ЛАБОРАТОРНАЯ РАБОТА N 1

**Часть 1**

* 1. **VBA. ЭЛЕМЕНТЫ УПРАВЛЕНИЯ**

**Цель – ознакомиться с интерфейсом VBA, некоторыми его объектами, их свойствами и методами.**

VBA относится к языкам объектно-ориентированного программирования (**ООП**). ООП можно описать как методику анализа, проектирования и написания приложений с помощью объектов. ***Объект*** – комбинация кода и данных, рассматривается как единое целое, например элемент управления, форма и компонент приложения. Каждый объект определяется по принадлежности к классу. Примеры объектов: рабочий лист (Worksheet), диапазон (**Range**), диаграмма (**Chart**), форма (**UserForm**) и т.д.

Доступ к данному языку программирования можно осуществлять практически из любого приложения Windows. Мы будем работать вместе с Microsoft Excel, который будет являться основным приложением для проекта VBA.

**Элементы объектно-ориентированного программирования в Excel VBA.**

Объектами в Excel называется всё то, на что вы можете указать. Например, это рабочие листы, ячейки, меню, кнопки и диапазоны ячеек. Ссылка на объект часто состоит из нескольких компонент, каждая из которых ограничивает область возможных объектов, пока не будет указан точно конкретный объект. Каждая ступень ограничения области отделяется точкой (**.**). Например, запись

**Workbooks("Книга2").Worksheets("Лист1")**

позволяет получить ссылку на рабочий лист с именем **Лист1**, который находится в рабочей книге с именем **Книга2**.

То есть среди множества объектов типа рабочих книг (Workbooks) Excel нас интересует конкретный объект с именем, указанным в виде строковой константы "Книга2", а среди множества объектов типа рабочих листов (Worksheets) этой книги мы хотим получить ссылку (указатель) на конкретный лист с именем, указанным в вид е строковой константы "Лист1".

При этом следует подчеркнуть, что этот лист также является объектом Excel типа Worksheet и кроме того он сам является контейнером для множества других объектов, которые содержит в себе. Например, объекты ячеек, столбцов, строк, диапазонов и т.д. Для любого объекта в Excel определены, характеризующие этот объект конкретные наборы ***свойств и методов***.

***Свойства*** – это атрибуты, описывающие строго определенные характеристики объекта. Так, ячейка имеет такие свойства, как имя, формат, значение. Изменяя свойства, вы можете менять те или иные характеристики объекта. Например, чтобы рабочий лист "Лист1" из рабочей книги "Книга2" переименовать на "Мой лист" надо свойству Name этого листа присвоить новое значение:

**Workbooks("Книга2").Worksheets("Лист1").Name="Мой лист"**

***Методы*** – выполняют определенные действия с объектами. Например, выделяют, удаляют, создают и изменяют объекты. Так, для очистки содержимого ячейки **В6** на листе "Лист1" рабочей книги "Книга2" можно использовать метод ***Clear*** :

**Workbooks("Книга2").Worksheets("Лист1").Range("B6").Clear**

При этом свойство **Range**("имя ячейки") возвращает ссылку на указанную ячейку, к которой и применяется метод **Clear**.

Если надо очистить ячейку **B6** активного листа текущей рабочий книги, то указывать **Workbooks("Книга2").Worksheets("Лист1")** не надо, и метод по умолчанию будет применем к ячейке активного лист:

**Range("B6").Clear**

Для того, чтобы среди множества листов рабочей книги "Финансы" активным сделать лист с имеем "Квартал\_3" можно использовать метод **Activate** :

**Workbooks("Финансы").Worksheets("Квартал\_3").Activate**

Для добавления в Excel новых экземпляров объектов того или иного типа используется метод **Add.** Например, его можно использовать для добавления нового листа в рабочую книгу. При этом можно указать аргументы, которые определят тип вставляемого листа (рабочий или модульный), место вставки и количество вставляемых листов. Синтаксис метода **Add** для объекта **Worksheets** имеет вид:

**Worksheets.Add ([Before], [After], [Count], [Type])**

Аргументы **Before** и **After** позволяют указать, где должен появиться новый лист. Если вы указываете аргумент **Before**, то лист добавляется перед указанным листом (по умолчанию - перед активным). Аргумент **Count** указывает количество добавляемых листов (по умолчанию 1). Аргумент **Type** определяет тип добавляемого листа (по умолчанию - рабочий лист). При вызове надо присваить (знак ":=") значения тем аргументам, которым считаете нужным точно указать, а остальные будут использованы по умолчанию. Например, добавим два новых рабочих листа перед листом "Лист3":

**Worksheets.Add Count:=2, Before:=Sheets("Лист3")**

Если метод **Add** используется для ссылки на вновь созданный объект и является частью оператора VBA, то аргументы необходимо заключать в скобки. Например, добавим в рабочую книгу после листа "Квартал\_3" новый лист и присвоим ему имя "Квартал\_4":

**Worksheets.Add( After:=Sheets("Квартал\_3") ).Name = "Квартал\_4"**

**1.2. Редактор VBA**

Редактор VBA активизируется командой **Файл, Visual Basic (Alt+F11).**

Возвратиться из редактора VBA в рабочую книгу можно нажатием кнопки **Вид Microsoft Excel.** Интерфейс VBA состоит из следующих основных компонентов:

**окно проекта,**

**окно свойств,**

**окно редактирования кода,**

**окна форм,**

**меню и панели инструментов.**

**Окно проекта**

Окно проекта в редакторе VBA активизируется выбором команды **View►Project Explorer** (Ctrl+R ) или нажатием кнопки **Project Explorer**

 В окне проекта представлена иерархическая структура файлов форм и модулей текущего проекта

В проекте автоматически создается модуль для каждого рабочего листа и для всей книги. Кроме того, модули создаются для каждой пользовательской формы макросов и классов. По своему предназначению модули делятся на два типа- *модули объектов* и *стандартные.* К стандартным модулям относятся те, которые содержат макросы. Такие модули добавляются в проект командой **Insert►Module** (**Вставка ►Модуль)**. К модулям объектов относятся модули, связанные с рабочей книгой, рабочими листами, формами, и модули класса.

Формы создаются командой **Insert►UserForm**, а модули класса - командой **Insert►Class Module** (**Вставка►Модуль** **класса**). По мере создания добавления и удаления файлов из проекта эти изменения отображаются в окне проекта.

В окне проекта выводится проекты всех открытых рабочих книг. Это позволяет легко копировать формы и коды из одного проекта в другой, что убыстряет процесс создания новых приложений.

Для создания диалоговых окон, разрабатываемых приложений в VBA, используются формы. Редактор форм является одним из основных инструментов визуального программирования. После добавления формы в проект на экран выводится незаполненная форма с панелью инструментов **ToolBox (Панель элементов)** (рис. 1).

**Рис. 1**

Используя панель инструментов **ToolBox (Панель элементов)**, можно сконструировать любое требуемое для приложения диалоговое окно.

Размещение нового управляющего элемента в форме осуществляется следующей последовательностью действий:

1. Щелкните значок того элемента, который вы собираетесь разместить в форме.

2. Поместите указатель мыши на то место, где будет располагаться управляющий элемент.

3. Нажмите левую кнопку мыши и, не отпуская ее, растяните появившийся прямоугольник до требуемых размеров.

4. Отпустите кнопку мыши.

Размеры формы и расположенных на ней элементов управления можно изменять. Технология изменения размеров стандартная для Windows: выделить изменяемый элемент, разместить указатель мыши на одном из размерных маркеров и протащить его при нажатой левой кнопки мыши так, чтобы объект принял требуемые размеры. Окно редактирования форм поддерживает операции буфера обмена. Таким образом, можно копировать, вырезать и вставлять элементы управления, расположенные на поверхности формы.

Любой управляющий элемент, который вы разместили на форме, обладает рядом свойств

**Окно свойств**

В окне отображаются свойства активного в данный момент объекта.. Используя это окно, можно просматривать свойства и изменять их установки. Для просмотра свойств выбранного объекта надо либо щелкнуть кнопку **Properties Window (Окно свойств)**, либо выбрать команду **View►Properties Window (F4).**

На рис. 2 показано окно свойств активной в данный момент формы **UserForm1.**

Окно свойств состоит из двух составных частей: верхней и рабочей. В верхней части окна свойств располагается раскрывающийся список, из которого можно выбрать любой элемент управления текущей формы или саму форму. Рабочая часть состоит из двух вкладок: **По алфавиту** (Alphabetic) и **По категориям** (Categorized), отображающие набор свойств в алфавитном порядке или по категориям. В обоих вкладках свойство Name (имя элемента управления) будет первым. Изменяются значения свойств одним из следующих способов:

* Вводом с клавиатуры значения свойства в соответствующее поле.
* Значения большинства свойств можно выбрать из раскрывающегося списка. Раскрывающийся список активизируется щелчком в соответствующем поле окна свойств

**Рис. 2**

**Инструменты панели элементов**

При создании форм используются инструменты панели элементов, назначение которых рассмотрено ниже.

Допускается настройка панели элементов путем добавления к ней страниц или элементов управления с помощью команды Дополнительные элементы из меню **Сервис**.

При добавлении страницы в нее автоматически вставляется инструмент Выбор объектов.

**Стандартные элементы управления панели элементов:**

 **Выбор объектов**

Это единственный инструмент на панели элементов, не создающий никаких элементов управления. Он служит для изменения размеров и положения элементов формы.

 **Надпись**

Позволяет отобразить в форме неизменяемый текст, например подпись к рисунку.

 **Поле**

Содержит вводимый и изменяемый пользователем текст.

 **Поле со списком**

Вставляет объект, являющийся сочетанием списка и поля. Пользователь может либо выбрать нужное значение из списка, либо ввести его в поле.

 **Список**

Вставляет список выбираемых пользователем элементов. Допускается прокручивание списка, если не все его элементы видны одновременно.

 **Флажок**

Создает ячейку, которая может быть помечена пользователем, как имеющая значение истина или ложь, а также использующуюся для предоставления выбора нескольких вариантов.

 **Переключатель**

Используется для предоставления выбора одного варианта из многих.

 **Выключатель**

Создает кнопку, имеющую два состояния: включено и выключено.

 **Группа**

Позволяет установить графическую или функциональную группировку элементов управления. Для создания группы следует сначала создать ее рамку, а затем внутри нее создать необходимые элементы.

 **Кнопка**

Создает кнопку, при нажатии которой выполняется команда.

 **Набор вкладок**

Позволяет создать несколько страниц в одной и той же области окна или окна диалога.

 **MultiPage**

Служит для представления нескольких экранов информации в виде единого набора.

 **Полоса прокрутки**

Создает графический инструмент для быстрого перемещения по длинным спискам элементов или по большим документам, отображающий текущее положение.

 **SpinButton**

Прокручивающий элемент управления используется совместно с другими элементами для увеличения или уменьшения числовых значений. Допускается его использование для выбора объекта из диапазона значений или из списка элементов.

 **Рисунок**

Отображает в форме точечный рисунок, значок или метафайл.

**ЗАДАНИЯ К ВЫПОЛНЕНИЮ**

***Запустите Excel.***

***Файл сохраните в рабочей папке с именем Lab\_1\_VBA. (Тип файла «Книга Excel c поддержкой макроса».)***

***Откройте окно редактора VBA (Alt+F11).***

 ***Вставьте в проект пользовательскую форму (Insert UserForm).***

 ***и далее выполните:***

1. Расположите на форме следующие элементы:
* Label1;
* TextBox1;
* CommandButton1.
1. Активизируйте Label1, в окне свойств найдите свойство Caption и измените Label1 на название вашего факультета.
2. Те же действия произведите с CommandButton1, изменяя Caption на - "Close".
3. Активизируйте TextBox1 и измените свойство Text, набрав абревиатуру своей группы.
4. С помощью элемента В*ыбор объекта* выделяйте элементы на форме. В окне свойств. изменяя свойства Font (Arial, полужирный, курсив, размер шрифта14), BackColor (светлая тень для кнопки), Visible(False), проследите все изменения элементов управления на форме. (примерный вид формы на рис.1)
5. Осуществить запуск программы с помощью кнопки на панели инструментов или команды **Run Sub/UserForm** меню **Run или F5.**
6. Вернуться в режим редактора VBA для этого нажмите крестик на форме.
7. Самостоятельно изменяйте другие свойства элементов данной формы и пронаблюдайте их изменения.
8. После ознакомления с разделом **Часть 2** доработайте проект, создайте процедуру обработки события для кнопки **Close.**

Private Sub CommandButton1\_Click()

 UserForm1.Hide

End Sub

 **Часть 2**

 **VBA. ЛИНЕЙНАЯ ПРОГРАММА**

Программа (код программы) записывается в окне кода.

Окно кода используется при написании любой программы VBA, будь это код макроса, запуск которого осуществляется при нажатии кнопки в созданной пользователем форме, или подпрограмма.

Под строкой заголовка окна расположены два списка. В первом списке выводятся все объекты модуля, а во втором – список процедур, связанных с выбранным объектом.

Код программы вводится непосредственно в окно кода, так же как текст в любом текстовом редакторе.

Код программы может быть связан с формой -UserForm-(программа пишется для соответствующей формы) так и не связан с ней (пример программ на Паскале). В последнем случае программу пишут в окне модуля.

Для того, чтобы получить окно модуля необходимо выполнить следующие действия:

 *Вставка → Модуль*

Чаще в VBA имеют дело с формой (UserForm). Чтобы получить окно формы необходимо произвести такие действия:

 *Вставка → UserForm*

Если программа пишется под форму, то ее код будет включать ет ряд маленьких подпрограмм. Для каждого события, возникающего в форме необходимо написать процедуру (последовательность совместно выполняемых инструкций, имеющая имя) обработки.

В проекте VBA часто требуется создать собственную форму. Например, может понадобиться задать значения параметров перед выполнением некоторой операции.

В VBA, формы можно создать самостоятельно. Форма - это тоже самое, что и любое диалоговое окно. Панель элементов позволяет разместить ряд элементов управления в форме.

После размещения элементов управления на форме необходимо связать объект на форме с кодом.

В VBA очень просто связать объект с кодом. Для выполнения данной операции:

1. Дважды щелкните по элементу управления в форме. Появляется окно модуля для выбранного объекта. Выберите событие, для которого требуется создать процедуру обработки, в списке, расположенном в верхнем правом углу окна модуля. Введите текст процедуры.
2. Вызвать контекстное меню необходимого объекта правой клавишей мыши и нажать поле *Программа.*

**Решение любой задачи имеет три части:**

1. Ввод данных
2. Обработка данных
3. Вывод результата

 Под вводом данных понимается инициализация (присвоение значений) всех переменных, констант и массивов, используемых в программе.

Вывод результата –отображение полученных результатов: на экране (лист excel, форма), на принтере и т.д.

Для ввода данных, их обработки и вывода создается программа?содержащая необходимые коды.

**Пример создания линейной программы.**

*(Пример выполнить самостоятельно, далее возможно использование для выполнения индивидуального задания )*

Для ввода исходных данных, управления ходом решения и вывода результатов используется пользовательская форма (USER FORM)

***Решим задачу: найдем сумму а + в = с***

Порядок выполнения работы:

1. Открыть окно редактора VBA (Alt+F11……)
2. Выполнить команду ВСТАВКА(Insert)/USER FORM
3. Поместить на форму элементы, требуемые для решения задачи, с панели элементов, и расположить их нужным образом.

Label1, Label2, Label3 соответственно

TextBox1, TextBox2, TextBox3 соответственно

CommndButton1

CheckBox1

1. Изменить свойства объектов на форме c помощью окна свойств.

|  |  |
| --- | --- |
| ***Свойство*** | ***Значение*** |
| Label1.Caption | A |
| Label2.Caption | В |
| Label3.Caption | С |
| СоmmandButton1 | Результат |
| CheckBox1.Caption | Очистка окон |
| Для всех объектов свойство .BackColor | По своему вкусу выбрать цветИз палитры цветов |
|  Для Label1, Label2 ,Label3Свойство **Font** | В диалоговом окне “Шрифт”, которое появится после щелчка по Кнопке с изображением трех маленьких точек, расположенной напротив свойства Font в окне свойств, выбрать размер **16** |

1. Написать программный код. Для этого рекомендуется выполнить двойной щелчок по кнопке *результат* и перейти в окно программы, где набрать текст процедуры обработки события Click() для кнопки и для флажка(CheckBox. (При наборе текста используйте копирование для ввода сходных по содержанию строк

Пояснения к программе:

1) **Dim а As Integer**

Эта инструкция описывает переменные как Integer — целые числа от -32768 и до 32767. При попытке присвоить **а** число, выходящее за пределы этого диапазона, возникает ошибка. При присваивании а дробного числа, выполняется округление.

*Инструкция* **Dim** - Описывает переменные и выделяет для них память.

2) **CInt** - функция преобразования типов данных (преобразовывает выражение в скобках к типу Integer).

Синтаксис **CInt(выражение)**

3) **c=a+b**

 Оператор присваивания ( = )-вычисляется значение выражения, стоящего справа от знака присваивания, и присваивается переменной, стоящей слева от знака присваивания.

4) **MsgBox "результат смотри в TextBox3"**

 Появляется на экране окно сообщений MsgBox, в котором отображается сообщение, записанное в кавычках, и выполнение программы останавливается до тех пор пока не будет нажата кнопка "OK".

5) **TextBox3.Text = c**

 Результат выполнения программы (с) выводится на форму в TextBox3

6) TextBox1.Text = "", TextBox2.Text = "", TextBox3.Text = ""

 Производится очистка полей TextBox1, TextBox2, TextBox3.

7) TextBox1.SetFocus

 Устанавливается фокус (курсор) в TextBox1.

8) **CheckBox1.Value = False**

 Исчезает галочка у флажка CheckBox1.

**Блок-схема к программе**

Ввод а,b

 c=a+b

Вывод с

##  ЗАДАНИЯ ДЛЯ ВЫПОЛНЕНИЯ

1. f(x,y¸z) = (x^2-y^2)/(1+z+x^2).
2. 2.f(x,y,z) = (x+y+z)/(x^2+y^2+z^2).
3. f(x,y)=x./(1+y)+y/(1+x)+1/(x+y).
4. f(x,y,z)=(x+y+z)/(x\*y\*z).
5. f(a,b,c,x)=a\*x^2+b\*x+c.
6. Вычисления процентного отношения двух чисел (сколько процентов составляет величина первого от величины второго).
7. Вычисления дискриминанта квадратного уравнения.
8. f(x,y)=(x+y)(x^2+y^2)(x^3+y^3).
9. f(x,y,z)=(x\*y\*z)/(x+y^2+z^3).
10. f(x,y,z)=x/(y+z)+y/(x+z)+z/(x+y).
11. f(x,y,z)=(x+1)/y/z+sin(z)/y/x+1n(y)/x/z.
12. f(x,y,z)=x^3+y^4+z^5.
13. f(x,y,z)=x\*y/z+y\*z/x+z\*x/y.
14. f(x,y,z)=(x^3+1)/(y+z^2)
15. f(x,y)=(x^2+y^2)/(5\*x\*y)
16. f(x,y)=3\*(x+y)/x\*(x+y)

**СОДЕРЖАНИЕ ОТЧЕТА:**

**1.**     **Название лабораторной работы (Линейные алгоритмы).**

**2.**     **Цель работы (Изучение VBA на примере линейной программы).**

**3.**     **Задание.**

**5.**     **Тестирование программы (предъявить файл с разработанной программой преподавателю)**