Else
    ....
End If

```

Пример 1. Написать часть программы для алгоритма на рис. 3.3.

```

If  $x > 0$  Then
     $y = 2 * x$ 
     $z = 1$ 
Else
     $y = x - 5$ 
     $z = 0$ 
End If

```

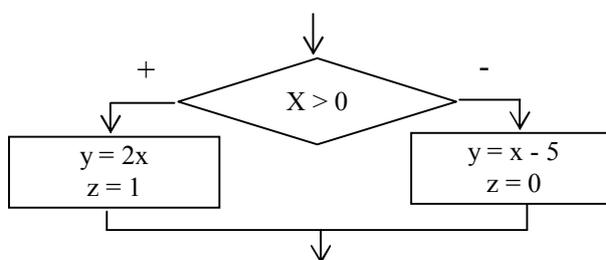


Рис. 3.3

Пример 2. Написать часть программы для алгоритма на рис. 3.4.

```

If  $x > 0$  Then
     $y = 2 * x$ 
Else
    If  $x = 0$  Then
         $y = x - 5$ 
    Else
         $y = x^2$ 
    End If
End If

```

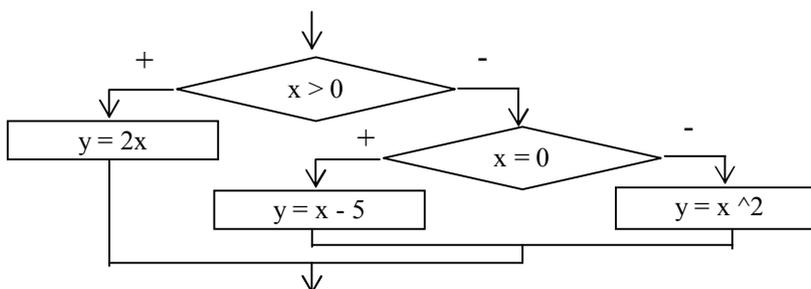


Рис. 3.4



3.5. Оператор выбора **Select Case**

Оператор **Select Case** удобно использовать, когда в зависимости от значения некоторого выражения, имеющего конечное множество допустимых значений, необходимо выполнить разные действия. Он также относится к условным операторам, но имеет другой вид:

Select Case *ПРОВЕРЯЕМОЕ_ВЫРАЖЕНИЕ*

Case *ЗНАЧЕНИЯ_1*

ОПЕРАТОРЫ_1

Case *ЗНАЧЕНИЯ_2*

ОПЕРАТОРЫ_2

...

Case *ЗНАЧЕНИЯ_N*

ОПЕРАТОРЫ_N

[Case Else

ИНАЧЕ_ОПЕРАТОРЫ]

End Select

ПРОВЕРЯЕМОЕ_ВЫРАЖЕНИЕ может иметь любой скалярный тип, кроме вещественного. *ЗНАЧЕНИЯ* состоят из произвольного количества значений или диапазонов, отделенных друг от друга запятыми.

Тип *ЗНАЧЕНИЙ* должен совпадать с типом *ПРОВЕРЯЕМОГО_ВЫРАЖЕНИЯ*.

Сначала вычисляется *ПРОВЕРЯЕМОЕ_ВЫРАЖЕНИЕ*. Если его значение совпадает с одним из значений *ЗНАЧЕНИЯ_1*, то выполняются *ОПЕРАТОРЫ_1* и управление передается оператору, стоящему после **End Select**. Если его значение не совпадает ни с одним из значений *ЗНАЧЕНИЯ_1*, то выполняются *ИНАЧЕ_ОПЕРАТОРЫ* и управление передается оператору, стоящему после

End Select

Например. Написать часть программы для алгоритма на рис. 3.5, определяющего значение переменной S в зависимости от значения переменной n.



Dim n As integer

Dim s As String

n=1997

Select Case n

Case 1960

s = "Год рождения"

Case 1961 to 1966

s = "Дошкольные годы"

Case 1967 to 1977, 1981 to 1985

s = "Годы учёбы"

Case Else

s = "Годы работы"

End Select

' Результат s = "Годы работы"

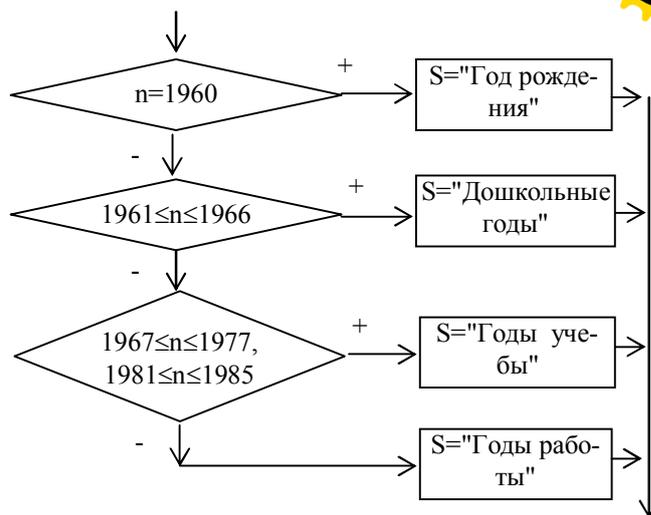


Рис. 3.5

3.6. Операторы цикла

Для реализации циклического вычислительного процесса, т. е. многократного выполнения одного или нескольких операторов, служит оператор цикла **For...Next**, который имеет следующий синтаксис:

```

For СЧЕТЧИК=НАЧ_ЗНАЧЕНИЕ To КОН_ЗНАЧЕНИЕ Step ШАГ
    БЛОК_ОПЕРАТОРОВ
[Exit For]
    БЛОК_ОПЕРАТОРОВ
Next СЧЕТЧИК
  
```

Цикл **For...Next** перебирает значения переменной *СЧЕТЧИК*, которая является параметром цикла, от начального до конечного значения с указанным шагом изменения. При этом обеспечивается выполнение блока операторов тела цикла при каждом новом значении счетчика. Если **Step ШАГ** в конструкции от-



сутствует, то по умолчанию считается, что шаг равен 1. По оператору **Exit For** можно выйти из оператора цикла до того, как *СЧЕТЧИК* достигнет последнего значения.*

Для перебора объектов из группы подобных объектов, например, ячеек из диапазона или элементов массива, удобно использовать оператор цикла **For... Each...Next**.

For Each *ЭЛЕМЕНТ In ГРУППА*

БЛОК_ОПЕРАТОРОВ

[Exit For]

БЛОК_ОПЕРАТОРОВ

Next *ЭЛЕМЕНТ*

В VBA для организации циклов с неизвестным заранее числом повторений используются и другие операторы цикла:

циклы с предусловием – **Do While ... Loop,**

Do Until ... Loop;

циклы с постусловием – **Do ... Loop While,**

Do ... Loop Until.

Ниже приведен синтаксис этих операторов цикла:

' Цикл с предусловием **Do While ... Loop**

Do While *УСЛОВИЕ*

БЛОК_ОПЕРАТОРОВ

[Exit Do]

БЛОК_ОПЕРАТОРОВ

Loop

* **Примечание.** Не рекомендуется принудительно изменять значения параметра цикла, его начального и конечного значения в теле цикла For...Next.



' Цикл с предусловием **Do Until ... Loop**

Do Until УСЛОВИЕ

БЛОК_ОПЕРАТОРОВ

[Exit Do]

БЛОК_ОПЕРАТОРОВ

Loop

' Цикл с постусловием **Do ... Loop While**

Do

БЛОК_ОПЕРАТОРОВ

[Exit Do]

БЛОК_ОПЕРАТОРОВ

Loop While УСЛОВИЕ

' Цикл с постусловием **Do ... Loop Until**

Do

БЛОК_ОПЕРАТОРОВ

[Exit For]

БЛОК_ОПЕРАТОРОВ

Loop Until УСЛОВИЕ

Оператор **Do While...Loop** обеспечивает многократное повторение блока операторов до тех пор, пока *УСЛОВИЕ* соблюдается, а оператор **Do Until...Loop** пока *УСЛОВИЕ* не соблюдается. Операторы **Do...Loop While**, **Do...Loop Until** отличаются от перечисленных выше операторов тем, что сначала блок операторов выполняется по крайней мере один раз, а потом проверяется *УСЛОВИЕ*.

Для избежания заикливания в теле цикла должен быть хотя бы один оператор, который изменяет значения переменных, стоящих в *УСЛОВИИ*.

Оператор **Exit Do** обеспечивает досрочный выход из оператора цикла.



Пример 1. Составить фрагмент программы для алгоритма на рис. 3.6.

```

S = 0
P = 1
For i = 1 to 5
    S = S + 2
    P = P * S
Next i

```

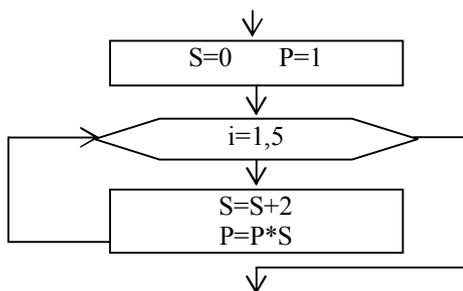


Рис. 3.6

Пример 2. Составить фрагмент программы для алгоритма на рис. 3.7.

```

k = 0
f = 1
Do While f <= 15
    k = k + 1
    f = f + 2
Loop

```

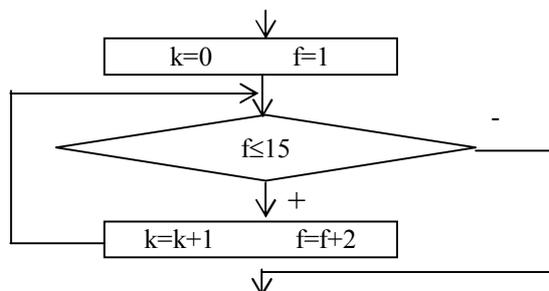


Рис. 3.7

Пример 3. Составить фрагмент программы, соответствующей алгоритму на рис.3.8.

```

D = 1
S = 0
Do
    S = S + D
    D = D + 1
Loop Until D >= 10

```

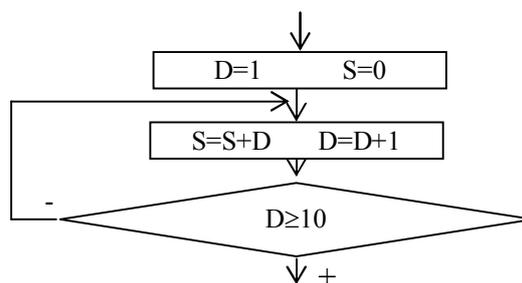


Рис. 3.8



4. СТРУКТУРА ПРОГРАММЫ. МОДУЛИ, ПРОЦЕДУРЫ И ФУНКЦИИ

Модуль представляет собой текстовый ASCII-файл с программным кодом, содержащим подпрограммы, переменные и константы. Проект может состоять из множества программных модулей. Для их создания необходимо выполнить команду **Вставка** → **Модуль**. Рабочее окно модуля представлено на рис. 4.1.

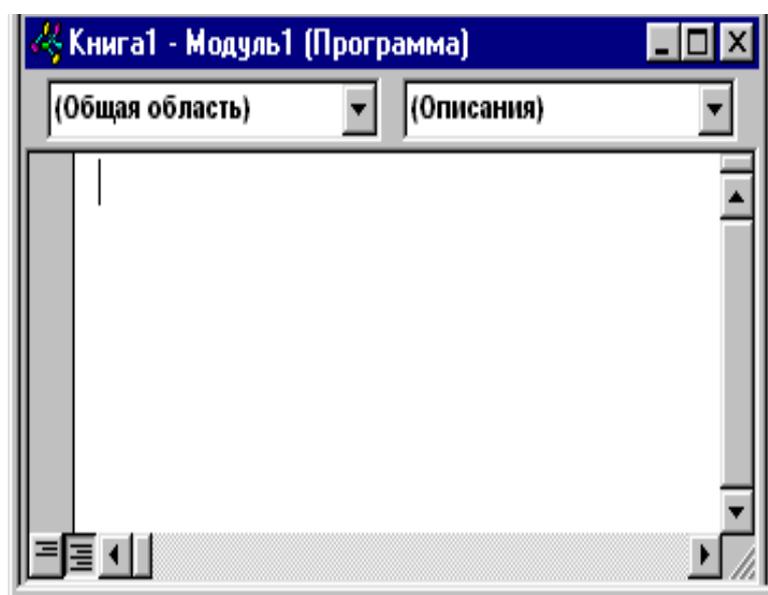


Рис. 4.1. Окно модуля

Основу программ в VBA составляют процедуры и функции.

Процедура Sub— это обособленная совокупность операторов VBA, выполняющая определенные действия. В общем случае процедура принимает некоторые параметры (переменные, которые передаются процедуре в качестве исходных данных), выполняет программу и может возвращать результирующие значения, которые присваиваются параметрам внутри процедуры. Однако чаще используются процедуры без параметров. Например, процедуры, выполняющиеся при возникновении определенных событий. Вложенность процедур в другие процедуры не допускается. Структура процедуры следующая:



[ДОСТУП] Sub ИМЯ_ПРОЦЕДУРЫ ([СПИСОК_ПАРАМЕТРОВ])

ТЕЛО_ПРОЦЕДУРЫ

End Sub

Ключевое слово *ДОСТУП* является необязательным и определяет область видимости процедуры. **Public** указывает, что процедура доступна для всех других процедур во всех модулях (глобальная). **Private** указывает, что процедура доступна для других процедур только того модуля, в котором она описана (локальная). *СПИСОК_ПАРАМЕТРОВ* также является необязательным элементом и позволяет передавать процедуре различные исходные данные при вызове, которые называются формальными параметрами. При этом ключевое слово **Dim** не указывается. *ТЕЛО_ПРОЦЕДУРЫ* состоит из описательной части и блока операторов, выполняющихся один за другим. Если необходимо прекратить выполнение процедуры в некотором конкретном месте, это можно сделать с помощью оператора **Exit Sub**. *ИМЯ_ПРОЦЕДУРЫ* – это любой идентификатор, определенный пользователем. **Идентификатор** – это последовательность букв, цифр и символа подчеркивания, начинающаяся с буквы (пробелы внутри идентификатора недопустимы). Имя процедуры всегда определяется на уровне модуля. Для использования процедуры в тексте программы (т.е. для её вызова), необходимо указать имя процедуры и список фактических параметров, которые должны по типу и порядку следования совпадать с формальными параметрами.

Функция Function во многом похожа на процедуру, но в отличие от неё при вызове всегда возвращает значение. Функция получает параметры, называемые **аргументами**, и выполняет с ними некоторые действия, результат которых возвращается функцией. Структура функции следующая:

[ДОСТУП] Function ИМЯ_ФУНКЦИИ(СПИСОК_АРГУМЕНТОВ) As ТИП

ТЕЛО_ФУНКЦИИ

ИМЯ_ФУНКЦИИ = ВЫРАЖЕНИЕ

End Function

ТИП определяет тип данных возвращаемого результата. В теле функции



Обязательно должен присутствовать, по крайней мере, один оператор, присваивающий имени функции значение вычисляемого выражения. Досрочное завершение функции возможно с помощью оператора **Exit Function**. В программе вызов функции осуществляется с помощью оператора присваивания, в правой части которого указывается имя функции с перечнем фактических параметров, как и любой другой встроенной функции, например, Sqr, Cos или Chr.

Процедуры и функции, не описанные явно с помощью ключевых слов Public или Private, по умолчанию являются общими.

Для быстрого добавления в модуль подпрограмм удобно воспользоваться командой **Вставка → Процедура**. В появившемся окне (рис. 4.2) нужно выбрать необходимые опции.

В MS Excel с функциями, созданными пользователем, можно работать с помощью Мастера функций точно так же, как и со встроенными функциями рабочего листа.

Пример. В MS Excel создать функцию пользователя, математически определенную следующим образом:

$$y = \sin(x) \cdot e^{-5x}$$

Создадим модуль, как указано выше (рис 4.2), и введем в него текст следующей программы:

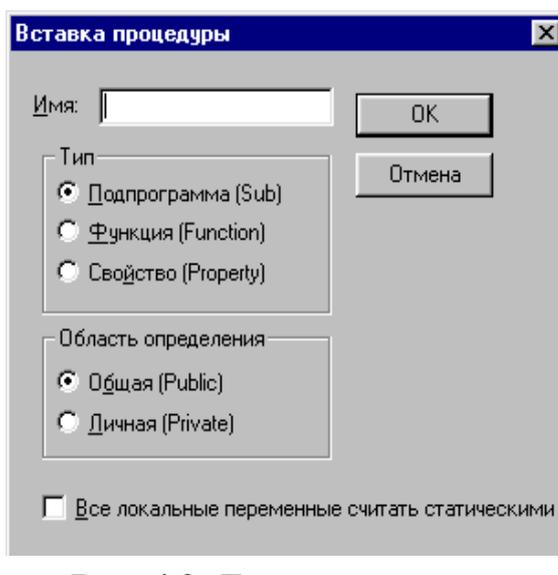
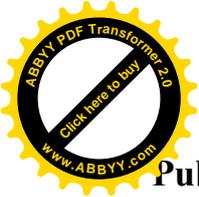


Рис. 4.2. Диалоговое окно “Вставка процедуры”



Public Function Y(x As Single) As Single

$$Y = \text{Sin}(x) * \text{Exp}(-5 * x)$$

End Function

Для использования созданной функции на рабочем листе MS Excel введем в ячейку A2 число 0.1. В ячейке B2 вычислим значение функции Y при x = 0.1. Для этого в ячейку B2 достаточно ввести формулу =Y(A2). Это можно сделать и с помощью Мастера функций, который будет содержать функцию Y наряду с другими встроенными функциями MS Excel (рис 4.3).

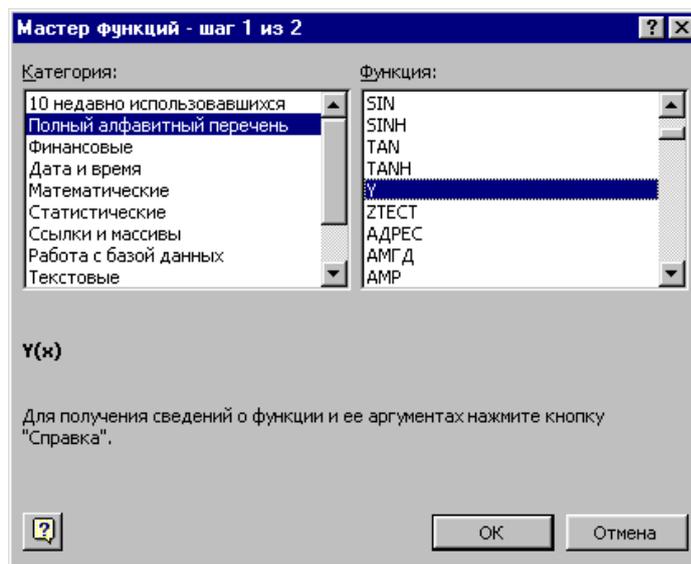


Рис. 4.3. Вызов функции Y()

Общие принципы организации программ VBA в модуле следующие. Обычно текст программы начинается с опций, которые управляют описанием переменных, способом сравнения строк и т. д.

Затем следует объявление **глобальных** для данного модуля переменных и констант, т.е. таких, которые используются во всех процедурах модуля.

Далее располагают непосредственно текст функций и процедур, составляющих саму программу.

Разделителем операторов в одной строке при записи программы является символ “:”.



Для переноса оператора на другую строку используется символ “_” (знак подчеркивания).

Иногда внутри программы удобно помещать **комментарии** – пояснительный текст, который игнорируется компилятором и может быть записан в любом месте программы. Комментарии удобно также использовать при отладке программы для временного отключения операторов. Каждая строка комментариев начинается со знака апострофа.

Пример организации 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
```

Здесь инструкция **Option Explicit** указывает на необходимость описания типов переменных, используемых на данном листе модуля.

Инструкция **Option Base 1** указывает, что индексация элементов массива будет начинаться с 1.

Функция **Sqr** вычисляет квадратный корень аргумента.

Функция **CStr** переводит числовой формат в строковый.



5. СОЗДАНИЕ ФОРМ В VBA И ВКЛЮЧЕНИЕ ИХ В ПРОЕКТЫ

5.1. Создание форм. Свойства, события и методы форм

Форма – это главный объект, образующий визуальную основу приложения. По своей сути форма представляет собой окно, в котором можно размещать различные управляющие элементы при создании приложений. Для создания формы необходимо выполнить команду **Вставка** → **UserForm**. В окне Конструктора форм появится форма, имеющая стандартный вид для ОС Windows (рис. 5.1).

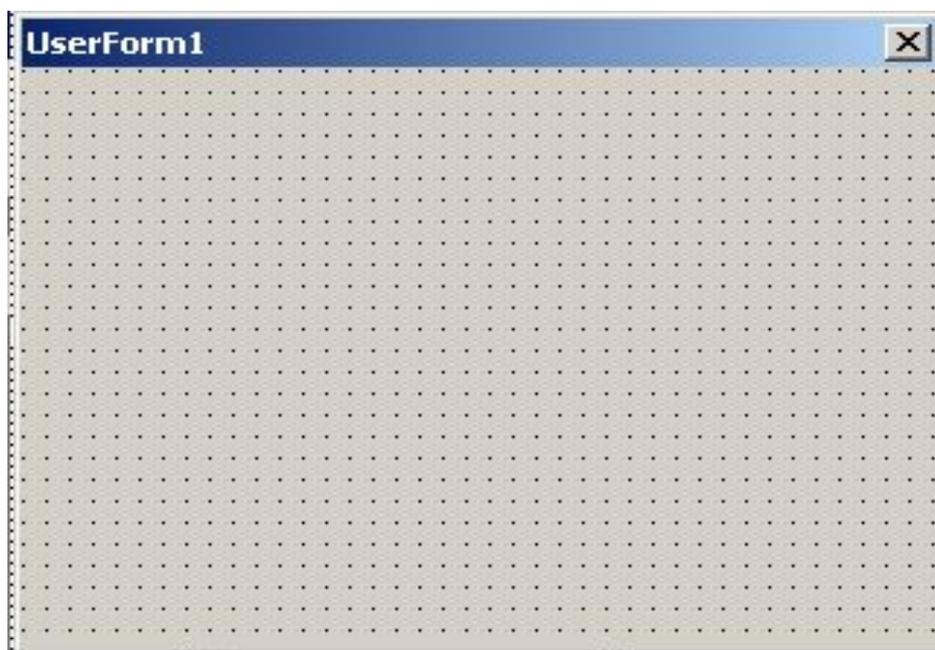


Рис. 5.1. Объект форма

Как и любой другой объект VBA форма имеет набор **свойств**, основные из которых приведены в таблице 5.1. Для получения справки по любому свойству достаточно выделить его в окне свойств и нажать F1.



Таблица 5.1. Основные свойства формы

Свойство	Описание
<i>BackColor</i>	Цвет фона для формы.
<i>BorderStyle</i>	Определяет тип границы, окружающей форму
<i>Caption</i>	Текст, который выводится в заголовке формы.
<i>Font</i>	Определяет тип и вид шрифта в форме.
<i>Height</i>	Определяет высоту формы в твипах.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Width</i>	Определяет ширину формы в твипах.

Свойства можно изменять в режиме конструирования в окне свойств, либо программно в режиме выполнения. Например, в ходе выполнения программы можно изменить заголовок формы командой:

```
frmForm1.Caption = "Привет"
```

Программы в ОС Windows управляются **событиями**. Каждый раз, когда нажимается кнопка, перемещается мышь, изменяются размеры формы и т.д., ОС генерирует сообщение. Сообщение доставляется соответствующему объекту, например форме, а та генерирует соответствующее событие. Следовательно, можно составить фрагмент программы, в котором объект будет реагировать на событие определенным образом, т.е. любому стандартному событию соответствует определенная процедура. Чтобы просмотреть события связанные с формой, необходимо в режиме конструирования дважды щелкнуть на ней – появится окно программы, в котором щелкнуть на списке Процедура. В табл. 5.2 приведены наиболее часто используемые события.

Таблица 5.2. Основные события форм

Событие	Описание
<i>Initialize</i>	Происходит во время конфигурации и до загрузки формы в память.
<i>Activate</i>	Происходит после загрузки формы в память.
<i>Deactivate</i>	Происходит, если форма перестает быть активной.
<i>Click</i>	Происходит при нажатии левой кнопки мыши на форме.



Следующий пример изменяет заголовок формы при активизации, и уменьшает размер формы после щелчка левой кнопкой мыши на форме.

```
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
```

Также форма обладает набором методов и инструкций. **Метод** определяет действие, которое может быть выполнено с объектом. **Инструкция** инициирует действие. Она может выполнить метод или функцию, В табл. 5.3 и 5.4 приведены наиболее часто используемые методы и инструкции для работы формами.

Таблица 5.3. Основные методы форм

Метод	Описание
<i>Hide</i>	Скрывает объект UserForm, но не выгружает его.
<i>Show</i>	Выводит на экран объект UserForm.

Таблица 5.4. Основные инструкции форм

Инструкция	Описание
<i>Load</i>	Загружает объект UserForm, но не отображает его на экране.
<i>Unload</i>	Удаляет объект UserForm из памяти.

В следующем примере предполагается, что в проекте созданы две формы frmForms. При запуске проекта происходит событие Initialize для формы frmForm1, форма frmForm2 загружается и выводится на экран. Когда при помощи мыши выбирается frmForm2, она делается невидимой, и появляется форма frmForm1. Если же выбирается frmForm1, frmForm2 появляется вновь.

'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. Выбор и использование управляющих элементов

Создание управляющих элементов на форме выполняется с помощью **Панели инструментов**, которая выводится на экран командой **Вид → Панель элементов** (рис. 5.2).

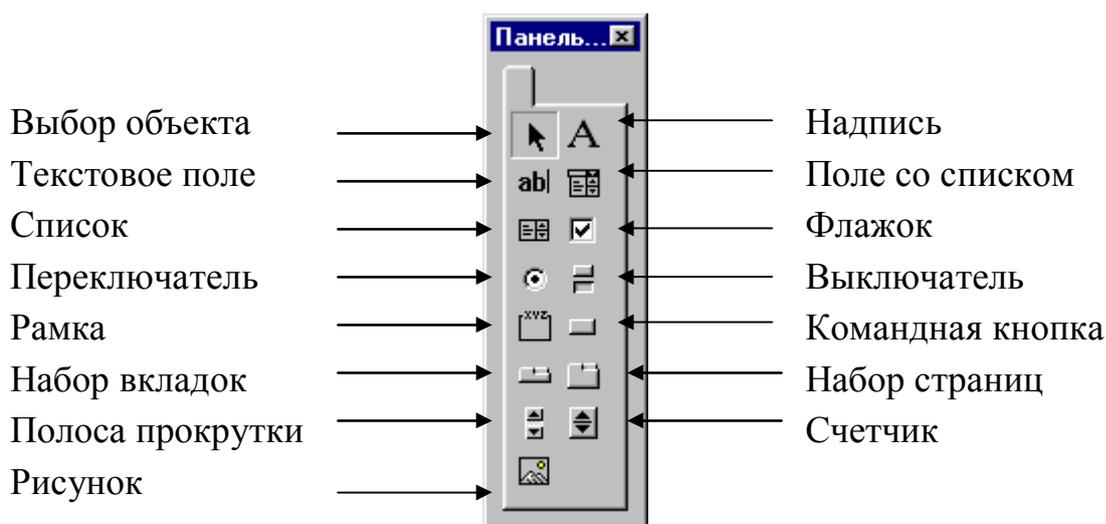


Рис. 5.2. Панель элементов VBA



С помощью кнопок этой панели можно поместить в форму необходимый элемент управления. Для этого нужно щелкнуть на значке элемента управления, далее при нажатой левой кнопке мыши определить размер и место расположения элемента в форме. Когда элемент на форме выделен (рамка объекта содержит маленькие прямоугольники) можно изменять его размеры и перемещать с помощью мыши, а также просматривать и изменять его свойства в окне свойств.

Каждый управляющий элемент (объект) характеризуется набором свойств (которые можно изменять в режимах конструирования или выполнения), событий и методов.

Для каждого объекта проекта необходимо определить его имя. В соответствии с общепринятыми соглашениями об именах объектов первые три символа имени должны отражать вид элемента, а остальные символы - назначение. В табл. 5.5. представлены сочетания первых трех символов для наиболее часто используемых элементов.

Таблица 5.5. Рекомендуемые сочетания первых трех символов имен

Объект	Первые 3 символа имени	Пример имени
Форма	frm	frmMyForm
Надпись	lbl	lblInfo
Текстовое поле	txt	txtInput
Командная кнопка	cmd	cmdExit
Флажок	chk	chkSound
Переключатель	opt	optLevel
Список	lsb	lsbTypes
Рамка	fra	fraChoices
Полоса прокрутки	vcb	vcbSpeed
Рисунок	pic	picChema

Командная кнопка является самым распространенным элементом управления, и может использоваться для организации выполнения вычислений



и других действий, вызова процедур и функций пользователя, открытия форм и т.д. Основные свойства командной кнопки представлены в табл. 5.6. В свойстве *Caption* можно ставить символ **&** перед буквой, которая будет использоваться в сочетании с клавишей **Alt** для ускоренного доступа к кнопке. Также можно перейти к кнопке клавишей **Tab**, а затем нажать **Enter**.

Таблица 5.6. Свойства командных кнопок

Свойство	Описание
<i>BackColor</i>	Цвет фона кнопки.
<i>Caption</i>	Текст, который выводится на кнопке.
<i>Enabled</i>	Значение False делает кнопку недоступной.
<i>Font</i>	Определяет тип и вид шрифта на кнопке.
<i>ForeColor</i>	Определяет цвет шрифта на кнопке.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Picture</i>	Добавляет рисунок на кнопку.
<i>PicturePosition</i>	Определяет расположение текста и рисунка на кнопке.
<i>Visible</i>	Значение False делает кнопку невидимой.

Основным **событием** кнопки является **Click**. Для написания программного кода, который будет выполняться при нажатии командной кнопки, достаточно два раза щелкнуть на ней левой кнопкой мыши в режиме конструирования проекта.

Наиболее полезным **методом** командной кнопки является **SetFocus**, позволяющий вернуться к кнопке (передать ей фокус). Например, следующая команда позволяет вернуться к кнопке по умолчанию после ввода данных в текстовое поле: **cmdMyButtum.SetFocus**

Текстовое поле применяется для ввода или вывода информации. Основные свойства текстового поля представлены в табл. 5.7.



Таблица 5.7. Свойства текстового поля

Свойство	Описание
<i>Enabled</i>	Значение False делает поле недоступным.
<i>Font</i>	Определяет тип и вид шрифта в текстовом поле.
<i>ForeColor</i>	Определяет цвет шрифта в текстовом поле.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>MaxLength</i>	Определяет количество вводимых символов в текстовое поле.
<i>PasswordChar</i>	Определяет символ, отображаемый при вводе в текстовое поле.
<i>Text</i>	Определяет содержимое текстового поля.

Например, для очистки содержимого текстового поля в ходе выполнения программы необходимо ввести в требуемом месте программного кода команду:

```
txtResult.Text=" "
```

Основным **событием** текстового поля является **Change**, происходящее при вводе или удалении символов. Например, команду `cmdMyButtun.SetFocus` можно поместить в процедуру события `Change` текстового поля.

Надпись применяется как самостоятельно для вывода справочной информации, так и в виде "подсказок" для текстового поля, списка или другого элемента. Главное её отличие от текстового поля в том, пользователь не может изменить текст надписи (хотя его можно изменить как свойство во время выполнения программы). Основные свойства надписи представлены в табл. 5.8.

Таблица 5.8. Свойства надписи

Свойство	Описание
<i>Caption</i>	Определяет текст, содержащийся в надписи.
<i>Font</i>	Определяет тип и вид шрифта надписи.
<i>ForeColor</i>	Определяет цвет шрифта надписи.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Picture</i>	Добавляет рисунок в надпись.
<i>PicturePosition</i>	Определяет расположение текста и рисунка надписи.



Список позволяет работать с перечнем из нескольких вариантов. Пользователь может просмотреть содержимое списка и выбрать один из вариантов для последующей обработки. Прямое редактирование содержимого списка невозможно. Если в списке помещаются не все строки, то автоматически добавляется вертикальная полоса прокрутки. Основные свойства списка представлены в табл. 5.9.

Таблица 5.9. Свойства списка

Свойство	Описание
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>ListIndex</i>	Возвращает номер текущей выделенной строки списка -1.
<i>Text</i>	Содержимое текущей выделенной строки списка.

Для списка чаще всего используются **события Click** и **DbClick** (двойной щелчок левой кнопкой мыши на одной из строк списка). Во втором случае пользователь одновременно выделяет строку и начинает ее обработку.

Работа со списком начинается с его заполнения **методом AddItem**, который может вызываться несколько раз подряд. Часто метод AddItem помещается в процедуру **UserForm_Initialize()**, чтобы список заполнялся при загрузке формы. Метод **RemoveItem** удаляет строки из списка. Метод **Clear** очищает сразу весь список. Следующий пример показывает, как работают списки, при этом предполагается, что в проекте создана форма с двумя списками (List1 и List2). Двойной щелчок на любой строке одного списка перемещает её в другой список. Строка включается в другой список до того, как она будет удалена из текущего.

```
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
```

Переключатели позволяют выбрать один вариант из группы. Обычно они группируются в рамках (см. далее), однако их можно располагать прямо на форме, если используется только одна группа переключателей. Основные свойства переключателя представлены в табл. 5.10.

Таблица 5.10. Свойства переключателя

Свойство	Описание
<i>Caption</i>	Задаёт текст, определяющий назначение переключателя.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Value</i>	Значение True указывает, что переключатель выбран.

Наиболее важным является свойство Value значение True (переключатель находится в установленном состоянии), которого в режиме конструирования задается только у одного переключателя в группе. В режиме выполнения это свойство чаще всего проверяется в процедуре события Click кнопки, нажатой после установки нужного переключателя, что позволяет проверить перед вызовом следующей процедуры некоторое условие. Однако определенные действия можно выполнять сразу же после выбора переключателя в процедуре его события Click.

Флажок частично аналогичен переключателю, но в отличие от него может использоваться как отдельный самостоятельный элемент. Даже объединен-

ные в группу флажки работают независимо друг от друга. Основные свойства флажков такие же, как и у переключателя (см. табл. 5.10). Однако свойство Value может принимать три значения (флажок находится в установленном состоянии, снятом или неопределенном).

Наиболее часто используемым событием флажков является **Click**, в процедуре которого можно проверять состояние флажка по свойству Value. Следующий пример иллюстрирует работу флажков, при этом предполагается, что в проекте создана форма с двумя флажками (ChkBold и ChkItalic) и текстовым полем TxtExam (рис.5.3). После ввода символов в текстовое поле, с помощью флажков можно делать текст полужирным или курсивом. Свойства **FontBold** и **FontItalic** текстового поля устанавливают способы начертания текста.

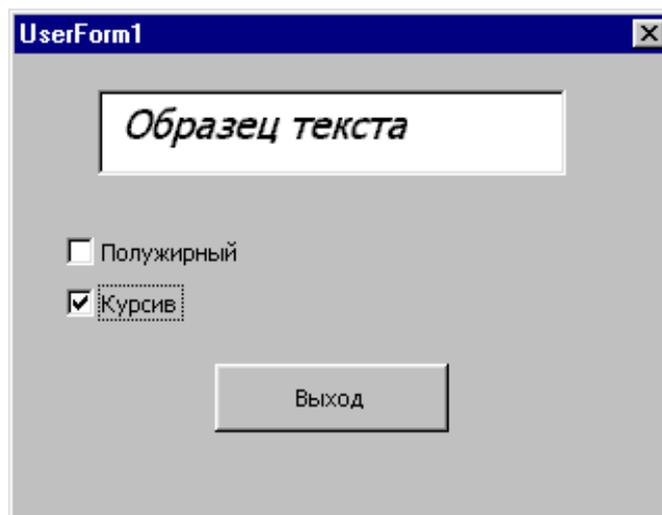


Рис. 5.3. Использование флажков

```
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 используется для группировки переключателей или флажков, и помещается на форму раньше элементов, находящихся внутри неё. Переключатели находящиеся внутри рамки, работают как самостоятельная группа и не влияют на состояние переключателей в других рамках. Основным свойством рамки является **Caption**, которое задает текст, определяющий назначение элементов в рамке.

Рисунок используется для простейшего вывода изображения на форме. Он может отображать растровые файлы (.BMP), значки (.ICO), метафайлы (WMF), а также файлы в формате JPEG (.JPG) и GIF (.GIF). Основные свойства рисунка представлены в табл. 5.11.

Таблица 5.11. Свойства рисунка

Свойство	Описание
<i>Autosize</i>	Значение True подгоняет размер элемента под размер содержимого.
<i>(Name)</i>	Имя объекта, для программы VBA.
<i>Picture</i>	Задаёт файл для рисунка.

События и методы рамок и рисунков практически не используются.



6. ЛАБОРАТОРНЫЙ ПРАКТИКУМ

6.1. Лабораторная работа № 1.

Тема: "Знакомство с интегрированной средой разработки VBA"

Цель работы: получить навыки работы в среде VBA.

Задание:

1. Загрузить приложение Excel.
2. На "Лист1" Excel занести исходные данные:

	A	B	C
1	a=	5	
2	b=	2	
3	n=	3	

3. Загрузить редактор VBA: **Сервис** → **Макрос** → **Редактор VBA**.
4. Создать новый модуль: **Вставка** → **Модуль**. В его окне написать текст программы:

```
Sub Prtm1()
```

```
Dim a As Single, b As Single, c As Single
```

```
Dim d As Single, i As Integer, n As Integer
```

```
Sheets("Лист1").Activate
```

```
a= Cells(1,2)
```

```
b= Cells(2,2)
```

```
n= Cells(3,2)
```

```
For i=1 To n
```

```
c=a*b/(a-b)*i
```

```
d=c*b/4
```

```
Cells(i+1,4)=i
```

```
Cells(i+1,5)=c
```

```
Cells(i+1,6)=d
```

```
Next i
```

```
Cells(1,4)="i"
```

```
Cells(1,5)="c"
```

```
Cells(1,6)="d"
```

```
End Sub
```

5. Сохранить модуль в книге с именем ОТЛАДКА в папке своей группы.
6. Закрыть окно VBA и Excel.
7. Загрузить приложение Excel и открыть книгу ОТЛАДКА, а затем окно Модуля1.
8. Запустить программу на выполнение.
9. Если есть ошибки, то исправить их и запустить программу повторно.
10. Переключиться на "Лист1" Excel, просмотреть результаты работы программы, а затем удалить результаты работы программы.
11. Вернуться в окно редактора VBA.
12. Осуществить пошаговое выполнение программы, просматривая при этом значения всех переменных.
13. Создать новый модуль, в который скопировать текст программы из Модуля1.
14. Изменить в тексте программы Модуля2 оператор *Sheets("Лист1").Activate* на оператор *Sheets("Лист2").Activate*
15. Переключиться на "Лист2" Excel и занести туда новые исходные данные для программы.
16. Вернуться в окно Модуля2 и запустить программу на выполнение.
17. Переключиться на "Лист2" Excel и просмотреть результаты работы программы.
18. Показать работу преподавателю.



6.2. Лабораторная работа №2.

Тема: "Создание экранных форм: работа с элементами управления"

Цель работы: научиться работать с элементами управления в среде VBA.

Задание:

1. Разработать блок-схему алгоритма и проект VBA с экранной формой (см. пример экранной формы) в соответствии с вариантом задания (варианты приведены ниже):
 - Войти в интегрированную среду разработки VBA: **Сервис** → **Макрос** → **Редактор VBA**.
 - Отобразить **Окно Проекта** : **Вид(View)** → **Project Explorer**
 - Отобразить **Окно Свойств** выделенного элемента управления : **Вид(View)** → **Properties Windows**
 - Добавить форму в проект : **Вставка(Insert)** → **USER FORM**.
 - Добавить нужные элементы управления на форму с помощью **Панели элементов (ToolBox)**.
 - Определить **Свойство (Name)** для текстовых полей (**TextBox**) и кнопок (**CommandButton**).
 - Определить **Свойство Caption** для формы пользователя (**User Form**), надписей (**Label**), командных кнопок (**CommandButton**).
2. Изменить шрифт надписей "Исходные данные" и "Результаты" (Свойства-**Font**).
3. Установить бирюзовый цвет фона формы пользователя (Свойства – **BackColor**).
4. Установить серый цвет фона кнопок (Свойства – **BackColor**).
5. Добавить рисунок в форму пользователя (**Свойства** → **Picture** → **Выбрать рисунок на свой вкус**).
6. Создать программный модуль в соответствии с вариантом задания:
 - Двойной щелчок по кнопке "**Вычисление**" → в окне редактора VBA ввести текст подпрограммы обработки щелчка по кнопке **CmdSolve**.

- Двойной щелчок по кнопке "Очистка" → в окне редактора VBA ввести текст подпрограммы обработки щелчка по кнопке
 - *CmdClear*.
 - Двойной щелчок по кнопке "Выход" → в окне редактора VBA ввести текст подпрограммы обработки щелчка по кнопке *CmdExit*.
7. Когда текст всех подпрограмм введен, выполнить **Запуск (Run Macro)** программы и если есть сообщения об ошибках компиляции – исправить их и вновь выполнить запуск программы.
 8. В режиме выполнения программы ввести исходные данные в соответствующие поля.
 9. Щелкнуть по кнопке "Вычисление".
 10. Щелкнуть по кнопке "Очистка".
 11. Ввести другие значения входных данных.
 12. Щелкнуть по кнопке "Выход".

Методические указания.

Рассмотрим пример экранной формы:

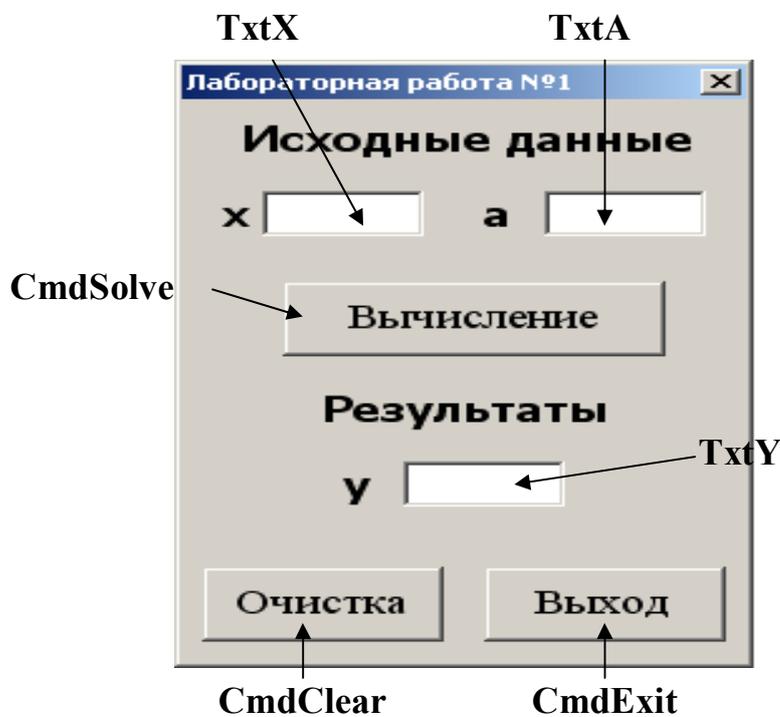


Рис. 6.1. Пример экранной формы



Таблица 6.1. Варианты заданий к лабораторной работе №2

<i>№ n/n</i>	<i>Модель</i>	<i>Исходные данные</i>	<i>Выводимые данные</i>
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>
1	$y = \begin{cases} ax + b, & \text{если } x > 10 - b^2 \\ ax - c , & \text{если } x = 10 - b^2 \\ cx, & \text{если } x < 10 - b^2 \end{cases}$ $x = \begin{cases} a^2/b^2, & \text{если } a \neq b \\ a + b, & \text{если } a = b \end{cases}$	$a = 3.2$ $b = 2.3$ $c = 4.5$	$x, y.$
2	$y = \begin{cases} x - c, & \text{если } x \leq a \\ x + c, & \text{если } a < x \leq b \\ x/c, & \text{если } x > b \end{cases}$ $c = \begin{cases} ax + b, & \text{если } x < 2 \\ a + bx, & \text{если } x \geq 2 \end{cases} \quad x = \sqrt{a^2 + b^2}$	$a = 1.2$ $b = 3.1$	$x, c, y.$
3	$y = \begin{cases} x^3 + 3, & \text{если } x > 3 \\ x^3 - 3, & \text{если } x < 3 \\ abx, & \text{если } x = 3 \end{cases}$ $x = \begin{cases} (a + 1)/(b - 1), & \text{если } a < b \\ (a - 1)(b + 1), & \text{если } a \geq b \end{cases}$	$a = 1.7$ $b = 2.4$	$x, y.$
4	$y = \begin{cases} x - ab, & \text{если } x < 4 \\ x + ab, & \text{если } 4 \leq x \leq 5 \\ (x + a)/b, & \text{если } x > 5 \end{cases}$ $x = \begin{cases} a + b, & \text{если } a < b \\ a - b, & \text{если } a \geq b \end{cases}$	$a = 3.5$ $b = 4.7$	$x, y.$

1	2	3	4
5	$y = \begin{cases} a + cx, & \text{если } x < 1 \\ b + d/x, & \text{если } 1 \leq x < 3 \\ c - ax, & \text{если } x \geq 3 \end{cases}$ $x = \begin{cases} \sqrt{ab}, & \text{если } ab \geq cd \\ \sqrt[3]{cd}, & \text{если } ab < cd \end{cases}$	$a = 2.5$ $b = 1.3$ $c = 1.5$ $d = 2.3$	$x, y.$
6	$z = \begin{cases} x + y, & \text{если } y^2 > x^2 + 1 \\ xy, & \text{если } y^2 = x^2 + 1 \\ x - y, & \text{если } y^2 < x^2 + 1 \end{cases}$ $y = \begin{cases} ax^2/bx, & \text{если } x > 3 \\ ax - b, & \text{если } x \leq 3 \end{cases}$	$x = 3.5$ $a = 3.2$ $b = 2.3$	$y, z.$
7	$z = \begin{cases} \sqrt[3]{ax+1}, & \text{если } x < d \\ \sin(bx+1), & \text{если } x = d \\ \cos(cx+1), & \text{если } x > d \end{cases}$ $x = \begin{cases} \sqrt{ab}, & \text{если } a < b \\ \sqrt{a+b}, & \text{если } a \geq b \end{cases}$	$a = 3.1$ $b = 4.2$ $c = 0.5$ $d = 5.3$	$x, z.$
8	$z = \begin{cases} \ln ax, & \text{если } x < 3 \\ bx^3, & \text{если } x = 3 \\ cx - 1, & \text{если } x > 3 \end{cases} \quad x = \begin{cases} a + bc, & \text{если } ab \geq c \\ abc, & \text{если } ab < c \end{cases}$	$a = 1.2$ $b = 2.5$ $c = 3.1$	$x, z.$
9	$y = \begin{cases} \sqrt[3]{a+x}, & \text{если } x < 1 \\ \ln bx, & \text{если } 1 \leq x \leq 5 \\ \sqrt{a+bx}, & \text{если } x > 5 \end{cases} \quad x = \begin{cases} a^2b, & \text{если } a < b \\ ab^2, & \text{если } a \geq b \end{cases}$	$a = 1.5$ $b = 2.1$	$x, y.$
10	$z = \begin{cases} \sin^2 x + 1, & \text{если } x \leq c \\ \cos x - 1, & \text{если } c < x < d \\ e^x + 1/a, & \text{если } x \geq d \end{cases}$ $x = \begin{cases} (a+c)d, & \text{если } a < c \\ (a-c)/d, & \text{если } a \geq c \end{cases}$	$a = 2.4$ $c = 3.2$ $d = 4.7$	$x, z.$



1	2	3	4
11	$y = \begin{cases} a\sqrt[3]{x}, & \text{если } x < 1 \\ bx, & \text{если } 1 \leq x \leq 3 \\ cx^2, & \text{если } x > 3 \end{cases} \quad x = \begin{cases} ab + c, & \text{если } a \leq b + 1 \\ a/b - c, & \text{если } a > b + 1 \end{cases}$	$a = 3.7$ $b = 2.9$ $c = 0.3$	$x, y.$
12	$y = \begin{cases} x^3 + 1, & \text{если } x < 4 \\ x^2 + 1, & \text{если } 4 \leq x < 5 \\ x + 1, & \text{если } x \geq 5 \end{cases} \quad x = \begin{cases} a^2 / b^2, & \text{если } a \leq b \\ a / b, & \text{если } a > b \end{cases}$	$a = 1.3$ $b = 4.5$	$x, y.$
13	$y = \begin{cases} (a + b)x, & \text{если } x < 3 \\ (a - b)x, & \text{если } x = 3 \\ ax / b, & \text{если } x > 3 \end{cases} \quad x = \begin{cases} a^2 + 1, & \text{если } ab \geq 1 \\ b^2 - 1, & \text{если } ab < 1 \end{cases}$	$a = 3.6$ $b = 2.3$	$x, y.$
14	$y = \begin{cases} a + x, & \text{если } x < 5 \\ ax, & \text{если } 5 \leq x < 7 \\ x / a, & \text{если } x > 7 \end{cases} \quad x = \begin{cases} \sqrt{a^2 + 1}, & \text{если } a \geq 2 \\ \sqrt[3]{a^3 + 1}, & \text{если } a < 2 \end{cases}$	$a = 2.5$	$x, y.$
15	$y = \begin{cases} \sqrt{b + x^2}, & \text{если } x < 1 \\ abx, & \text{если } 1 \leq x \leq 5 \\ bx^3, & \text{если } x > 5 \end{cases} \quad x = \begin{cases} ab + 3, & \text{если } ab \leq 3 \\ a / b - 3, & \text{если } ab > 3 \end{cases}$	$a = 1.5$ $b = 2.4$	$x, y.$
16	$y = \begin{cases} a\sqrt{x}, & \text{если } x < 2 \\ bx^2, & \text{если } 2 \leq x < 3 \\ c \cdot e^x, & \text{если } x \geq 3 \end{cases} \quad x = \begin{cases} (a + b) / c, & \text{если } a \leq b \\ (a - b)c, & \text{если } a > b \end{cases}$	$a = 5.4$ $b = 2.4$ $c = 1.9$	$x, y.$
17	$y = \begin{cases} b \cdot e^x, & \text{если } x < 2 \\ \sin x, & \text{если } 2 \leq x \leq 3 \\ ax^2 + b, & \text{если } x > 3 \end{cases} \quad x = \begin{cases} ab + 2, & \text{если } a \geq b \\ a / b + 2, & \text{если } a < b \end{cases}$	$a = 4.1$ $b = 3.7$	$x, y.$
18	$z = \begin{cases} \sin x + a, & \text{если } x < a \\ \cos \pi x - b, & \text{если } a \leq x \leq b \\ \operatorname{tg}(x^2), & \text{если } x > b \end{cases} \quad x = \begin{cases} (a - 1)(b - 2), & \text{если } a > b - 1 \\ (a + 1)(b + 2), & \text{если } a \leq b - 1 \end{cases}$	$a = 2.7$ $b = 3.5$	$x, z.$

1	2	3	4
19	$y = \begin{cases} x^3 + a, & \text{àñëè } 2 \leq x \leq 5 \\ x^2 + b, & \text{àñëè } -5 \leq x \leq -2 \\ c + x^2, & \text{à ñòàèüí ùõ ñëó÷àÿõ} \end{cases}$ $x = \begin{cases} a\sqrt{bc}, & \text{àñëè } c \leq 5 \\ b\sqrt{ac}, & \text{àñëè } c > 5 \end{cases}$	$a=4.6$ $b=1.4$ $c=3.8$	$x, y.$
20	$y = \begin{cases} 1+x+x^2, & \text{если } x \geq 5 \\ 1+x, & \text{если } -5 < x < 5 \\ 1/x^2, & \text{если } x \leq -5 \end{cases} \quad x = \begin{cases} c+0.7, & \text{если } c \leq d \\ d-0.5, & \text{если } c > d \end{cases}$	$c=2.5$ $d=1.9$	$x, y.$
21	$z = \begin{cases} x+y, & \text{если } xy < a \\ xy, & \text{если } a \leq xy \leq b \\ x-y, & \text{если } xy > b \end{cases} \quad y = \begin{cases} ax/b, & \text{если } x < 3 \\ ab/x, & \text{если } x \geq 3 \end{cases}$	$a=1.5$ $b=1.9$	$z, y.$
22	$z = \begin{cases} x^2 + y^2, & \text{если } y > x+1 \\ x^2 \ln y, & \text{если } y = x+1 \\ x^2 - y^2, & \text{если } y < x+1 \end{cases} \quad y = \begin{cases} x+a, & \text{если } x = a \\ x/a, & \text{если } x \neq a \end{cases}$	$a=3.4$ $x=1.4$	$z, y.$
23	$z = \begin{cases} ax+bx, & \text{если } x < 1 \\ ax/by, & \text{если } 1 \leq x \leq 9 \\ ax-by, & \text{если } x > 9 \end{cases} \quad y = \begin{cases} \sqrt{ab}, & \text{если } a \leq b \\ \sqrt{a+b}, & \text{если } a > b \end{cases}$	$a=3.2$ $b=2.4$ $x=4.1$	$z, y.$
24	$z = \begin{cases} ax+by, & \text{если } a \leq x \leq b \\ \ln(bx)+ay, & \text{если } -b \leq x \leq -a \\ xy+1, & \text{в остальных случаях} \end{cases}$ $y = \begin{cases} a^2 + b^2, & \text{если } a \neq b \\ ab-1, & \text{если } a = b \end{cases}$	$a=2.7$ $b=4.3$ $x=3.1$	$z, y.$
25	$y = \begin{cases} a + \sqrt{cx}, & \text{если } x < 3 \\ b + \sin \pi x, & \text{если } 3 \leq x \leq 5 \\ c - \cos ax, & \text{если } x > 5 \end{cases}$ $x = \begin{cases} c+ab, & \text{если } b \leq a-1 \\ c-ab, & \text{если } b > a-1 \end{cases}$	$a=3.7$ $b=2.9$ $c=0.3$	$x, y.$

1	2	3	4
26	$z = \begin{cases} ax^2 + bx + c, & \text{если } x < 2 \\ bx^2 + ax + c, & \text{если } 2 \leq x \leq 4 \\ cx^2 + ax + b, & \text{если } x > 4 \end{cases}$ $x = \begin{cases} (a+b)+c, & \text{если } a+1 < cb \\ (a-b)/(a-c), & \text{если } a+1 \geq cb \end{cases}$	$a=4.3$ $b=5.1$ $c=1.4$	$x, z.$
27	$y = \begin{cases} t \cdot e^{-x} + 5, & \text{если } x < 0 \\ x^2 / (t+1), & \text{если } 0 \leq x \leq 2.5 \\ x + \cos\left(\frac{\pi}{2}x\right), & \text{если } x > 2.5 \end{cases}$ $t = \begin{cases} a + 2.5x, & \text{если } a = x \\ ax - 1.4, & \text{если } a \neq x \end{cases}$	$a=2.7$ $x=3.2$	$t, y.$
28	$z = \begin{cases} \ln ax + \sqrt[5]{x}, & \text{если } x \geq 1 \\ 1 + bx, & \text{если } -1 < x < 1 \\ x^2 + e^{-x}, & \text{если } x \leq -1 \end{cases}$ $x = \begin{cases} a^2 / b^2 + 0.5, & \text{если } a - 1 \leq b \\ a / b - 0.5, & \text{если } a - 1 > b \end{cases}$	$a=5.1$ $b=3.6$	$x, z.$
29	$y = \begin{cases} \sqrt{abx} + 1, & \text{если } x < 4 \\ a + b + x, & \text{если } 4 \leq x \leq 6 \\ (a-b) / x^2, & \text{если } x > 6 \end{cases}$ $x = \begin{cases} \sqrt{a^2 + 1}, & \text{если } a \leq b \\ \sqrt{b^2 + 1}, & \text{если } a > b \end{cases}$	$a=4.7$ $b=2.3$	$x, y.$
30	$y = \begin{cases} ab + tgx, & \text{если } x < 1 \\ a\sqrt{x} - 1.5b, & \text{если } 1 \leq x \leq 3 \\ 1 + bx^2, & \text{если } x > 3 \end{cases}$ $x = \begin{cases} a^2 + 1, & \text{если } ab > 2 \\ b^2 - 1, & \text{если } ab \leq 2 \end{cases}$	$a=1.4$ $b=5.4$	$x, y.$



6.3. Лабораторная работа №3.

Тема: "Создание программных модулей VBA для обработки табличных данных MS Excel".

Цель работы: научиться обрабатывать табличные данные в среде VBA.

Задание:

Создать проект, формы пользователя и программные модули для обработки данных, организованных в таблице Excel соответствующей варианту задания (варианты заданий приведены на стр.67). Во всех заданиях обрабатываются данные таблицы, созданной в п.1.

Методические указания.

Рассмотрим пример решения следующей задачи:

Создать проект, формы пользователя и программные модули для обработки сессии студентов:

- 1 Создать таблицу следующей структуры:
 - индекс группы;
 - фамилия студента;
 - оценки по пяти экзаменам;
 - стипендия;
- 2 Добавить в таблицу произвольное количество строк;
- 3 Рассчитать стипендию студентам по результатам сессии, причем отличникам стипендию увеличить на 100%, сдавшим без троек – увеличить на 30%, а при наличии хотя бы одной двойки стипендию не начислять;
- 4 Создать на 2-ом листе книги ведомость для выдачи стипендии студентам;
- 5 Сортировать данные в таблице по выбранной колонке;
- 6 Изменить показатели студента с заданной фамилией;
- 7 Удалить из таблицы данные о студентах, имеющих более двух двоек.



Пример таблицы:

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

Для решения данной задачи предлагается следующий проект:



Проект решения
задачи
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

58

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*

Корректировка данных

Задайте ФИО студента (выбор или ввод)

выбор из списка

ввод

ФИО: []

Поиск | Отмена

- Петров
- Иванов
- Юшимова
- Морозова
- Абрамова
- Сидоров
- Жуковская
- Ющенко
- Жукова
- Петров

← *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*

Программный 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
```

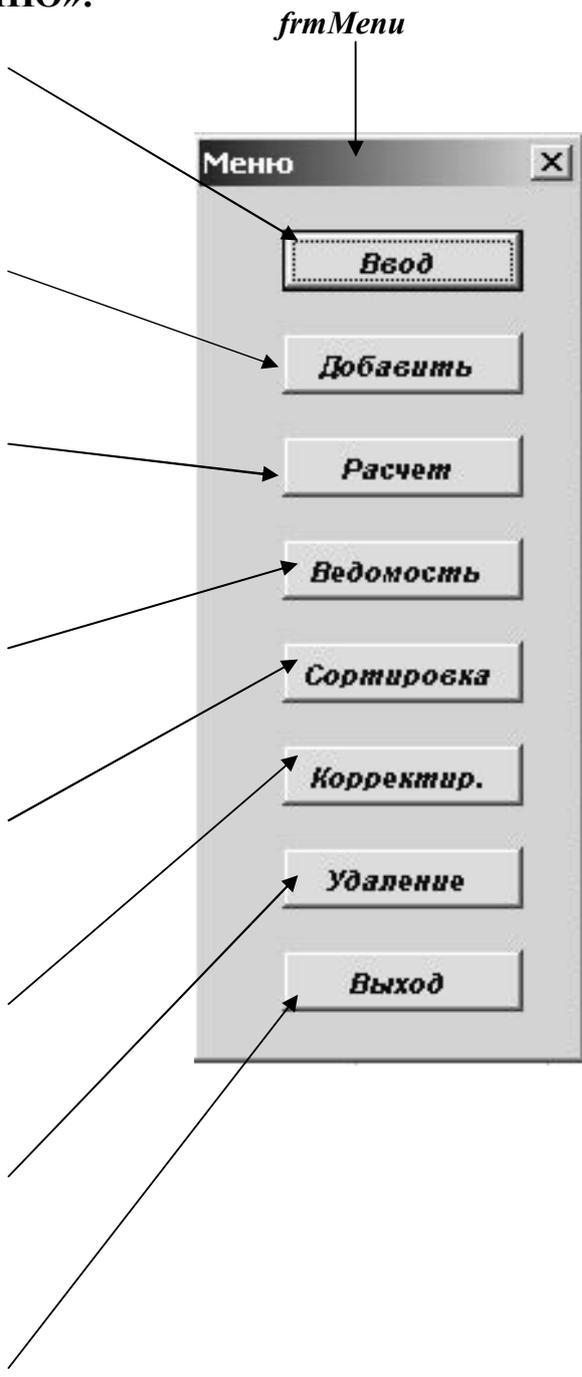
```
Private Sub CmdMenu4_Click()  
    Load frmL2  
    frmL2.Show  
End Sub
```

```
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
```





Программный модуль «ВВОД» (вызывается 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



Программный 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

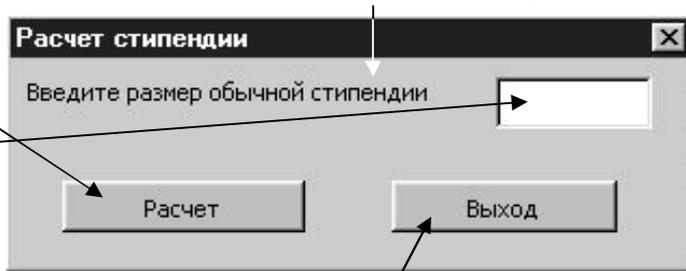
Loop

End Sub

Private Sub CmdE_Click()

frmRas.Hide

End Sub





Программный 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 сформированной таблицы на листе 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	

Программный 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

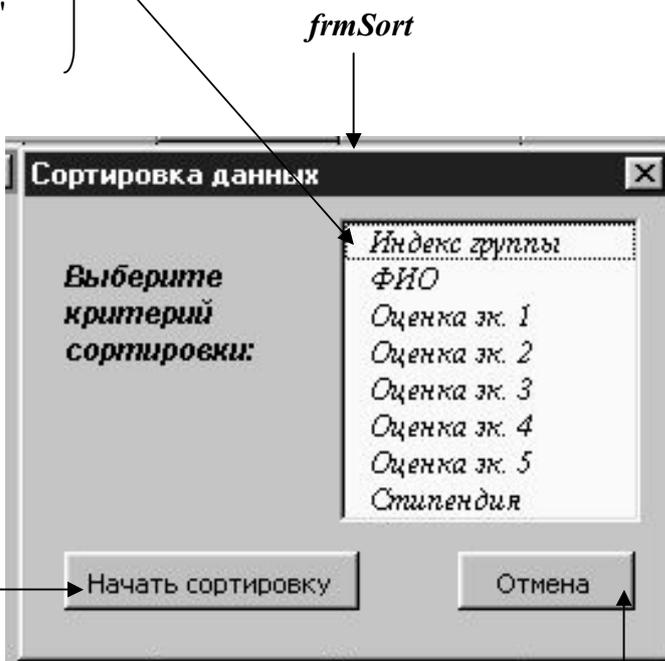
```

Private Sub CmdVuh_Click()

```

frmSort.Hide
End Sub

```



Программный 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

```

```

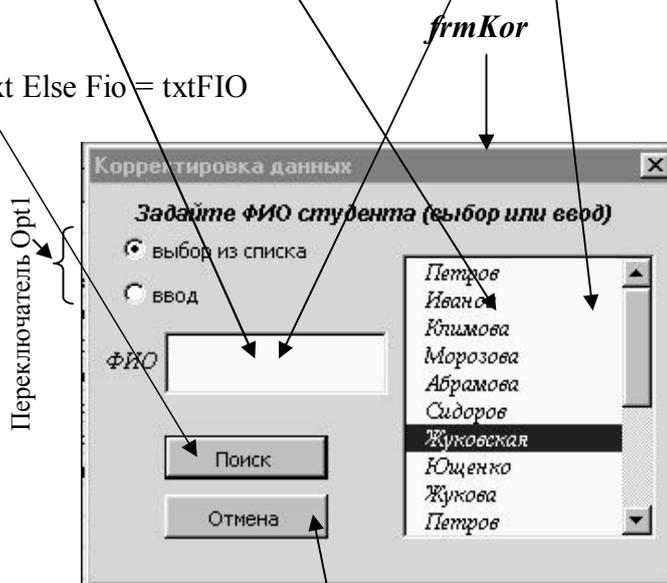
Private Sub CmdSeek_Click()
' Выбор фамилии из списка ListBox1
' или введенной в поле txtFIO
If Opt1 = True Then Fio = ListBox1.Text Else Fio = txtFIO
' Поиск фамилии в таблице
N = 2: Pr = 0
Do While Cells(N, 2) > " "
If Cells(N, 2) = Fio Then
Cells(N, 1).Select
Pr = 1
Exit Do
End If
N = N + 1
Loop
If 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

```



Программный модуль «КОРРЕКТИРОВКА» (продолжение):

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

Программный 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
```





Варианты заданий к лабораторной работе № 3

Вариант 1

1. Создать таблицу, которая содержит сведения о месячной заработной плате рабочих завода.
Структура таблицы:
 - табельный номер;
 - фамилия рабочего;
 - наименование цеха;
 - размер заработной платы в месяц.
2. Предусмотреть режим добавления в таблицу произвольного количества строк.
3. Вычислить:
 - общую сумму выплаты за месяц по заданному цеху
 - среднемесячный заработок этого цеха.
4. Создать на 2 листе книги ведомость для начисления заработной платы рабочих заданного цеха.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить размер заработной платы у рабочего с заданной фамилией на X% (значение X задано).
7. Удалить из таблицы 2 записи. Номера записей задать.

Вариант 2

1. Создать таблицу, которая содержит сведения о количестве изделий, собранных сборщиками цеха за неделю.
Структура таблицы:
 - фамилия сборщика;
 - количество изделий, собранных им ежедневно в течение шестидневной недели, то есть – в понедельник, вторник, и т.д.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - общее количество деталей, собранных сборщиками цеха;
 - фамилия сборщика, который собрал наибольшее число изделий, и определить день, когда он достиг наивысшей производительности труда.



4. Создать на 2 листе книги таблицу, которая содержит следующую информацию: фамилию сборщика и общее количество детали, собранное им за неделю.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить в X раз (значение X задано) количество деталей, собранных в заданный день недели.
7. Удалить из таблицы запись о сборщике с заданной фамилией.

Вариант 3

1. Создать таблицу, которая содержит сведения о количестве изделий категории А, В, С, собранных рабочим за месяц.

Структура таблицы:

- фамилия сборщика;
 - наименование цеха;
 - количество изделий (по категориям), собранных рабочим за месяц;
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
 3. Рассчитать:
 - общее количество изделий каждой категории;
 - общее количество изделий, собранных всеми рабочими заданного цеха;
 - средний размер заработной платы работников заданного цеха.
 4. Создать на 2 листе книги ведомость для начисления заработной платы рабочих заданного цеха.
 5. Сортировать данные в таблице по заданному столбцу.
 6. Уменьшить количество изделий заданной категории на $X\%$ (значение X - задано).
 7. Удалить из таблицы записи, в которых количество изделий заданной категории менее Y штук (значение Y - задано).

Вариант 4

1. Создать таблицу, которая содержит сведения о телефонных абонентах.

Структура таблицы:

- фамилия абонента;
- номер телефона;



- год установки телефона;
 - наличие задолженности в оплате.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
 3. Определить:
 - общее количество телефонов, установленных начиная с заданного года по нынешний год.
 - по заданной фамилии абонента номер его телефона.
 4. Создать на 2 листе книги таблицу, которая содержит фамилии задолжников и их телефоны.
 5. Сортировать данные в таблице по заданному столбцу.
 6. Изменить номер телефона у абонента с заданной фамилией
 7. Удалить из таблицы запись, которая содержит заданный номер телефона.

Вариант 5

1. Создать таблицу, которая содержит сведения об ассортименте игрушек в магазине.
Структура таблицы:
 - название игрушки;
 - цена;
 - количество;
 - возрастные границы, например, 2-5, то есть от 2 до 5 лет.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить стоимость самой дорогой игрушки и ее наименование. По вводимым значениям A , B и X выдавать название игрушки, которое по стоимости не превышает X и подходит ребенку от A до B лет.
4. Создать на 2 листе книги таблицу, которая содержит следующую информацию: наименования игрушек, которые подходят детям от X до Y лет и их цены.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить цену игрушки для заданной возрастной группы на $Z\%$ (значение Z – задано).
7. Удалить из таблицы запись об игрушке с заданным названием.



Вариант 6

1. Создать таблицу, которая содержит сведения о сдаче студентами первого курса сессии.
Структура таблицы:
 - группа;
 - фамилия студента;
 - оценка по 5 экзаменам
 - признак участия в общественной работе: "Да"/"Нет".
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить общее число активистов в списке.
4. Создать на 2 листе книги таблицу, содержащую ведомость начисления стипендии студентам заданной группы. Рассчитать размер стипендии по следующему принципу. Студенту, получившему все оценки «5» и активно участвующему в общественной работе, назначается повышенная стипендия – доплата 50%. Студенту, получившему «4» и «5», назначается обычная стипендия – ее необходимо задать. Студенту, получившему одну оценку «3», но активно занимающемуся общественной работой, также назначается обычная стипендия. Остальным студентам стипендия не назначается. Предусмотреть ввод размера обычной стипендии.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить признак участия в общественной жизни студентов с оценками «55555» на противоположный.
7. Удалить из таблицы записи о студентах, которые имеют хотя бы одну двойку.

Вариант 7

1. Создать таблицу, которая содержит информацию о личной коллекции книголюбца.
Структура таблицы:
 - шифр книги;
 - название книги;
 - фамилия автора;
 - год издания;
 - место расположения (номер стеллажа, номер шкафа, номер полки);
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.



3. Определить:
 - общее количество книг в коллекции, а также число книг заданного года издания.
 - по заданному автору и названию книги информацию о местонахождении книги.
4. Создать на 2 листе книги таблицу, которая содержит информацию о книге заданного автора, которая находится в коллекции.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить место расположения книг самого позднего года издания.
7. Удалить из таблицы записи о книгах, изданных до заданного года.

Вариант 8

1. Создать таблицу, которая содержит сведения о сдаче сессии студентами.
Структура таблицы:
 - группа;
 - фамилия студента;
 - результаты экзаменов по 5 предметам (например, 55445);
 - результаты зачетов по 5 предметам (например, 33Н33, где «3» – зачет, а «Н» - незачет).
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Вычислить средний балл, полученный каждым студентом заданной группы, и средний балл этой группы, по каждому предмету. Определить общее количество задолженностей (по экзаменам и зачетам в сумме) каждого студента заданной группы и общее число студентов-должников той же группы.
4. Создать на 2 листе книги таблицу, которая содержит сведения о неуспевающих студентах: группу, фамилию и количество задолженностей.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить результат экзамена по заданному предмету студенту с заданной фамилией.
7. Удалить из таблицы записи о студентах, которые имеют три задолженности.



Вариант 9

1. Создать таблицу, которая содержит сведения о рейсах Аэрофлота.
Структура таблицы:
 - номер рейса;
 - пункт назначения;
 - дата вылета;
 - время вылета;
 - время прибытия;
 - количество свободных мест в самолете.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - номера рейсов и время отправления самолетов в заданный город;
 - по заданному городу и дате отправления количество свободных мест на рейсы;
 - общее количество рейсов за сутки в заданный город.
4. Создать на 2 листе книги таблицу, которая содержит номера рейсов, дату и время отправления самолетов в заданный город.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить время вылета и время прибытия у заданного рейса.
7. Удалить информацию о ночных рейсах (считать ночными рейсы с временем вылета в интервале 23:00–3:00).

Вариант 10

1. Создать таблицу, которая содержит сведения об ассортименте обуви в магазине.
Структура таблицы:
 - артикул обуви (артикул начинается с буквы Д – для дамской обуви, М, – для мужской, П – для детской обуви, например, Д0321);;
 - наименование;
 - размер;
 - количество пар обуви;
 - стоимость.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.



3. Определить:
 - стоимость обуви заданного артикула, и какие размеры имеются в наличии;
 - общее количество пар детской обуви, имеющейся в магазине, и ее суммарную стоимость.
4. Создать на 2 листе книги таблицу, которая содержит информацию о всех моделях обуви заданного вида.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить стоимость обуви с заданным наименованием на $X\%$ (значение X задано).
7. Удалить записи о мужской обуви со стоимостью ниже заданной.

Вариант 11

1. Создать 2 таблицы, содержащие сведения о десяти нападающих хоккейных команд «Динамо» и «Шахтер».
Структура таблиц:
 - фамилия хоккеиста;
 - число заброшенных им шайб;
 - число сделанных им голевых передач;
 - заработанное штрафное время.
2. Предусмотреть возможность добавления в таблицы произвольного количества строк.
3. Вычислить:
 - общее число шайб, забитых хоккеистами каждой команды
 - суммарное штрафное время каждой команды
4. Создать на 2 листе книги таблицу, которая содержит фамилию шести лучших игроков из обеих команд (по 3 игрока из каждой команды), и сумму очков каждого игрока (голы + передача).
5. Сортировать данные в таблице по заданному столбцу.
6. Для хоккеистов «Динамо» уменьшить количество забитых шайб на заданное число, если это число больше 0.
7. Удалить информацию о нападающих обеих команд, которые заработали штрафное время выше заданного.



Вариант 12

1. Создать таблицу, которая содержит сведения о том, какую из пяти дисциплин желает слушать студент.
Структура таблицы:
 - фамилия студента;
 - группа;
 - наименование пяти дисциплин (выбираемая дисциплина замечается символом «1» иначе – пробел;
 - средний балл успеваемости.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - количество слушателей каждой дисциплины.
 - число слушателей заданной дисциплины, у которых средний балл превышает заданный.
4. Создать на 2 листе книги таблицу, которая содержит фамилию, группу и средний балл всех слушателей заданной дисциплины. Если число их превысит заданное, то отобразить студентов, которые имеют более высокий средний балл успеваемости.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить сведения для студента с заданной фамилией.
7. Удалить информацию о студентах со средним баллом ниже заданного.

Вариант 13

1. Создать таблицу, которая содержит сведения об отправлении поездов.
Структура таблицы:
 - номер поезда;
 - станция назначения;
 - дата отправления;
 - время отправления;
 - время прибытия;
 - наличие (количество) билетов в вагоны каждого вида отдельно: купе и плацкарт.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:



- количество свободных мест в купейные вагоны поезда с заданными номером и датой;
 - количество поездов, следующих до заданной станции назначения.
4. Создать на 2 листе книги таблицу, которая содержит информацию о поездах, которые отправляются до заданной станции в заданном интервале времени (временной интервал задать двумя значениями, например 13:00 и 18:30).
 5. Сортировать данные в таблице по заданному столбцу.
 6. Изменить сведения о наличии билетов на поезд с заданным номером.
 7. Удалить запись о поездах, которые следуют до заданной станции.

Вариант 14

1. Создать таблицу, которая содержит сведения о сотрудниках института.
Структура таблицы:
 - фамилия;
 - пол;
 - название отдела;
 - дата рождения;
 - дата поступления на работу;
 - должность;
 - оклад.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Рассчитать:
 - стаж работы всех сотрудников;
 - средний стаж работы сотрудников заданного отдела;
 - количество сотрудников с окладом ниже заданного.
4. Создать на 2 листе книги таблицу, которая содержит список сотрудников пенсионного возраста (на сегодняшний день), указав стаж работы. Учесть разницу в пенсионном возрасте женщины и мужчины.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить оклад на заданное число процентов лицам со стажем работы выше заданного.
7. Удалить из таблицы сведения о сотрудниках пенсионного возраста.



Вариант 15

1. Создать таблицу, которая содержит сведения о пациентах клиники.
Структура таблицы:
 - фамилия пациента;
 - пол;
 - возраст;
 - место проживания (город);
 - диагноз.
2. Разработать режим добавления в таблицу произвольного количества строк.
3. Определить:
 - количество иногородних, прибывших в клинику;
 - количество пациентов с заданным диагнозом;
 - количество пациентов пенсионного возраста.
4. Создать на 2 листе книги таблицу, которая содержит список пациентов старше заданного возраста с заданным диагнозом.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить информацию у пациента с заданной фамилией.
7. Удалить информацию о пациентах пенсионного и детского возраста.

Вариант 16

1. Создать таблицу, которая содержит сведения о багаже, сданном пассажиром в камеру хранения.
Структура таблицы:
 - фамилия пассажира;
 - номер ячейки;
 - шифр багажа;
 - количество вещей;
 - вес багажа.
2. Разработать режим добавления в таблицу произвольного количества строк.
3. Определить:
 - общий средний вес одной вещи;



- багаж, у которого средний вес одной вещи отличается не более чем на X кг от общего среднего веса одной вещи (значение X задано);
 - число владельцев багажа с количеством вещей большим заданного.
4. Создать на 2 листе книги таблицу, которая содержит информацию о багаже (и о его владельце), вес которого превышает заданный.
 5. Сортировать данные в таблице по заданному столбцу.
 6. Изменить информацию о багаже пассажира с заданной фамилией.
 7. Удалить сведения о багаже с максимальным весом.

Вариант 17

1. Создать таблицу, которая содержит сведения о работе обменного пункта за день (продажа и покупка одного вида валюты).
Структура таблицы:
 - наименование банка;
 - курс продажи;
 - курс покупки.
 - количество проданных у.е.;
 - количество купленных у.е.;
2. Разработать режим добавления в таблицу произвольного количества строк.
3. Определить:
 - максимальную разницу между курсом продажи и покупки у.е.;
 - суммарное количество у.е., проданных всеми обменными пунктами за день, и общую сумму, на которую они проданы.
4. Создать на 2 листе книги таблицу, которая содержит наименование фирмы, количество проданных у.е. и общую сумму продажи.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить курс покупки валюты у заданного банка.
7. Удалить информацию о банке с максимальной разницей между курсом продажи и курсом покупки.



Вариант 18

1. Создать таблицу, которая содержит сведения о заказе на ремонт оборудования.
Структура таблицы:
 - номер заказа;
 - дата оформления заказа;
 - фамилия заказчика;
 - наименование оборудования;
 - сложность ремонта (низкая, средняя, высокая);
 - стоимость;
 - дата окончания ремонта.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - суммарную стоимость всех заказов;
 - количество заказов на ремонт заданного вида сложности;
 - минимальную стоимость ремонта.
4. Создать на 2 листе книги таблицу, которая содержит сведения о продолжительности ремонта заказов, оформленных в течение заданного сезона (например, весной): номер заказа, фамилия заказчика, наименование оборудования, продолжительность заказа.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить стоимость ремонта на $X\%$ (X задано) у оборудования с заданным наименованием.
7. Удалить информацию о заказах со стоимостью ремонта ниже средней.

Вариант 19

1. Создать таблицу, которая содержит сведения о междугородных разговорах.
Структура таблицы:
 - фамилия абонента;
 - номер телефона;
 - дата разговора;
 - код города;
 - длительность разговора в минутах;
 - стоимость 1 минуты.



2. Предусмотреть возможность добавления в таблицу произвольного количества строки.
3. Определить:
 - максимальную стоимость разговора;
 - суммарную стоимость всех разговоров;
 - общее количество разговоров в город с заданным кодом.
4. Создать на 2 листе книги таблицу, которая содержит сведения о стоимости разговоров, которые состоялись в интервале между двумя заданными датами.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить стоимость всех разговоров на $X\%$ (значение X задано).
7. Удалить сведения о разговорах абонента с заданной фамилией.

Вариант 20

1. Создать таблицу, которая содержит сведения о продаже товаров. Структура таблицы:
 - наименование товара;
 - шифр товара;
 - оптовая цена;
 - розничная цена;
 - дата продажи;
 - количество проданного товара.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. По заданному шифру товара выдавать информацию о нем. Определить:
 - суммарная прибыль от продажи всех товаров;
 - наименование товаров, которые продаются по самой высокой и самой низкой цене.
4. Создать на 2 листе книги таблицу, которая содержит сведения о продаже товаров в интервале между двумя заданными датами. Таблица должна содержать наименование товара и суммарную выручку.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить розничную цену на $X\%$ у товара с заданным наименованием (значение X задано).



7. Удалить сведения о товарах, проданных в течение заданного сезона (например, зимой).

Вариант 21

1. Создать таблицу, которая содержит сведения о выпуске изделий на заводе.
Структура таблицы:
 - наименование изделия;
 - код изделия;
 - номер цеха-изготовителя;
 - дата изготовления;
 - количество;
 - себестоимость;
 - цена.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - суммарную разницу между себестоимостью и ценой всех изготовленных изделий;
 - общее количество изделий, изготовленных ранее заданной даты;
 - цену изделия по заданному наименованию.
4. Создать на 2 листе книги таблицу, которая содержит сведения о товарах, изготовленных в заданном цехе и себестоимость которых составляет половину цены.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить себестоимость и цену у заданного изделия.
7. Удалить сведения о товарах с ценой ниже средней цены.

Вариант 22

1. Создать таблицу, содержащую сведения о повременной оплате труда рабочих.
Структура таблицы:
 - фамилия рабочего;
 - разряд;
 - номер цеха;



- дата;
 - стоимость одного часа;
 - количество отработанного времени.
2. Разработать режим добавления в таблицу произвольного количества строк.
 3. Определить:
 - среднее количество времени, отработанного за день;
 - максимальную стоимость одного часа;
 - по заданной дате и фамилии рабочего кол-во отработанных часов.
 4. Создать на 2 листе книги таблицу, которая содержит следующие сведения о работниках заданного цеха: фамилия работника; суммарная стоимость отработанного им времени.
 5. Сортировать данные в таблице по заданному столбцу.
 6. Изменить стоимость одного часа у работников заданного цеха на $X\%$ (значение X задано).
 7. Удалить информацию о работниках, у которых количество отработанных часов за день меньше среднего.

Вариант 23

1. Создать таблицу, которая содержит сведения о заказах на пошив одежды.
Структура таблицы:
 - фамилия клиента;
 - номер заказа;
 - наименование изделия;
 - дата приема заказа;
 - дата выполнения заказа;
 - фамилия мастера;
 - стоимость.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - количество заказов, выполненных мастером с заданной фамилией;
 - стоимость наиболее дорогого заказа;



- среднюю стоимость заказов.
4. Создать на 2 листе книги таблицу, которая содержит фамилию клиента, номер заказа и продолжительность его выполнения.
 5. Сортировать данные в таблице по заданному столбцу.
 6. Изменить сведения о заказах, оформленных позже заданной даты.
 7. Удалить информацию о клиенте с заданной фамилией.

Вариант 24

1. Создать таблицу, которая содержит сведения о товарах, которые экспортируются.
Структура таблицы:
 - шифр товара;
 - наименование товара;
 - страна, которая импортирует товар;
 - размер партии, которая поставляется, в штуках;
 - стоимость 1 штуки в у.е..
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - суммарный объем товаров, которые импортируются заданной страной;
 - суммарную стоимость партии товара по заданному шифру
 - минимальную стоимость товара.
4. Создать на 2 листе книги таблицу, которая содержит сведения о стоимости товаров, которые импортируются заданной страной. Таблица должна содержать наименование товара и суммарную стоимость партии.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить на X% стоимость заданного товара (значение X задано).
7. Удалить информацию о товаре, который имеет минимальную стоимость.

Вариант 25

1. Создать таблицу, содержащую сведения об изготовлении деталей.
Структура таблицы:
 - номер цеха;



- фамилия рабочего;
 - количество изготовленных деталей;
 - количество бракованных деталей;
 - оплата за изготовление одной качественной детали.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
 3. Определить:
 - общее количество бракованных деталей, изготовленных всеми мастерами заданного цеха;
 - сумму штрафа за каждую бракованную деталь (штраф составляет 20% от оплаты за качественную);
 - фамилия мастера, который изготовил максимальное количество качественных деталей.
 4. Создать на 2 листе книги таблицу, которая содержит сведения об оплате труда рабочих. Таблица должна содержать фамилию рабочего, номер цеха и сумму к выплате с учетом штрафа и налога (налог составляет X% от стоимости оплаты).
 5. Сортировать данные в таблице по заданному столбцу.
 6. Увеличить оплату сотрудникам, которые работают без брака на Y% (значение Y задано).
 7. Удалить сведения о работниках, у которых процент брака составляет более 50% от количества изготовленных.

Вариант 26

1. Создать таблицу, которая содержит информацию о стоимости лечения больных в стационарной клинике.
Структура таблицы:
 - название болезни;
 - номер отделения;
 - длительность лечения;
 - стоимость 1 дня лечения;
 - расходы на лекарства.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - расходы на лекарства по заданному названию болезни;



- среднюю стоимость 1 дня лечения;
 - рассчитать суммарную стоимость лечения каждой болезни, включая расходы на лекарства.
4. Создать на 2 листе книги таблицу, которая содержит сведения о стоимости лечения в заданном отделении.
 5. Сортировать данные в таблице по заданному столбцу.
 6. Увеличить стоимость 1 дня лечения заданной болезни на $X\%$ (значение X задано).
 7. Удалить сведения о лечении болезней с минимальной стоимостью одного дня.

Вариант 27

1. Создать таблицу, которая содержит сведения о продаже книг. Структура таблицы:
 - название книги;
 - фамилия автора;
 - номер магазина;
 - цена;
 - продано;
 - остаток.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - количество проданных книг в заданном магазине;
 - суммарную стоимость всех непроданных книг;
 - среднюю цену одной книги.
4. Создать на 2 листе книги таблицу, которая содержит суммарную стоимость книг, проданных каждым магазином
5. Сортировать данные в таблице по заданному столбцу.
6. Уменьшить цену на книги, у которых остаток превышает количество проданных более чем в 2 раза.
7. Удалить сведения о книгах с ценой ниже заданной.

**Вариант 28**

1. Создать таблицу, которая содержит сведения об учете использования материалов.
Структура таблицы:
 - шифр детали;
 - количество изготовленных деталей;
 - количество брака;
 - шифр материала;
 - расход материала на 1 деталь.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - для всех деталей суммарный расход материала на брак;
 - количество качественных деталей;
 - деталь, на которую расходуется больше всего материала.
4. Создать на 2 листе книги таблицу, которая содержит шифр детали и процент брака.
5. Сортировать данные в таблице по заданному столбцу.
6. Изменить сведения о детали с заданным шифром.
7. Удалить сведения о деталях, на изготовление которых расходуется количество материала, превышающее заданное значение.

Вариант 29

1. Создать таблицу, содержащую сведения о продаже лекарств в аптеке
Структура таблицы:
 - номер аптеки;
 - название лекарства;
 - категория лекарства (антибиотик, жаропонижающе, витамин, снотворное и так далее);
 - цена 1 упаковки;
 - количество проданных упаковок;
 - остаток;
 - дата окончания срока годности.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:



- суммарное количество упаковок лекарств, проданных всеми аптеками и принадлежащих заданной категории;
 - стоимость всей непроданных упаковок;
 - среднюю стоимость лекарств.
4. Создать на 2 листе книги таблицу, которая содержит информацию о просроченных лекарствах: номер аптеки, название, категорию и дату истечения срока.
 5. Сортировать данные в таблице по заданному столбцу.
 6. Изменить цену у лекарств, которые принадлежат заданной категории.
 7. Удалить сведения о просроченных лекарствах.

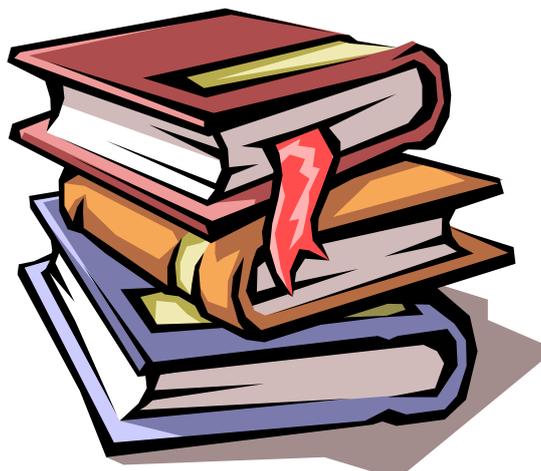
Вариант 30

1. Создать таблицу, которая содержит сведения о сотрудниках фирмы.
Структура таблицы:
 - фамилия сотрудника;
 - должность;
 - дата рождения;
 - дата поступления на работу;
 - номер отдела;
 - оклад.
2. Предусмотреть возможность добавления в таблицу произвольного количества строк.
3. Определить:
 - возраст сотрудников (количество полных лет) при поступлении на работу и на текущее время;
 - количество сотрудников заданного отдела младших X лет (значение X задано);
 - минимальный размер оклада.
4. Создать на 2 листе книги таблицу, которая содержит сведения о 5-ти самых высокооплачиваемых сотрудниках фирмы.
5. Сортировать данные в таблице по заданному столбцу.
6. Увеличить на $Y\%$ оклады сотрудникам, которые проработали не меньше Z лет (значение X и Z заданы).
7. Удалить информацию о сотрудниках старше заданного возраста.



СПИСОК ЛИТЕРАТУРЫ

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. Ченел 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.





**" Основы программирования
в среде Visual Basic for Application"**

**Методические указания и задания
к выполнению лабораторных работ**

(для студентов всех специальностей)

Составители: *Зензеров Владимир Иванович, ст. преподаватель*
Славинская Людмила Васильевна, ст. преподаватель
Едемская Евгегия Николаевна, ст. преподаватель