

Министерство науки и высшего образования Российской Федерации
Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева

В.В. Буряченко

Информационные технологии в цифровой экономике

Лабораторный практикум по Microsoft Excel 2010

Лабораторная работа №8. ЗНАКОМСТВО С VISUAL BASIC. МАКРОСЫ

Буряченко Владимир Викторович
к.т.н., доцент, доцент каф. ИВТ



Кафедра ИВТ, Л304
BuryachenkoVV@gmail.com
СибГУ им. М.Ф. Решетнева

Красноярск, 2024

ОГЛАВЛЕНИЕ

ЛАБОРАТОРНАЯ РАБОТА №8. Знакомство с VisualBasic. Макросы	3
8. Основные понятия.....	3
Константы.....	5
Библиотеки	5
Модули.....	5
Модули диалоговых окон	5
Подпрограммы «Sub»	5
Функции.....	6
Операция ветвления.....	6
Параметры/Элементы	7
8.1. Автоматическое создание макросов.....	9
8.1.1. Запись макроса	9
8.1.2. Ограничения макрорегистратора	10
8.1.3. Просмотр текста макроса	12
8.1.4. Удаление макросов	12
8.1.5. Переименование макроса	13
8.1.6. Назначение макроса графическому объекту	13
8.1.7. Назначение макроса кнопке	14
8.2. Редактирование макросов.....	17
8.3. Ручное создание макросов.....	20
8.4. Использование элементов управления.....	24
Вопросы для самоконтроля	25

ЛАБОРАТОРНАЯ РАБОТА №8. ЗНАКОМСТВО С VISUALBASIC. МАКРОСЫ

Цель работы: знакомство с средствами программирования и автоматизации работы в MicrosoftExcel при помощи языка программирования Visual Basic и написания макросов.

8. Основные понятия

Макрос (от англ. *macros*, мн.ч. от *macro*) — программный объект, при обработке «развёртывающийся» в последовательность действий и/или команд.

Во многих «офисных» продуктах (Microsoft Office, OpenOffice.org и др.), в текстовых редакторах (например, Vim) при обработке макроса автоматически выполняется заданная для каждого макроса последовательность действий — нажатия на клавиши, выбор пунктов меню и т. д. Предоставляется интерфейс для записи новых и перезаписи существующих макросов.

Макросы используются для автоматизации действий в Excel. Макрос может автоматизировать такие действия, которые иначе потребовали бы длительных ручных манипуляций с возможными ошибками. В настоящее время автоматизированные действия наиболее легко выполняются написанием макросов в **VisualBasicForApplications (VBA)**. Новая среда для макросов в версии Excel должна облегчить использование других языков, но Basic все еще наиболее легкий в использовании. Вот несколько преимуществ использования языка **VisualBasic** для управления Excel:

- легок для изучения;
- поддерживает объекты COM (ActiveX) и расширенные возможности GUI в MicrosoftOffice;
- есть сообщество пользователей в Интернет;
- это — решение для нескольких платформ (Linux, Windows ...).

Задание переменных

В VisualBasic объявление переменных может быть неявным. Объявить переменную можно с помощью инструкции **Dim**. Можно объявлять несколько переменных одновременно, если разделить их имена запятыми. Для определения типа переменной служит либо знак объявления типа после имени, либо соответствующее ключевое слово.

Примеры объявлений переменных

Таблица 1. Объявление переменных

DIM a\$	Объявляет переменную "a" как строку.
DIM a As String	Объявляет переменную "a" как строку.
DIM a\$, b As Integer	Объявляет одну переменную как строку, другую — как целое число.
DIM c As Boolean	Объявляет "c" как логическую переменную, которая может иметь значение либо TRUE, либо FALSE.

Типы переменных

В VisualBasic предусмотрена поддержка четырех классов переменных:

- **Числовые** переменные могут содержать числовые значения. Некоторые переменные служат для хранения больших или малых чисел, остальные используются для чисел с плавающей точкой или дробей.
- **Строковые** переменные содержат строки символов.
- **Логические** переменные содержат одно из значений: TRUE или FALSE.
- **Объектные** переменные могут содержать объекты различных типов, например таблицы и документы внутри документа.

Массивы

В Visual Basic есть одно- и многомерные массивы, определяемые с помощью заданного типа переменной. Массивы применяются для изменения списков и таблиц в программах. К отдельным элементам массива можно обращаться с помощью числового индекса.

Массивы **необходимо** объявлять с помощью инструкции **Dim**. Для определения диапазона индексов массива используется несколько способов.

Таблица 2. Объявление массивов

DIM text\$(20)	21 элемент, пронумерованный от 0 до 20
DIM text\$(5,4)	30 элементов (матрица из 6 x 5 элементов)
DIM text\$(5 to 25)	21 элемент, пронумерованный от 5 до 25
DIM text\$(-15 to 5)	21 элемент (включая 0), пронумерованный от -15 до 5

Диапазон индексов может включать положительные и отрицательные числа.

Константы

Константы имеют фиксированное значение. Они определяются в программе только раз и не могут быть переопределены позднее.

<code>CONST ConstName=Expression</code>

Библиотеки

Библиотеки служат инструментом организации модулей и могут присоединяться к документу или шаблону. При сохранении документа или шаблона все модули, содержащиеся в библиотеке, также автоматически сохраняются.

Библиотека может включать до 16000 модулей.

Модули

Наряду с объявлениями переменных модуль содержит процедуры и функции. Длина программы, которую можно сохранить в модуле, ограничена 64 КБ. Если требуется память большего объема, можно разделить проект VisualBasic между несколькими модулями и затем сохранить их в одной библиотеке.

Модули диалоговых окон

Модули диалоговых окон могут содержать определения диалоговых окон, в том числе свойства диалогового окна, свойства каждого элемента диалогового окна и назначенные события. Так как модули диалоговых окон могут содержать только по одному диалоговому окну, они часто упоминаются как диалоговые окна.

Подпрограммы «Sub»

Sub — краткая форма слова **subroutine** (подпрограмма). Подпрограмма используется для управления конкретной задачей в программе. Подпрограммы служат для разбиения задачи на отдельные процедуры. Разбиение программы на процедуры и подпроцедуры улучшает читаемость и уменьшает возможность возникновения ошибок. Подпрограмма по возможности использует некоторые параметры как аргументы, но не возвращает значения вызывающей подпрограмме или функции, например:

<code>DoSomethingWithTheValues (MyFirstValue, MySecondValue)</code>

Функции

Функция, по существу, является подпрограммой, возвращающей значение. Функцию можно использовать в правой части описания переменной или в других местах, где обычно используются значения, например:

```
MySecondValue = myFunction(MyFirstValue)
```

Операция ветвления

Инструкция **If...Then** используется для выполнения участков программы в зависимости от заданных условий. Когда Visual Basic встречается инструкцию **If**, то проверяется логическое условие. Если оно равно True, то выполняются все дальнейшие инструкции вплоть до следующей инструкции **Else** или **ElseIf**. Если это условие равно False и встречается инструкция **ElseIf**, то VisualBasic проверяет следующее условие и выполняет дальнейшие инструкции, если это условие равно True. Если оно равно False, программа переходит к следующей инструкции **ElseIf** или **Else**. Инструкции после **Else** выполняются только в том случае, если ни одно из предыдущих условий не было равно True. После того как все условия проверены и соответствующие инструкции выполнены, программа переходит к инструкции, следующей за **EndIf**.

Инструкции **If...Then** можно вкладывать одну в другую.

Else и **ElseIf** — необязательные инструкции.

Пример:

```
If a>0 Then  
MsgBox "a>0"  
EndIf
```

Инструкция Do...Loop [время выполнения]. Циклы

Повторяет инструкции между инструкцией Do и Loop, пока условие равно True или пока условие не станет равно True.

Синтаксис

```
Do [{While | Until} Условие = True]  
Участок_программы  
[ExitDo]  
Участок_программы  
Loop
```

ИЛИ

```
Do  
Участок_программы  
[ExitDo]  
Участок_программы
```

```
Loop [{While | Until} Условие = True]
```

Параметры/Элементы

Условие. Сравнение, числовое или строковое выражение, которое вычисляется либо как True, либо как False.

Участок_программы. Инструкции, которые нужно повторять, пока условие равно True или пока условие не станет равно True.

Инструкция **Do...Loop** выполняет цикл, пока определенное условие равно True или пока оно не станет равно True. Условие для выхода из цикла должно быть введено после инструкции **Do** или **Loop**. В следующих примерах представлены допустимые комбинации:

Синтаксис

```
DoWhile Условие = True  
участок_программы  
Loop
```

Участок программы между инструкциями DoWhile и Loop повторяется, пока условие истинно.

```
DoUntil Условие = True  
участок_программы  
Loop
```

Участок программы между инструкциями DoUntil и Loop повторяется, пока условие ложно.

```
Do  
участок_программы  
LoopWhile Условие = True
```

Участок программы между инструкциями Do и Loop повторяется, пока условие истинно.

```
Do  
участок_программы  
LoopUntil Условие = True
```

Участок программы между инструкциями Do и Loop повторяется, пока условие не станет истинно.

Инструкция **ExitDo** используется для безусловного завершения цикла. Эту инструкцию можно добавлять в любое место инструкции **Do...Loop**. Также можно определить условие выхода, используя структуру **If...Then** следующим образом:

```
Do...  
инструкции
```

```
If Условие = True Then Exit Do
инструкции
Loop...
```

Инструкция For...Next [время выполнения]

Повторяет инструкции между For и Next указанное число раз.

Синтаксис:

```
For Счетчик=Начало To Конец [Step Шаг]
Участок_программы
[ExitFor]
Участок_программы
Next [Счетчик]
```

Переменные:

Счетчик. Счетчику циклов первоначально присваивается значение справа от знака равенства (Начало). Допустимы только числовые переменные. Счетчик циклов увеличивается или уменьшается в зависимости от значения переменной Шаг до достижения значения Конец.

Начало. Числовая переменная, которая определяет исходное значение счетчика в начале цикла.

Конец. Числовая переменная, которая определяет последнее значение счетчика в конце цикла.

Шаг. Задаёт значение, на которое увеличивается или уменьшается счетчик цикла. Если параметр Шаг не указан, счетчик цикла увеличивается на 1. В этом случае параметр Конец должен быть больше, чем параметр Начало. Если нужно уменьшать параметр Счетчик, параметр Конец должен быть меньше параметра Начало, а параметру Шаг необходимо присвоить отрицательное значение.

Цикл **For...Next** повторяет все инструкции в цикле число раз, определяемое параметрами.

По мере изменения переменной счетчика VisualBasic проверяет, достигнуто ли значение Конец. После прохождения счетчиком конечного значения цикл автоматически завершается.

Инструкции **For...Next** могут быть вложенными. Если переменная, следующая за инструкцией **Next**, не указана, инструкция **Next** автоматически ссылается на самую последнюю инструкцию **For**.

Если задается приращение 0, инструкции между **For** и **Next** повторяются непрерывно.

При отсчете переменной счетчика VisualBasic проверяет переполнение или потерю значимости. Цикл завершается, если значение Счетчик превышает

значение **Конец** (положительное значение **Шаг**) или становится меньше значения **Конец** (отрицательное значение **Шаг**).

Инструкция **ExitFor** используется для безусловного завершения цикла. Эта инструкция должна находиться внутри цикла **For...Next**. Используйте инструкцию **If...Then** для проверки условия выхода следующим образом:

```
For...  
инструкции  
If Условие = True Then Exit For  
инструкции  
Next
```

Примечание. Во вложенных инструкциях **For...Next**, если производится безусловный выход из цикла с помощью инструкции **ExitFor**, происходит выход только из одного цикла.

Для выполнения лабораторной работы создайте новую рабочую книгу в программе Excel и назовите её «**Лаб8_Фамилия_И.О.**».

8.1. Автоматическое создание макросов

Для того чтобы создавать простые макросы в Excel не требуется самостоятельно писать программный код. В Excel имеется встроенная программа-регистратор, которая может запоминать все действия пользователя с документом и записывать их последовательно в специальный текстовый модуль называемый Макрос.

8.1.1. Запись макроса

Для автоматического создания макроса необходимо выполнить следующие действия

1. Откройте документ, для которого нужно записать макрос.
2. Выберите команды **Разработчик => Макросы => Запись макроса**.
3. Выполните действия, которые хотите записать в документе.
4. Нажмите клавишу ESC, чтобы снять выделение объекта, поскольку в настоящее время макрорегистратор не записывает это действие, произведенное щелчком мыши.
5. Нажмите кнопку «**Завершить запись**».

Откроется диалоговое окно **Запись макроса**, в котором можно сохранить макрос, присвоить ему название и назначить на быстрые клавиши (Рис.1).

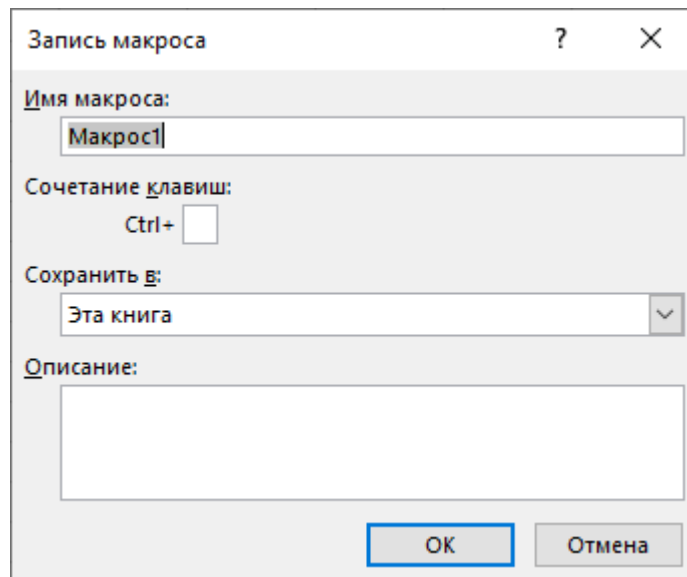


Рис. 8.1. Настройки записи макроса

6. Если нужно прервать запись, не сохраняя макрос, нажмите кнопку **Отмена** в диалоговом окне **Макрос**.

8.1.2. Ограничения макрорегистратора

Следующие действия **нерегистрируются**:

1. Открытие окон не регистрируется.
2. Действия, выполненные в других окнах (в которых регистратор не запущен), не регистрируются.
3. Переключение окон не регистрируется.
4. Действия, не связанные с содержимым документа, не регистрируются. Например, изменения, внесенные в диалоговое окно "Параметры", организатор макросов, настройки.
5. Выделения регистрируются только в том случае, если они выполняются с помощью клавиатуры (перемещением курсора), но не регистрируются при использовании мыши.
6. Макрорегистратор работает только в Excel и Word.

Задание 1

Создайте макрос, который будет выводить на экран столбец с названиями месяцев года.

1. Установите курсор в любую ячейку листа, за исключением A1.
2. Выполните команду **Вид\ Макросы\ Запись макроса**. Появляется окно диалога **Запись макроса**, рисунок 8.1.
3. Введите в поле ввода **Имя макроса** название создаваемого макроса: **Месяц_абс**. В поле ввода **Описание** введите текст: **Вводит названия месяцев**. Введите клавишу быстрого вызова макроса. Для этого перейдите в поле ввода **Ctrl**, переключитесь на латинский регистр, удалите находящуюся

там букву и введите букву **m**. После этого нажмите кнопку **ОК**. Начиная с этого момента, осуществляется запись макроса.

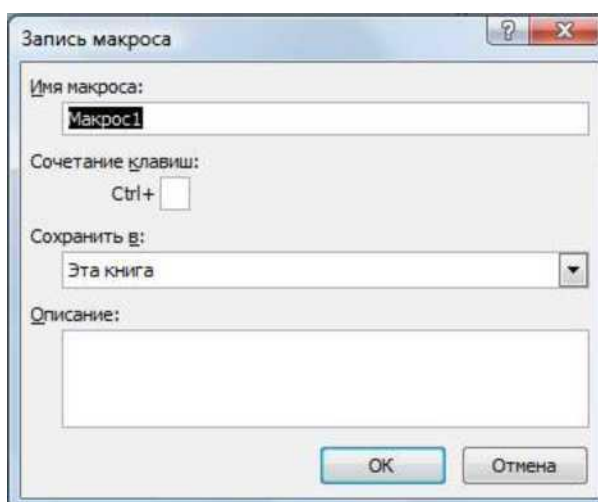


Рисунок 8.2 - Окно записи макроса

4. Выполните последовательность действий, которую будет выполнять макрос: установите курсор в ячейку A1; введите слово январь; поместите указатель мыши в правый нижний угол ячейки A1, при этом указатель мыши изменит вид на черный крестик; нажмите кнопку мыши и, удерживая ее нажатой, продолжите выделение до ячейки A12.

5. Для остановки записи макроса выполните **Вид\ Макросы\ Остановить запись**.

***Примечание:** если не будет выполнено условие первого шага, то есть курсор будет находиться в ячейке A1, то макрос будет правильно работать только в том случае, когда перед нажатием клавиш [Ctrl]+[m] поместите курсор будет в ячейке A1.*

Листу1 присвойте имя **Месяц**. Макрос записан и теперь нажатие клавиш [Ctrl]+[m] вызовет появление в ячейках A1:A12 названий месяцев. При записи макроса не был включен режим записи с относительными ссылками, поэтому названия месяцев будут появляться только в **ячейках A1:A12** (созданный макрос является макросом с **абсолютными ссылками**). Для устранения этого неудобства предназначен режим записи с **относительными ссылками**.

8.1.3. Просмотр текста макроса

Задание 2. Просмотрите текст макроса Месяц_абс.

6. Для входа в режим редактирования макроса выполните команду **Вид\Макросы\ Макросы**. Появляется окно диалога **Макрос**, рисунок 8.2, в котором после выбора нужного для редактирования макроса становятся доступными кнопки, в том числе кнопка **Изменить**.

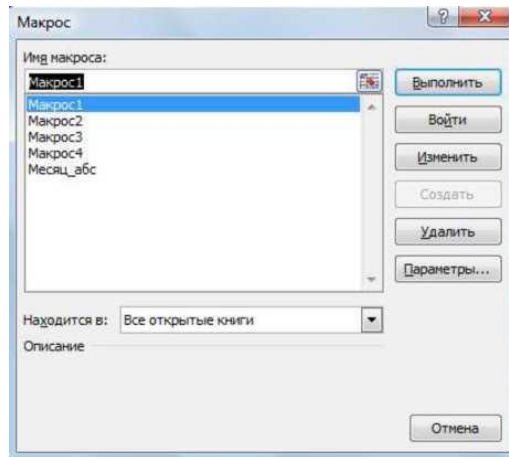


Рисунок 8.3 - Список макросов

7. Нажмите кнопку **Изменить**, рисунок 8.3. Появится текст макроса Месяц_абс, записанный на языке VisualBasic:

```
SubМесяц_абс()  
,  
' Месяц_абс Макрос ' Вводит названия месяцев  
,  
' Сочетание клавиш: Ctrl+m  
Range("A1 ").Select  
ActiveCell.FormulaR1C1 = "Январь"  
Range("A1 ").Select  
Selection.AutoFill Destination:=Range("A1:A12"), Type:=xlFillDefault  
Range("A1:A12").Select  
EndSub
```

Таким образом, появляется возможность изменять текст программы, записанной на языке Visual Basic for Application, что позволяет писать собственные программы.

8.1.4. Удаление макросов

Если макрос назначен клавише быстрого доступа, графическому объекту или кнопке, то процесс удаления является стандартным. Необходимо выполнить **Вид\ Макросы\ Макросы**. Появляется окно диалога **Макрос**,

содержащее список макросов текущей рабочей книги. После выбора макроса, который предполагается удалить, станут доступными кнопки окна диалога **Макрос**, в том числе кнопка **Удалить**. Для удаления выбранного макроса достаточно нажать эту кнопку.

8.1.5. Переименование макроса

Специальной команды для переименования макроса нет, а часто возникает потребность исправить имя макроса или присвоить ему другое имя. Для этого нужно войти в режим редактирования макроса и в тексте программы исправить заголовок.

Задание 3. Измените имя макроса **Месяц_абс** на **Месяцы_с_абсолютными_адресами**.

8. Выполните **Вид\ Макросы\ Макросы** и в появившемся окне диалога **Макрос** выберите макрос **Месяц_абс** и нажмите клавишу **Изменить**. Первая строка является комментарием.

9. Замените **Месяц_абс** на **Месяцы_с_абсолютными_адресами**. Соответствующим образом измените комментарий. В результате получим новый текст программы макроса:

```
SubМесяцы_с_абсолютными_адресами()  
,  
' Месяцы_с_абсолютными_адресами Макрос  
' Вводит названия месяцев  
,  
' Сочетание клавиш: Ctrl+m  
,  
  
Range("A1 ").Select ActiveCell.FormulaR1C1 = "Январь"  
Range("A1 ").Select  
Selection.AutoFill Destination:=Range("A1:A12"),Type:=xlFillDefault  
Range("A1:A12").Select  
EndSub
```

Новое имя автоматически заменит старое в списках макросов. По клавише быстрого вызова будет вызываться макрос с новым именем. Но не все замены происходят автоматически. Если макросу были назначены графические объекты или кнопки, то нужно самим сделать для них переназначения.

8.1.6. Назначение макроса графическому объекту

Макрос может вызываться не только нажатием клавиш быстрого вызова, но и нажатием кнопки мыши на графическом объекте. Для

назначения макроса графическому объекту необходимо наличие такого объекта на экране.

Задание 4. Назначьте макрос **Месяц_абс** графическому объекту.

10. Чтобы нарисовать закрашенный овал или прямоугольник, выполните **Вставка\ Иллюстрации\ Фигуры \ Основные фигуры**. Выберите место для объекта. Поместите указатель мыши в нужное место таблицы и, удерживая нажатой кнопку мыши, нарисуйте объект — прямоугольник или овал. Для выбора заливки объекта нажмите правую кнопку мыши на активном объекте. Появится меню, в котором выберите пункт **Формат фигуры**. Появится окно диалога **Формат фигуры**, в котором выбирается цвет и вид заливки. Для ввода надписи нажмите правую кнопку мыши на активном объекте. Появится меню, в котором выберите пункт **Изменить текст**. Введите требуемый текст и нажмите кнопку мыши вне текстового окна. Построен графический объект, рисунок 8.4.

	A	B	C	D
1	Январь			
2	Февраль			
3	Март			
4	Апрель			
5	Май			
6	Июнь			
7	Июль			
8	Август			
9	Сентябрь			
10	Октябрь			
11	Ноябрь			
12	Декабрь			
13				




Рисунок 8.4 - Назначение макроса графическому объекту

11. Назначим макрос созданному графическому объекту. нажмите правую кнопку мыши на графическом объекте. Появится меню, в котором выберите пункт **Назначить макрос**. Появится окно диалога **Назначить макрос объекту** со списком макросов. Выберите требуемый макрос и нажмите на нем кнопку мыши. Имя макроса появится в верхней строке окна диалога. Нажмите кнопку ОК. Произойдет выход из режима назначения макроса. Нажмите кнопку мыши в любом месте вне объекта.

Макрос назначен графическому объекту. Если теперь поместить указатель мыши на этот объект, то белый крестик перейдет в изображение руки с отогнутым указательным пальцем. Нажатие левой кнопки мыши приводит к выполнению макроса, который назначен этому объекту.

8.1.7. Назначение макроса кнопке

Рассмотрим назначение макроса кнопке. Эта кнопка может быть пользовательской, стандартной кнопкой Excel или кнопкой, расположенной на листе.

Для ее расположения на листе нужно отобразить еще одну вкладку: **Разработчик**. Для ее отображения необходимо щелкнуть по кнопке **MicrosoftOffice** и выбрать команду **Параметры Excel**. Затем во вкладке **Основные** диалогового окна **Параметры Excel** установить соответствующий флажок. На вкладке **Разработчик** собраны средства создания макросов и форм, а также функции для работы с XML. Файлы и шаблоны предыдущих версий Excel могли содержать пользовательские панели инструментов. В этом случае при открытии таких файлов в Excel2007 появляется еще одна постоянная вкладка - **Надстройки**.

Сначала необходимо нарисовать кнопку, а затем назначить ей макрос. Для этого выполнить **Разработчик Элементы управления\ Вставить**. На появившейся панели инструментов **Элементы управления формы**, рисунок 9.4, нажать кнопку, соответствующую добавляемому элементу управления (например элемент Кнопка).

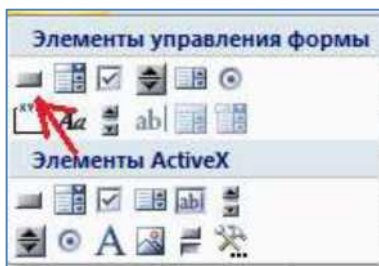


Рисунок 8.5 - Панель элементов управления

Установить курсор в то место, где будет располагаться кнопка и, удерживая нажатой кнопку мыши, выделить прямоугольник, где будет расположена кнопка. После осуществления нажатия на кнопку мыши появится окно диалога **Назначить макрос объекту**. Надо выбрать требуемый макрос из списка **Имя макроса**.

Чтобы изменить надпись на кнопке надо нажать правую кнопку мыши на созданной кнопке и в появившемся меню выбрать **Изменить текст**, ввести вместо номера кнопки требуемый текст. Эта операция нужна для входа в режим форматирования надписи на следующем шаге. Затем нажать правую кнопку мыши на созданной кнопке, в появившемся меню выбрать команду **Формат объекта**. Появится окно диалога форматирования. Теперь следует выбрать вкладки **Выравнивание** и **Шрифт** и установить там нужные параметры надписи на кнопке. Теперь можно нажать кнопку ОК для выхода из режима форматирования. Кнопка с назначенным ей макросом создана.

Упражнение 9.5. Закрепите макрос **Заголовок** за кнопкой на рабочем листе.

12. Воспользуемся приведенной выше последовательностью действий для создания кнопки на рабочем листе. Приведем пример назначения макроса **Заголовок** кнопке на рабочем листе с надписью **Заголовок**. Сначала создадим

кнопку с изображением надписи **Заголовок**, а затем назначим ей макрос **Заголовок**.

13. Нажмите кнопку **Кнопка** на панели инструментов **Элементы управления формы**.

14. Укажите место, где будет располагаться кнопка.

15. В окне диалога Назначить макрос объекту выберите макрос **Заголовок**, и нажмите кнопку **ОК**.

16. Нажмите правую кнопку мыши на созданной кнопке и вместо текста с номером кнопки введите слово **Заголовок**.

17. Нажмите кнопку мыши вне созданной кнопки.

18. Нажмите правую кнопку мыши на созданной кнопке. Появится меню со списком команд.

19. Выберите **Формат объекта**. Появится окно диалога форматирования.

20. Выберите вкладку **Выравнивание** и установите переключатель горизонтального и вертикального выравнивания в положение по центру. Выберите вкладку **Шрифт** и установите стиль надписи на кнопке **Arial** и размер 14.

21. Нажмите кнопку **ОК**. Нажмите кнопку мыши вне кнопки.

Задание 5.

1. Создайте новый лист и переименуйте лист2 в «**Автозаполнение**»;
2. Заполните таблицу как на рисунке (Рис. 8.6)

	А	В	С
1	1	Понедельник	Январь
2	2	Вторник	Февраль

Рис. 8.6 Таблица для макроса автозаполнения

3. Выполните команду **Разработчик => Макросы => Запись макроса**;
4. Сохраните макрос с именем «Задание5» в файле лабораторной работы.
5. Назначьте горячую клавишу для макроса. Сочетания клавиш и макросы применяются только для открытого сейчас документа и сохраняются совместно с файлом документа (рис. 8.7).

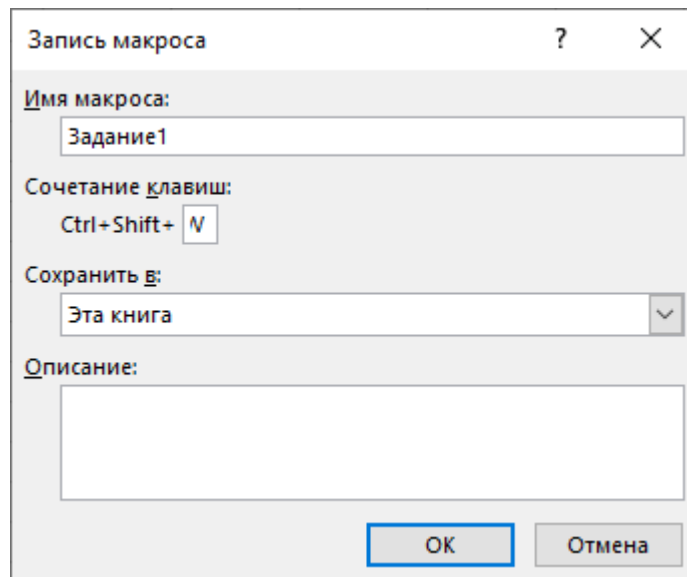


Рис. 8.7. Создание макроса автозаполнения

6. С помощью автозаполнения заполните первую колонку до 10;
7. Нажмите кнопку «**Остановить запись**».
8. Очистите в первой колонке ячейки до исходного состояния, т.е. оставьте заполненными только первую и вторую ячейки колонки;
9. Установите курсор в ячейку **A1** и выполните команду **Разработчик => Макросы => Выполнить макрос** или воспользуйтесь заданным сочетанием клавиш.

В результате у вас должны автоматически заполниться ячейки с 3 по 10.

Далее мы усовершенствуем свой макрос, добавив ему возможность автоматически заполнять нужное количество ячеек методом автозаполнения в той колонке, в которой установлен курсор.

8.2. Редактирование макросов

Как мы видим, в результате выполнения упражнения 1 макрос получился очень ограниченный. Он автоматически заполняет только значения в колонке **A** и только до десяти. Для того чтобы сделать макрос универсальным, необходимо воспользоваться встроенным редактором разработки макросов и вручную добавить ему необходимые возможности.

Задание 6.

1. Создайте новый макрос и назовите его **Задание2**.
2. Запустите редактор макросов командой **Макросы => НазваниеМакроса => Изменить** или **Разработчик => VisualBasic**;
3. В окне редактора найдите процедуру **SubЗадание2()** (Рис. 8.8). Именно эта процедура запускается на выполнение, когда вы запускаете макрос;

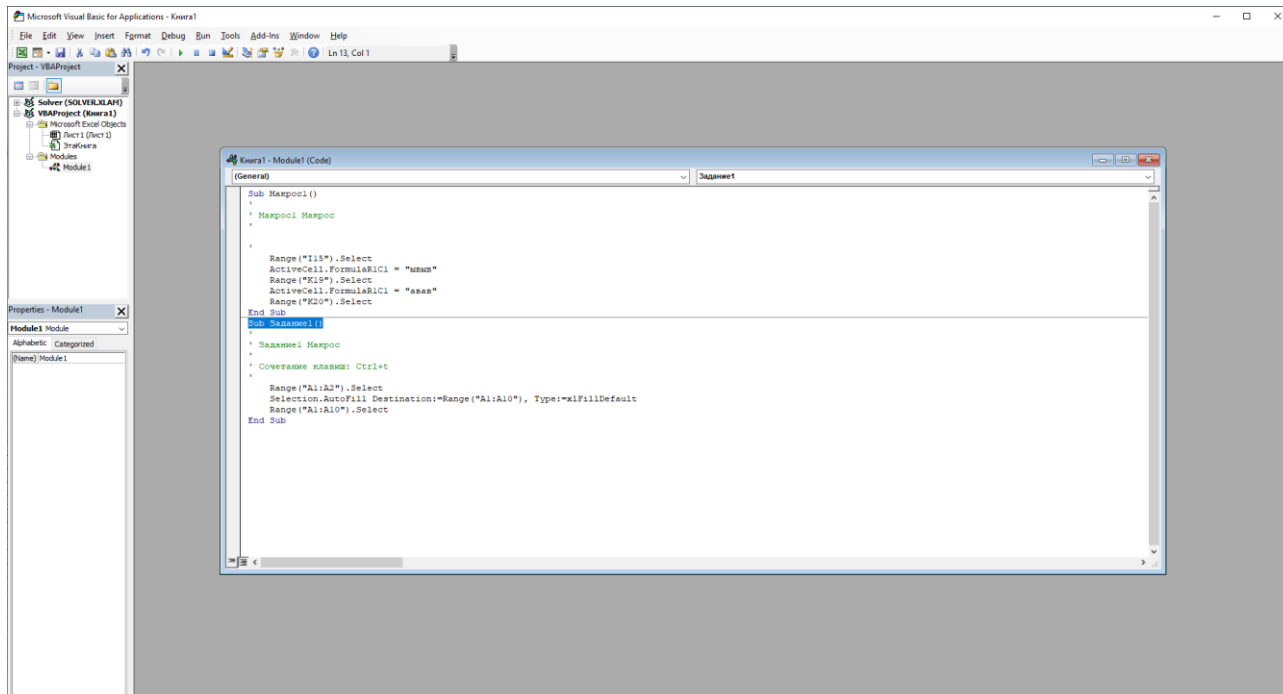


Рис. 8.8. Окно редактора макроса Microsoft Visual Basic for Applications

4. Отредактируйте процедуру примернотак, как предложено в листинге 8.2.1 и рис. 8.9 (комментарии писать не обязательно).

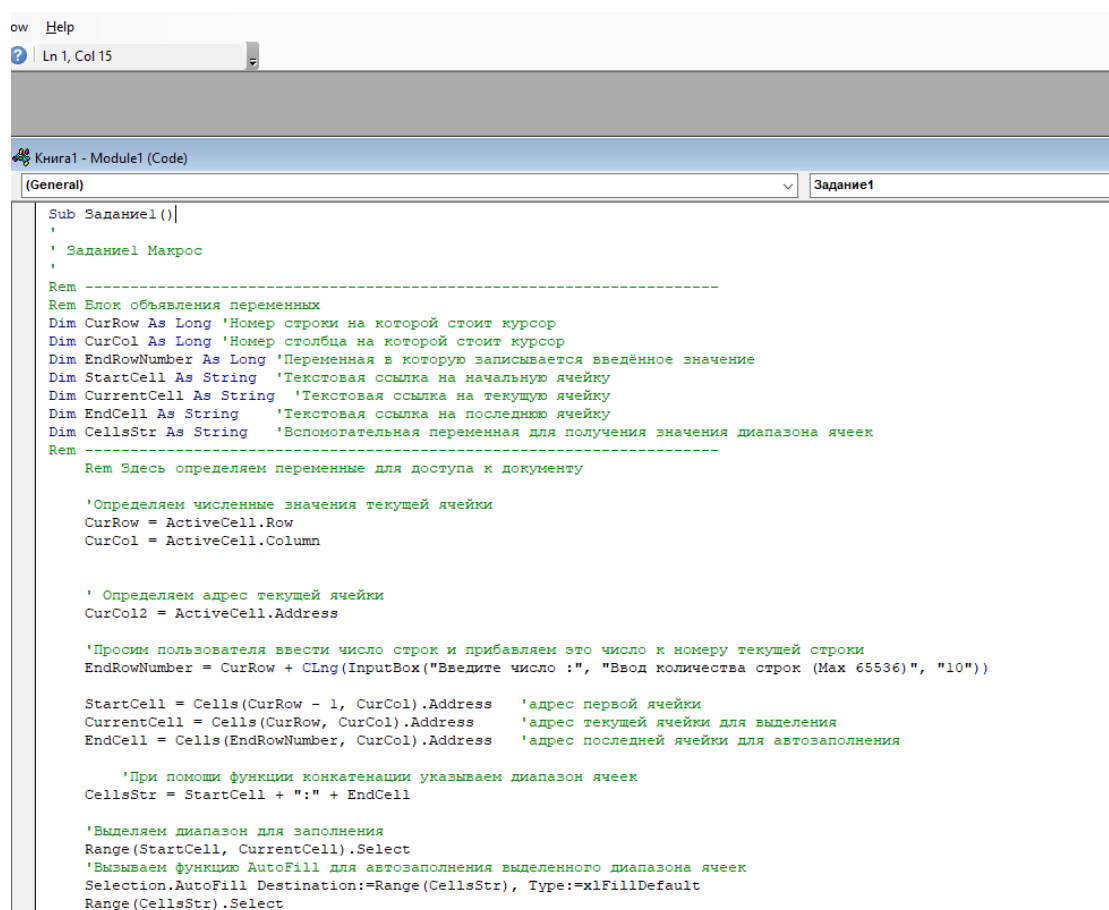


Рис. 8.9. Создание универсального макроса автозаполнения

```

Sub Задание2 ()
'
' Задание2 Макрос
'
Rem -----
Rem Блок объявления переменных
Dim CurRow As Long'Номер строки на которой стоит курсор
Dim CurCol As Long'Номер столбца на которой стоит курсор
Dim EndRowNumber As Long'Переменная в которую записывается введённое значение
Dim StartCell As String'Текстовая ссылка на начальную ячейку
Dim CurrentCell As String'Текстовая ссылка на текущую ячейку
Dim EndCell As String'Текстовая ссылка на последнюю ячейку
Dim CellsStr As String'Вспомогательная переменная для получения значения диапазона ячеек
Rem -----
Rem Здесь определяем переменные для доступа к документу

'Определяем численные значения текущей ячейки
CurRow = ActiveCell.Row
CurCol = ActiveCell.Column

' Определяем адрес текущей ячейки
CurCol2 = ActiveCell.Address

'Просим пользователя ввести число строк и прибавляем это число к номеру текущей строки
EndRowNumber = CurRow + CLng(InputBox("Введите число :", "Ввод количества строк (Max 65536)", "10"))

StartCell = Cells(CurRow - 1, CurCol).Address'адрес первой ячейки
CurrentCell = Cells(CurRow, CurCol).Address'адрес текущей ячейки для выделения
EndCell = Cells(EndRowNumber, CurCol).Address'адрес последней ячейки для автозаполнения

'При помощи функции конкатенации указываем диапазон ячеек
CellsStr = StartCell + ":" + EndCell

'Выделяем диапазон для заполнения
Range(StartCell, CurrentCell).Select
'Вызываем функцию AutoFill для автозаполнения выделенного диапазона ячеек
Selection.AutoFill Destination:=Range(CellsStr), Type:=xlFillDefault
Range(CellsStr).Select
End Sub

```

5. Изучите программный код, обратив внимание на комментарии и основные использованные операторы.

6. Сохраните полученный макрос командой **Файл => Сохранить**, в редакторе VisualBasic;

7. Для назначения макросу комбинации клавиш выполните команду **Разработчик =>Макросы** и откройте вкладку **Параметры**;

8. Выберите удобную комбинацию клавиш (обычно ctrl+shift+ любая буква) и назначьте на неё макрос.

9. Используя комбинацию клавиш, назначенную на макрос, заполните список дней недели и месяцев года (Рис. 8.10). Обратите внимание на то, какая ячейка должна быть выделена при работе макроса.

10. Сохраните документ.

	A	B	C	D
1	1	Понедельник	Январь	
2	2	Вторник	Февраль	
3	3	Среда	Март	
4	4	Четверг	Апрель	
5	5	Пятница	Май	
6	6	Суббота	Июнь	
7	7	Воскресенье	Июль	
8	8	Понедельник	Август	
9	9	Вторник	Сентябрь	
10	10		Октябрь	
11	11		Ноябрь	
12	12		Декабрь	
13			Январь	
14			Февраль	

Рис 8.10. Результат выполнения макроса

Исследовательское задание

Самостоятельно изучите диалог настройки (**Настройка ленты**), назначьте макрос «**Задание6**» на панель инструментов и создайте новое меню **Макросы**, в которое также назначьте макрос.

Совет: Используйте справку Excel. для выполнения заданий. (Клавиша **F1**, или меню **Справка=> Справка по Excel**)

8.3. Ручное создание макросов

Задание 7

1. Откройте редактор VisualBasic, используя меню **Разработчик=>Код=>VisualBasic**, чтобы начать работу с редактором VBA. С левой стороны окна диалога найдите документ, который Вы только что открыли. Нажмите кнопку «**Создать**» (Insert Module) для создания нового модуля.

2. Откроется среда редактирования и отладки макросов VisualBasicIDE. Введите (или скопируйте) текст, приведенный в Листинг 8.3.1.

Листинг 8.3.1: Ваш макрос, “Helloworld”

```
Sub Main
    Print"Hello World"
End Sub
```

3. Попробуйте выполнить макрос и увидеть результат работы.
4. Сохраните документ.

Задание 8

VisualBasic основан на процедурах и функциях, которые задаются ключевыми словами Sub и Function – далее они будут называться процедурами или соответственно функциями. Каждая процедура может вызывать другие процедуры.

Каждый модуль (module) содержит набор процедур (функций). Библиотека (Library) содержит набор модулей. Документ (document) может содержать библиотеку или несколько библиотек. Библиотека может существовать также на уровне Приложения (applicationlevel), такого как MicrosoftExcel.

Разница между Sub и Function в том, что функция возвращает значение, а процедура – нет. Макрос на Листинг 8.3.2 получает текстовую строку от функции с именем **HelloWorldString**.

1. Создайте самостоятельно макрос, состоящий из процедуры и функции (см. Листинг 3.2.).

Листинг 8.3.2: “Helloworld” с использованием процедуры и функции

```
Sub HelloWorld
Dim s As String
    s=HelloWorldString()
    MsgBox s
EndSub

Function HelloWorldString() As String
    HelloWorldString="Hello World"
End Function
```

2. Усовершенствуйте макрос, добавив в него возможность ввода пользователем своего имени, чтобы программа приветствовала создателя.

3. Подсказка: используйте функцию **InputBox()** как в предыдущем разделе.

Задание 9

Написать программу на VisualBasic для функции пользователя, которая вычисляет получаемую потребителем электроэнергию QН по количеству электроэнергии Q, которое вырабатывается производителем, и величине потерь Р (%) при ее передаче: $Q_H = Q \cdot (100 - P) / 100$. Заполнить ячейку C2 следующей таблицы с использованием построенной функции пользователя (рис. 8.11).

	A	B	C	D
1	Выработанная электроэнергия, ГДж	Потери, %	Получаемая электроэнергия, ГДж	
2	5750	5%		

Рис 8.11. Данные для расчетов электроэнергии

Решение задания 9:

На листе 2 в Excel создайте новую электронную таблицу. Воспроизведем данные, приведенные выше. Вызовите редактор VisualBasic: **Разработчик** => **Код** => **VisualBasic**. Выберите в диалоге выбора макроса пункт, соответствующий названию документа, и нажмите кнопку «Создать». В появившемся окне напишите следующий текст программы:

Листинг 8.3.3: Создание функции QH для расчета получаемой электроэнергии

```
Function QH(Q, P)
QH=Q*(100-P)/100
End Function
```

Далее необходимо вернуться в окно Рабочего листа 1, и при помощи мыши выбрать ячейку C2, в которую необходимо внести число, вычисляемое по только что созданной пользовательской функции. Для этого при вводе формул нужно написать следующую строку:

=QH(A2;B2)

где QH название только что написанной функции, A2 адрес ячейки для аргумента Q, B2 адрес ячейки для аргумента P. В результате в ячейку C2 запишется значение функции QH, и Лист 2 примет вид:

	A	B	C	D	E	F	G
1	Выработанная электроэнергия, ГДж	Потери, %	Получаемая электроэнергия, ГДж				
2	5750	5%	5462,5				
3							
4							
5							
6							
7							

Рис. 8.12. Результат вычисления функции

Задание 10

Написать программу на VisualBasic для следующей функции пользователя, используя ее дополнить таблицу вычисленными значениями и построить график зависимости **y=FY(x)**:

$$y = \frac{1 + |0,2 - x|}{1 + x + x^2} + \sin(x) + \ln(x + 2) - \arctg(x^3 + 1) + e^{-x} - tg(x^{3,13}) + \sqrt{x} + \cos(x + 1)$$

Таблица 3

X	0,10	0,15	0,20	0,25	0,30	0,35	0,40	0,45	0,50
Y									

Решение задания 10:

1. Напишите тело функции из нескольких строк:

Листинг 8.3.4

```
Function FY(x)
    a=1+Abs(0.2-x)
    b=1+x+x*x
    FY=a/b+Sin(x)
    FY=FY+Log(x+2)
    FY=FY-Atn(x^3+1)
    FY=FY+Exp(-x)-Tan(x^3.13)
    FY=FY+Sqr(x)+Cos(x+1)
End Function
```

2. Далее проведите расчеты с использованием данной функции пользователя и заполните таблицу. Не следует заполнять все ячейки значений аргумента x , вводя их с клавиатуры, так как это будет нерационально. Вместо этого введите два значения в ячейки **B1** и **C1**, представляющие собой первый и второй члены арифметической прогрессии. Для ввода следующих значений воспользуйтесь функцией автозаполнения или формулами.

Примечание:

В некоторых версиях Excel в VBA недопустимо использование оператора “^” для возведения в степень. В этом случае можно использовать функцию Power:

Application.WorksheetFunction.Power(основание, степень)

3. Для построения графика зависимости $y=FY(x)$ выберите **Вставка>Диаграмма**. Выберите тип диаграммы «График», «График с маркерами». Выберите нужный диапазон данных и настройте отображение названия диаграммы, осей и значений.

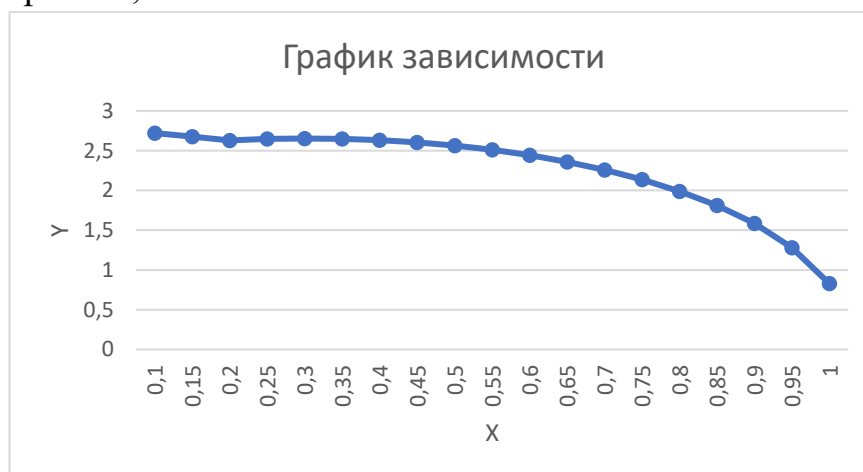


Рис. 8.13. Построение графика зависимости

4. Можете самостоятельно поработать с этим графиком с целью улучшения его вида.

Выполните самостоятельно задания, приведенные ниже.

Составить программу на VisualBasic для определения функции пользователя $y=y(x)$, заполнить таблицу значений функции при изменении аргумента в самостоятельно выбранном диапазоне и построить график зависимости $y=y(x)$:

Задание 11

$$y=x$$

Задание 12

$$y=x^2+\ln x+10,46 - \operatorname{tg} x$$

Задание 13

$$y=x^{25}+\ln |x+10,46| - \operatorname{tg} x$$

Задание 14

Создайте новый лист **НДС**. Постройте пользовательскую функцию для вычисления стоимости товаров с НДС по его стоимости без НДС и ставки НДС. С использованием этой функции заполните таблицу, отражающую стоимость товаров без НДС и с НДС. Исходные данные выберите самостоятельно. Таблица может выглядеть следующим образом:

Книга1.xlsm * x			
	A	B	C
1		Стоимость	Стоимость с НДС
2	Видеокарта	35 567,00 Р	
3	Блок питания	2 400,00 Р	
4	Процессор	12 800,00 Р	
5	Клавиатура	2 400,00 Р	
6	Монитор	12 500,00 Р	
7	Системный блок	3 490,00 Р	
8	Итого	69 157,00 Р	
9			
10	Сумма НДС	20%	

Рис. 8.14. Расчёт стоимости товаров с НДС

8.4. Использование элементов управления

Мы привыкли использовать элементы управления в диалоговых окнах, формы в динамических Web-страницах. Элементы управления – это флажки (CheckBox), переключатели (OptionButton), раскрывающиеся списки (ListBox),

счетчики (SpinButton), полосы прокрутки, кнопки (PushButton), поля для ввода текста (TextBox).

В электронных таблицах удобно использовать элементы управления для быстрого ввода информации при формировании какого-нибудь документа. Данные, полученные из форм, используются как исходная информация для последующей обработки с помощью обычных формул и функций.

Задание 15

Оператору, принимающему заказы на доставку пиццы или другой продукции по телефону, необходимо быстро сформировать бланк заказа, рассчитать стоимость заказа с учетом цен, скидок и других данных, распечатать бланк (рис. 8.15).

В поля для ввода текста вносится информация о заказчике. В определенных ячейках введено текущее время и дата заказа.

Данные об ассортименте продукции внесены в обычный список, содержащий три колонки (номер по порядку, наименование, цена). Различные варианты скидок или надбавок можно внести в определенные ячейки, а можно использовать прямо в формулах.

Список выбора (форма) формируется по диапазону ячеек с наименованием пиццы. Различные опции, скидки, надбавки указываются с помощью переключателей или флажков. Очистка бланка делается нажатием кнопки.

Вид пиццы	Количество товаров
Четыре сыра Американо Гавайская С ананасами Фирменная	2
	Кассир Иванов И.М. Семенова К.А. Абакумова Н.И.
Скидки <input checked="" type="checkbox"/> День рождения <input type="checkbox"/> Постоянный клиент <input type="checkbox"/> Тихий час	Рассчитать стоимость
Приготовление <input type="radio"/> В зале <input type="radio"/> С доставкой <input checked="" type="radio"/> На вынос	Итоговая стоимость 234,90 Р
	Кассир Иванов И.М.

Рис. 8.15. Пример интерфейса для заполнения параметров при заказе пиццы

1. Создайте новый лист «**Заказ пиццы**».
2. Для вставки элементов управления можно использовать панель инструментов **Формы** на панели Разработчик Элементы => управления (Рис.8.16).

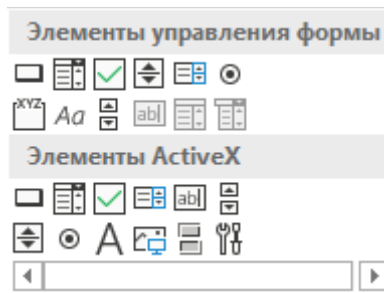




Рис. 8.16. Элементы управления

3. При компоновке документа удобно использовать визуальное проектирование формы. Для этого на панели инструментов  необходимо нажать на кнопку **Режим конструктора**.

4. Затем, выбрав необходимый элемент управления расположить его в нужном месте таблицы.

Например, мы хотим использовать флажок для указания срочности доставки. Если флажок включен – доставка срочная, за срочность установлена надбавка от стоимости заказа.

5. Выбираем  элемент Флажок. Определяем его в нужное место документа. Затем в контекстном меню выбираем **Элемент управления**.

6. Во вкладке **Свойства** можно определить текст подписи, цвет фона.

7. Во вкладке **Формат объекта** => **Формат элемента управления** можно установить привязку к ячейке электронной таблицы (рис. 8.17).

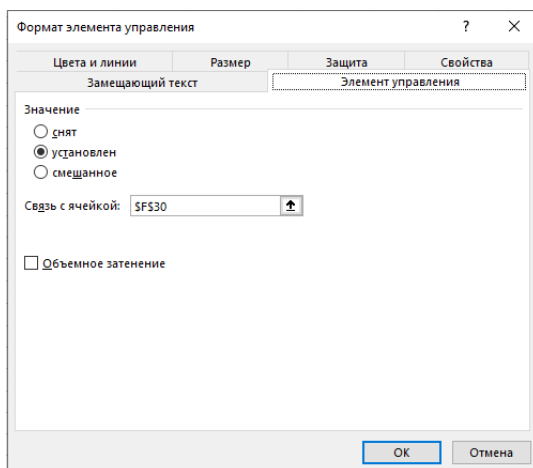


Рис. 8.17. Создание связи элемента с ячейкой

Удобно все возвращаемые из элементов управления значения связывать с ячейками одного столбца, который потом можно скрыть.

8. Если во вкладке «данные» не указывать значение индекса, например 1, если флажок установлен и 0, если флажок снят, то в связанной ячейке будет значение ИСТИНА, если флажок поставлен, или ЛОЖЬ, если флажок снят.


9. Для установки переключателей сначала нажимаем на кнопку на панели **Элементов управления** и выбираем **Группа**, после чего на неё можно добавлять **Переключатели**.

10. Далее указываем, какие значения будут при выборе одного из переключателей, и как будет называться окно группы.

11. Затем обязательно в контекстном меню выбираем **Элемент управления** и во вкладке **Данные** указываем связанную ячейку. Для обоих переключателей связанная ячейка одна, поэтому она принимает значения 1 или 2 в зависимости от выбора переключателя. Особенность состоит в том, что возвращаемые значения индекса при установке флажка и номер установленного переключателя задаются в текстовом формате.



Рис. 8.18. Создание группы с переключателями

12. Для того, чтобы раскрывался список имеющихся видов продукта, выбираем на панели инструментов элемент **Список** .

13. Необходимо в свойствах списка (вкладка **Элемент управления**) указать связанную ячейку и диапазон ячеек, по которому список формируется. Адреса ячеек вводятся вручную. Исходный диапазон ячеек – это таблица с наименованием пиццы и ценой.

14. Для печати сформированного бланка будем использовать выделенный диапазон. Ориентация страницы и диапазон для печати устанавливаются в меню **Формат**.

15. Для выбора значений из диапазона данных можно пользоваться функцией **Выбор** (рис. 8.19).

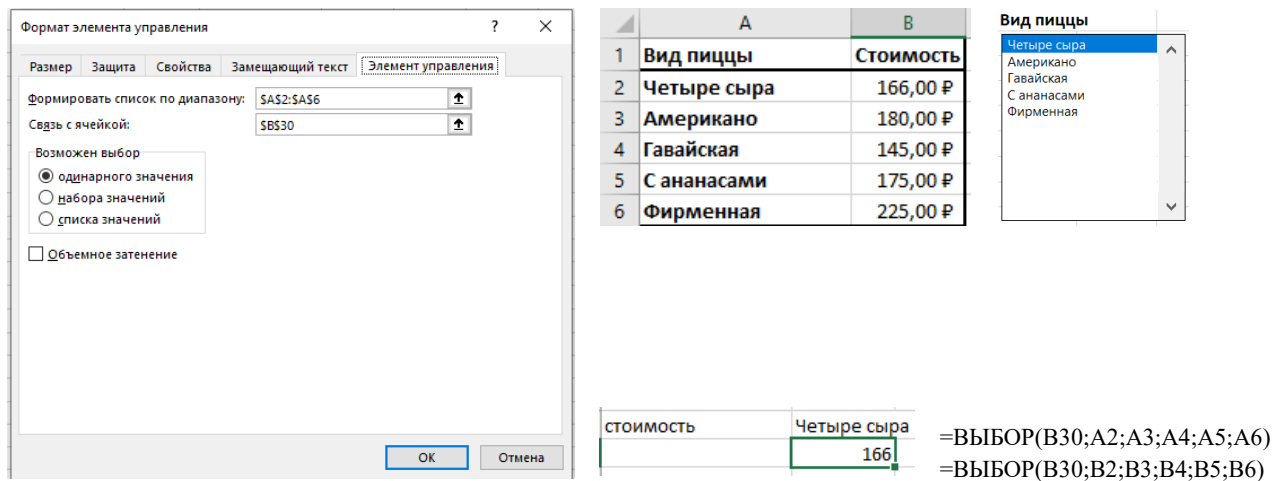


Рис. 8.19. Настройка списка значений для выбора вида пиццы

16. Учет скидок можно выполнять с использованием функции **Если**, либо при написании соответствующей функции в VBA.

17. Вычисление итоговой стоимости товаров нужно выполнять по нажатию кнопки **Рассчитать стоимость**, которая вызывает выполнение соответствующей функции либо формулы в Excel, и учитывает значения, представленные на рис. 8.15.

18. Используя панель рецензирования заблокируйте часть листа со справочными данными для редактирования. Итоговый результат (ваш вариант может отличаться от примера) представлен на рис. 8.20.

	A	B	C	D	E	F	G
1	Вид пиццы	Стоимость					
2	Четыре сыра	166,00 Р					
3	Американо	180,00 Р					
4	Гавайская	145,00 Р					
5	С ананасами	175,00 Р					
6	Фирменная	225,00 Р					
7							
8	Количество товаров						
9							
10							
11	Скидки						
12	День рождения	10%					
13	Постоянный клиент	15%					
14	Тихий час	12%					
15							
16	Приготовление						
17	В зале	110%					
18	С доставкой	130%					
19	На вынос	90%					
20	2						
21	Кассир						
22	Иванов И.М.						
23	Семенова К.А.						
24	Абакумова Н.И.						

Вид пиццы

Четыре сыра

Американо

Гавайская

С ананасами

Фирменная

Количество товаров

3

Кассир

Иванов И.М.

Семенова К.А.

Абакумова Н.И.

Скидки

☒ День рождения

☐ Постоянный клиент

☐ Тихий час

Приготовление

☐ В зале

☒ С доставкой

☐ На вынос

Рассчитать стоимость

Итоговая стоимость

789,75 Р

Кассир

Иванов И.М.

Рис. 8.20. Пример создания формы заказа пиццы для кассира

Вопросы для самоконтроля

1. Для чего предназначены макросы?
2. Как изменить имя макроса?
3. Какой самый простой способ записать макрос?
4. На каких языках программирования можно создавать макросы?
5. Возможно ли изменить макрос, созданный автоматически?
6. Как назначить макрос на панель инструментов?
7. Как назначить макросу горячие клавиши?
8. Что такое процедура?
9. Что такое функция?
10. Что такое модуль?
11. Как открыть макрос для редактирования?
12. Для чего предназначаются переменные?
13. Какая разница между процедурой и функцией?
14. Можно ли назначить макрос на событие?
15. На какие события можно назначить макрос?
16. Какие виды циклов вы знаете?
17. Что такое операция ветвления?
18. Перечислите основные объекты электронной таблицы?
19. Возможно ли создать свою (пользовательскую функцию)?
20. Какие основные типы переменных существуют в VisualBasic?
21. Перечислите элементы управления в ExcelVBA?
22. Для чего служат элементы управления?
23. Для чего предназначена форма?
24. Опишите как использовать значение, выбранное в элементе Список для вычислений.
25. Опишите как сделать элемент Переключатель для нескольких значений.
26. Как макрос сделать доступным из всех документов?
27. Как экспортировать макрос в другой документ?