

Министерство образования и науки Российской Федерации
Вологодский государственный университет

Visual Basic for Application **ActiveX**

Лабораторный практикум

Вологда
2014

УДК 681 3.06

Visual Basic for Application. ActiveX: лабораторный практикум.
2-е изд. перераб. и доп. – Вологда: ВоГУ, 2014. – 67 с.

Лабораторный практикум предназначен для студентов экономического факультета, изучающих разделы курсов «Информационные технологии», «Персональный компьютер в работе...», «Основы программирования». В нём содержатся основные сведения языка программирования Visual Basic for Applications(VBA). Приводятся варианты самостоятельных заданий для лабораторных работ.

Утверждено редакционно-издательским советом ВоГУ

Составитель Р.И. Бибикова, канд. физ.-мат. наук, доцент

Рецензент Н.А. Сердюков, канд. техн. наук, доцент

Тема 1. Введение в VBA

Одним из наиболее популярных и востребованных пакетов в современном мире является Microsoft Office –незаменимый помощник при проведении самых разнообразных работ на компьютере. MS Office позволяет автоматизировать любую офисную работу в организациях различного направления деятельности. VBA (Visual Basic for Applications) считается стандартным языком написания сценариев для приложений Microsoft. В настоящее время он входит в состав всех компонентов MS Office.

VBA относится к категории объектно-ориентированных языков. При этом под объектом подразумевается программный компонент, который содержит данные и программный код. В VBA имеется более 100 встроенных объектов, например, рабочий лист, ячейка, диапазон, кнопки, и т.д.

Семейство представляет собой объект, содержащий несколько других объектов, обычно одного и того же типа. Например, объект Worksheets представляет собой семейство рабочих листов и содержит все открытые объекты Worksheet.

Объекты характеризуются некими свойствами. Каждый из объектов обладает определённым набором свойств. Свойство представляет собой атрибут объекта, определяющий его характеристики, такие как размер, цвет, положение на экране и состояние объекта, например доступность или видимость. Для изменения характеристик объекта надо изменить его свойства. Синтаксис установки значения свойства:

Объект. Свойство = ЗначениеСвойства

Например, `TextBox1.Enable = True`, что означает, что свойству `Enable` объекта управления «Текстовое поле» `TextBox1` присваивается логическое значение `True`, и пользователь может вносить изменения непосредственно в содержание этого текстового поля.

Другим важным понятием, связанным с программными объектами, являются методы, относящиеся к программному объекту. Метод – это действие, которое совершается над объектом. Синтаксис применения метода: Объект. Метод.

Например, `Application.Quit`, здесь при помощи метода `Quit` закрывается приложение.

Событие представляет собой действие, распознаваемое объектом, для которого можно запрограммировать отклик. События могут возникать в результате действий пользователя (щелчок на объект, двойной щелчок, наведение мыши, выделения и пр.), работы программы или могут быть вызваны системой.

Каждый программный объект характеризуется ключевым свойством – именем (`Name`). Различные объекты, входящие в конкретную разработку, должны иметь уникальные имена.

Суть программирования на VBA состоит в двух понятиях: событие и отклик на него. В случае, когда пользователь осуществляет какое-либо воздействие на систему, например, нажимает кнопку, то в качестве отклика выполняется код, созданный пользователем процедуры. Если не написана соответствующая процедура, то система никак не реагирует на данное событие. Таким образом, действия, происходящие в системе, являются событиями, а отклики на них – процедурами обработки событий. Программирование на VBA состоит в создании кода программ, которые генерируют отклики на события.

Мы рассмотрим разработку рабочих книг MS Excel с размещением на рабочих листах разнообразных объектов, характерных для интерфейса приложений Windows. Это кнопки, текстовые поля, переключатели, поля со списками, флажки, изображения и др. Кроме того, объектами будут рабочие книги, листы, ячейки. Как язык VBA впервые появился в MS Office 97 для всех продуктов (Excel, Word, Access, Power Point, ...). В настоящее время в MS Office 2007 это средство получило значительное развитие.

Знакомство с редактором VBA

Среда программирования VBA реализована в виде Редактора Basic (VBE, Visual Basic Editor), который может быть открыт из любого приложения пакета MS Office, в том числе из Excel.

Для того чтобы запустить редактор Visual Basic для Excel, необходимо вначале открыть приложение Excel. Затем реализовать одно из двух действий:

- Нажать кнопку Visual Basic на панели вкладки Разработчик для MS Excel-7 (для других версий пункт меню Сервис/Макрос/Редактор VBA).
- Нажать сочетание клавиш Alt + F11. После выполнения любого из этих действий на экране появится окно редактора VB.

Интерфейс VBA состоит из следующих основных компонентов: окно проекта, окно свойств, окно редактирования кода, окно форм, меню, панель инструментов. Элементы управления можно размещать на формах или на самих рабочих листах. Все элементы управления делятся на две группы: элементы управления формами и элементы управления ActiveX. Оба набора элементов управления имеют свои недостатки и свои преимущества. В общем случае элементы управления формами проще в применении, но зато элементы ActiveX являются более гибкими. В дальнейшем мы будем работать с элементами управления ActiveX.

Основы программирования VBA

В программах можно использовать стандартный набор операций над данными (как в Excel). Имеются три типа операций:

- Математические - выполняются над числами, их результатом является число;
- Отношения со знаками <, >, =, <=, >= применяются не только к числам, их значениями являются логические значения True – Истина и False – Ложь;
- Логические (And, Or, Not, ...), используются в логических выражениях;

В VBA используются следующие типы данных:

Базовые типы переменных

Имя типа	Возможные значения	Байты
Byte (бит)	0,...,255	1
Boolean (логическое)	True, False	2
Integer (короткое целое)		2
Long (длинное целое)		4
Single (число с плавающей запятой)		4
Double (число с двойной точностью)		8
Currency (денежный)	Длинные с денежным знаком	8
String (строка)	Любое кол-во символов	до 2 ³²
Date (дата и время)	От 1.01.1900 до 31.12.9999	8
Object (объект)	Ссылка на объект	4
Variant (универсальный тип)	Значения-любые данные	16

Переменные, которые используются в программах, должны быть объявлены (описаны). Для объявления переменных используются инструкции *Dim* (размер), или *Public* (общий), или *Private* (частный), или *Static* (статический). Если переменная не описана, то она считается неявно описанной типа *Variant*. На имя (идентификатор) переменной есть ограничения: это последовательность букв, цифр и символа подчеркивания, первой должна быть буква. Имя должно быть уникальным, длиной не более 255 символов.

Объявления простых переменных имеет следующий синтаксис:

Dim <имя переменной> [**As** <имя типа>]

Здесь содержимое в [] можно пропускать, As (как) – ключевое слово, которое сообщает, что определяется переменная.

Примеры описаний:

Dim a As Long, b as String, s As Double.

Dim C(5) As Integer –это одномерный массив с элементами C(0), C(1), C(2), C(3), C(4), C(5).

Любой элемент документа (слово, абзац, таблица,...) является объектом Excel. И каждому объекту соответствует собственный тип данных. Таким образом, переменным, объявленным с определённым типом, можно присваивать абзацы, списки, таблицы, стили и пр. с целью их дальнейшей обработки. Типы таких данных в Excel:

- *Application* - активное приложение MS Excel;
- *Workbook*- книга в Excel из семейства *Workbooks*;
- *Worksheet* – рабочий лист в Excel из семейства *Worksheets*;
- *Chart*- диаграмма из семейства *Charts*;
- *Name*- имя любого объекта в приложении, данной рабочей книге или рабочем листе из семейства *Names*;
- *Selection* – выделенный фрагмент в документе;
- *Range* – непрерывный диапазон ячеек на листе;
- *Cells(n, m)*- ячейка, находящаяся в *n*-й строке, *m* -м столбце листа, является альтернативой *Range*;
- *Style*- стиль форматирования из семейства *Styles*.

Для переменных таких типов объявление переменных происходит стандартным образом, но присваивание делается с помощью оператора *Set*. Например,

Dim Z As Range, V As Range

...

Set V = Range("A2")

В программах могут использоваться константы. Константы, в отличие от переменных, не могут менять своё значение.

Например,

Const R as Single = 0.7

Const Firma as String = "Дар Солнца"

Функции преобразования типов

Для преобразования строки в число и обратно существуют следующие функции:

Val(a) - возвращает число, содержащееся в строке *a* как числовое значение;

Str(n) – возвращает значение типа *String*, являющееся строковым представлением числа *n*.

В качестве разделителя функция *Str(n)* воспринимает только точку. Далее в таблице представлены функции преобразования из одного типа в другой. Все эти функции начинаются с буквы *C* (*Convert* преобразовать).

Таблица преобразования типов

Функция	Тип ответа
CBool(a)	Boolean
CByte(a)	Byte
CCur(a)	Currency
CDate(a)	Date
CDBl(a)	Double
CInt(a)	Integer
CStr(a)	String
CLng(a)	Long
CSng(a)	Single
CVar(a)	Variant

Операторы, выражения, операции

Строка с кодом в исходном тексте называется программным оператором. Программный оператор – это неделимое предложение, выполняющее какое-либо действие. Он может представлять собой любую комбинацию ключевых слов VB, свойств, функций, операций и символов, совокупность которых имеет корректную конструкцию, распознаваемую VB. Правила, применяемые при построении программных операторов, называются синтаксисом. Выражения - это комбинация знаков операций и операндов и, возможно, круглых скобок. Помимо описания, существуют три типа операторов: операторы присваивания, условные операторы и операторы цикла. Перед оператором может стоять метка – последовательность символов, начинающаяся с буквы и кончающаяся двоеточием.

Операторы присваивания – это основное средство изменения состояния. Они служат для изменения значения переменных и объектов. Оператор присваивания присваивает значение выражения переменной или свойству объекта и всегда включает знак равенства.

Синтаксис оператора присваивания:

Переменная (или свойство объекта) = Выражение

Примеры: $X = 5$ – переменная X получила значение 5. В результате работы двух операторов: $S = 0$ и $S = S + 1$ переменная S получила значение 1.

Некоторые управляющие операторы представлены в таблице:

Оператор	Действие
Оператор безусловного перехода GoTo	Синтаксис: GoTo метка Задаёт переход на строку с указанной меткой Метка – любой идентификатор
Оператор условного перехода If Then Else	Синтаксис: If Условие Then Инструкции1 [Else Инструкции2] End If Если условие принимает значение True, то выполняются Инструкции1, иначе выполняются Инструкции2. Допускается использование оператора If в виде If Условие Then Инструкции1 ElseIf Условие Then Инструкции2 Else Инструкции3 End If
Оператор выбора Select Case	Select Case выражение-тест Case выражение1 Инструкции1 ... Case выражение n Инструкции n Case Else Инструкции End Select
Оператор цикла со счётчиком For - Next	Синтаксис: For Счётчик = Начало To Конец [Step Шаг] Инструкции [Exit For] [Инструкции] Next[Счётчик] Оператор цикла повторяет выполнение группы инструкций, пока Счётчик изменяется от начального значения до конечного с указанным шагом. Если шаг не указан, то он полагается равным 1. Альтернативный способ выхода из цикла предоставляет инструкция Exit For.
Оператор цикла типа For Each - Next	Синтаксис: For Each Элемент In Группа [Инструкции] [Exit For] Next [Элемент] Повторяется выполнение группы инструкций для каждого элемента массива или семейства. Альтернативный способ выхода Exit For.
Оператор цикла типа While - Wend	Синтаксис: While Условие Инструкции Wend Инструкции повторяются до тех пор, пока Условие имеет значение True.

Код в VBA пишется либо в виде процедуры, либо в виде функции. Процедура имеет синтаксис:

```
Sub Имя([список параметров])
....
Операторы
End Sub
```

Функция имеет синтаксис:

```
Function Имя([список параметров]) [As тип переменной Имя]
....
Операторы
End Function
```

Квадратные скобки показывают, что это необязательные параметры. Могут быть процедуры и функции без параметров. (Но в любом случае круглые скобки после имени обязательны!).

Функция отличается от процедуры тем, что среди операторов в теле функции должен быть хотя бы один оператор, присваивающий имени функции Имя значение, т.е. результат работы функции должен сохраняться в переменной Имя.

Лабораторная работа №1

Пример 1. Имеются списки абитуриентов и их баллов на экзаменах. Составить пользовательскую функцию, отбирающую абитуриентов по проходному баллу.

Пусть список абитуриентов и их баллы записаны на Лист1.

	A	B	C
1	ФИО	Баллы	
2	Алексеев А.И.	70	
3	Стукалов В.Д.	89	
4	Иванов М.Ю.	77	
5	Степашкин К.Л.	62	
6	Петров П.П.	59	

В поле модуля VBA (Макрос/Редактор VBA/Insert/Модуль) запишем функцию

```

Function Зачисление(Балл, ПроходнойБалл) As String
If Балл >= ПроходнойБалл Then
Зачисление = "Зачислен"
Else
Зачисление = "      -      "
End If
End Function

```

Теперь обратимся к созданной функции из Лист1. Поставим курсор в ячейку C2 и вызовем пользовательскую функцию *Зачисление*. Для этого выполним пункт меню Вставка Функции и среди Пользовательских функций найдём функцию *Зачисление*. Вместо первого параметра щёлкнем на B2, а второй параметр (Проходной Балл) зададим равным 75. В итоге получим следующий результат:

C2		fx =Зачисление(B2;75)	
A	B	C	
ФИО	Баллы		
Алексеев А.И.	70	-	
Стукалов В.Д.	89	Зачислен	
Иванов М.Ю.	77	Зачислен	
Степашкин К.Л.	62	-	
Петров П.П.	59	-	

Пример 2. Для данных Примера 1 написать оценку, исходя из следующего:

80 – 100 –отлично,
 60 – 79 – хорошо,
 40 -59 -удовлетворительно,
 < 40 –неудовлетворительно.

В поле модуля напишем функцию:

```

Function Ball(a As Long) As String
Select Case a
Case 80 To 100
Ball = "отл."
Case 60 To 79
Ball = "хор."
Case 40 To 59
Ball = "уд."
Case Else
Ball = "неуд."
End Select
End Function

```

Здесь показана работа оператора выбора *Select Case*.

Вернёмся на Лист1. Из ячейки С2 обратимся к функции Ball:

(Вставка/Функции/Пользовательские функции/Ball)

В появившемся окне щёлкнем на В2 и ОК.

В ячейке С2 получим результат. Распространим формулу для всех данных. Имеем:

А	В	С	Д
фио	балл		
Алексеев А.И.	70	хор.	
Стукалов В.Д.	89	отл.	
Иванов М.Ю.	77	хор.	
Степашин К.Л.	62	хор.	
Петров П.П.	59	уд.	

Пример 3. Для данных примера 1 найти средний балл группы. Конечно, эту задачу просто решить средствами Excel, написав команду `=срзнач(В2:В6)`

Мы хотим показать, как в VBA использовать данные из рабочего листа Excel. В поле модуля запишем код:

```

General) СреднийБалл
Function СреднийБалл() As Single
    Dim n As Long, i As Long, s As Long
    s = 0
    'узнаем, сколько абитуриентов в списке
    n = 0
    While Worksheets("Лист1").Cells(n + 2, 1).Value <> 0
        n = n + 1
    Wend
    'в цикле просуммируем баллы
    For i = 2 To n + 1
        s = s + Worksheets("Лист1").Cells(i, 2).Value
    Next i
    СреднийБалл = s / n
End Function

```

Записав в ячейке В8 Лист1 команду `=СреднийБалл()` и выполнив её, получим ответ 71,4.

	А	В	С
1	ФИО	Баллы	
2	Алексеев А.И.	70	
3	Стукалов В.Д.	89	
4	Иванов М.Ю.	77	
5	Степашин К.Л.	62	
6	Петров П.П.	59	
7			
8	средний балл	71,4	

В программе для вычисления среднего балла мы использовали описания переменных

Dim n As Long, i As Long, s As Single. Также пользовались в операторе цикла типа *While –Wend*, описанием листа *Worksheets(“ИмяЛиста”)* и значением данного в ячейке *Cells(номер_строки, номер_столбца).Value*.

Самостоятельная работа 1

Задание 1. Имеется сложная система налогообложения. За основную работу сотрудник облагается налогом в 13%, за частную практику сотрудник облагается налогом в 20%, за опубликование статей – налогом в 35%, а за фотоработы – налогом в 30 %. Подсчитать сумму налогов сотрудников, если имеется документ следующего вида:

ФИО	Основной доход	Доход за частную практику	Доход за публикацию	Доход за фотоработы
Иванов	15000	8000		11000
Петров	5000		12000	
Сидоров	25000	4000	500	3000
Агапов	12500	7650		4000
Мишин	6000	200	11000	
Зверев	11000	8000		8000

Задание 2. Менеджеру по продажам необходимо разработать функцию, позволяющую рассчитать комиссионные выплаты. Процент комиссионных зависит от объема проданного товара и начисляется по следующему принципу:

Таблица 1

Правила расчёта комиссионных

Объём продаж за неделю, руб	Комиссионные, %
От 0 до 9999	3
От 10000 до 19999	3,5
От 20000 до 39999	4
Более 40000	5

Построить пользовательскую функцию для расчёта комиссионных в стандартном модуле и использовать её при расчёте комиссионных для списка сотрудников

Таблица 2

Список сотрудников

Сотрудник	Объём продаж	Ставка	Комиссионные
Иванов	20000	1	
Петров	13000	0	
Сидоров	34000	1	
Агапов	32000	1	
Мишин	12000	0	
Зверев	40000	1	

Задание 3. Усложнить задачу 2, считая, что комиссионные зависят от ставки, занимаемой менеджером; если он принят в постоянный штат фирмы, то комиссионные начисляются по закону, описанному в таблице.1. Если же он находится на испытательном сроке, то его комиссионные составляют 75% от номинала. В таблице 2 в столбце Ставка указано 1, если менеджер на постоянной ставке и 0, если менеджер на испытательном сроке.

Задание 4. Получить таблицу температур по Цельсию от 1 градуса до 20 градусов и их эквивалент по шкале Фаренгейта, используя для перевода формулу $t_f = 9/5 * t_c + 32$.

Задание 5. У кассы магазина выстроилась очередь из n покупателей. Время обслуживания продавцом i -го покупателя равно t_i ($i = 1, 2, \dots, n$). Пусть даны натуральное n и действительные $t_1 \dots t_n$. Получить $c_1, c_2, \dots, c_n$, где c_i – время пребывания i -го покупателя в очереди. Указать номер покупателя, для обслуживания которого продавцу потребовалось самое малое время.

Задание 6. Дана таблица, содержащая фамилии и даты рождения сотрудников. Записать в отдельном столбце возраст сотрудников в годах, с указанием наименования «год», «года», «лет». Последнее выполнить в виде пользовательской функции.

Тема 2. Встроенные диалоговые окна

В проекте VBA часто встречаются две разновидности диалоговых окон: окна сообщений и окна ввода:

Окно сообщений **MsgBox** выводит простейшие сообщения для пользователей. Окно ввода **InputBox** - обеспечивает ввод информации.

Функция **InputBox** выводит на экран диалоговое окно, содержащее сообщение, поле ввода и две кнопки: **OK** и **Cancel**, устанавливает режим ожидания ввода текста пользователем и нажатия кнопки, а затем возвращает значение типа **String** в переменную (по нажатию кнопки **OK**). При нажатии кнопки **Cancel** возвращает пустую строку (**Empty**).

Синтаксис: `InputBox (P[,T] [,D] [,X][,Y] [,H,C])`

Здесь P – строковое выражение, отображаемое как сообщение в диалоговом окне, может содержать несколько строк. Допускается символ возврата каретки (`chr(13)`), символ перевода строки (`chr(10)`) или комбинация этих символов.

T – строковое выражение, отображаемое в строке заголовка диалогового окна.

D – строковое выражение, отображаемое в поле ввода как используемое по умолчанию. Если этот параметр опущен, то поле ввода изображается пустым.

X – числовое выражение, задающее расстояние по горизонтали между левой границей диалогового окна и левым краем экрана. Если этот параметр опущен, то поле ввода выравнивается по центру экрана по горизонтали.

Y – числовое выражение, задающее расстояние по вертикали между верхней границей диалогового окна и верхним краем экрана. Если этот параметр опущен, то диалоговое окно помещается по вертикали примерно на одну треть высоты экрана.

H – Строковое значение, определяющее имя файла справки, содержащего сведения о данном диалоговом окне.

C – числовое выражение, определяющее номер раздела справочной системы.

Процедура **MsgBox** выводит на экран диалоговое окно, содержащее сообщения, устанавливает режим ожидания нажатия кнопки пользователем, а затем возвращает значение типа Integer, указывающее, какая кнопка была нажата. Синтаксис:

MsgBox (P [, B] [, T] [, H, C])

Здесь P – строковое выражение, отображаемое как сообщение в диалоговом окне.

B – числовое выражение, представляющее сумму значений, которые указывают число и тип отображаемых кнопок, тип используемого значка, основную кнопку. Значение по умолчанию этого параметра равняется 0.

T – строковое выражение, отображаемое в заголовке диалогового окна. Если этот параметр опущен, то в строку заголовка пишется имя приложения.

H и C аналогично тому, как в операторе InputBox.

Значения параметра B (*Button*) процедуры **MsgBox**, определяющие отображаемые кнопки в диалоговом окне смотрите в приложении.

Оператор **MsgBox** удобен для вывода информации. Если требуется узнать, какие данные ввёл пользователь, то надо какой-либо переменной присвоить значение, возвращаемое MsgBox, а её параметры заключить в скобки.

Лабораторная работа 2

Пример1. Компьютер спрашивает у пользователя, закончить ли счёт и отвечает или До свидания или Продолжим работу. Набираем в поле модуля код:

Листинг 1. Листинг процедуры pr

```
Sub pr()
Dim a As String
a = MsgBox("Закончить счёт? ?", vbYesNo, "Пример окна MsgBox")
If a = vbYes Then
MsgBox "До свидания"
Else
MsgBox "Продолжим работу"
End If
End Sub
```

Запускаем код на выполнение (Кнопка ► или пункт меню *RUN*).

Пример 2. Компьютер запрашивает информацию о пользователе и выдаёт некоторые сообщения.

Листинг 2. Листинг процедур *Znakomstvo* и *Prodoljim*.

```
Private Sub Znakomstvo()
Dim name As String
name = InputBox("Привет, как тебя зовут?", "VBA")
If name = Empty Then
MsgBox " Жаль, что ты не в настроении", vbExclamation, "VBA"
Exit Sub
End If
Prodoljim name
End Sub
```

```
Private Sub Prodoljim(name)
Dim r As Long, b As Long
b = vbQuestion + vbYesNo
r = MsgBox("Привет, " & name & ", может займёмся VBA?", b, "VBA")
Select Case r
Case vbYes
MsgBox "Что же, продолжим изучение этой главы", vbInformation, "VBA"
Case vbNo
MsgBox "Ну что же, продолжим в следующий раз. ", vbInformation, "VBA"
End Select
End Sub
```

В этой программе показано, как из одной процедуры происходит обращение к другой процедуре. Первая процедура называется **Znakomstvo**, а вторая процедура называется **Prodoljim**. Обращение из первой процедуры ко второй **Prodoljim name**, т.е. параметр *name* при обращении пишется без скобок.

В VBA предусмотрен объект **Debug**, который представляет отладчик и позволяет выводить данные в окно *Immediate*, вызываемое командой **View/Immediate/Window** (или сочетанием клавиш *Ctrl+G*). Чтобы вывести значения в окно *Immediate*, надо использовать метод **Print** объекта **Debug** с указанием в качестве параметров имена требуемых переменных (константы и переменные можно перечислять через запятую). Если требуется, чтобы несколько операторов **Debug** выводило информацию на одной строке, то в конце оператора ставят точку с запятой.

Примеры.

`Debug.Print a, b;`

`Debug.Print c, d`

Значения переменных *a, b, c, d* будут напечатаны в окне **Immediate** в одной строке.

Листинг с работой оператора *Debug*:

```
Sub ff()
Dim a(1 To 3) As Long
a(1) = 2
a(2) = 4
a(3) = 7
Debug.Print "a(1)=" & a(1)
Debug.Print "a(2)=" & a(2)
Debug.Print "a(3)=" & a(3)
End Sub
```



Здесь в операторе **Debug.Print** используется знак **&**, который объединяет текст "*a(1) =* " и значение *a(1)*.

Самостоятельная работа 2

Задание 1. Напечатать таблицу умножения чисел от 2 до 10.

Задание 2. Задан массив из n чисел. Упорядочить элементы массива в порядке возрастания.

Задание 3. В столбце А имеются фамилии студентов группы. Упорядочить фамилии по алфавиту.

Задание 4. Компьютер запрашивает, какую оценку вы получили по «Информатике» и выдаёт соответствующий текст. Используйте оператор Select Case.

Задание 5. Компьютер спрашивает Ваше имя и в каком году вы родились и выдаёт сколько вам лет и какой - либо комплемент.

Тема 3. Встроенные функции для обработки строк

В VBA имеются функции для обработки строковых выражений. Основные функции приведены ниже.

<i>Asc(a)</i>	Возвращает ASCII – код начальной буквы строки <i>a</i>
<i>Chr(k)</i>	Преобразует ASCII- код в строку
<i>LCase, UCase</i>	Преобразуют строку к нижнему и верхнему регистру
<i>Len(a)</i>	Возвращает количество символов в строке
<i>Left (a,n), Right (a,n)</i>	Возвращает подстроку, состоящую из заданного числа n первых, последних символов исходной строки
<i>Mid (a, i, k)</i>	Возвращает подстроку, начиная с i -го элемента длиной k
<i>LTrim, RTrim, Trim</i>	-возвращает копию строки без пробелов в начале, в конце, в начале и в конце соответственно
<i>Spase(n)</i>	Возвращает строку, состоящую из n пробелов
<i>Split</i>	Преобразует строку в одномерный массив, нумеруемый с нуля; Пример. <code>Dim M, S As String, i As Integer</code> <code>S = "Visual Basic for Application"</code> <code>M=Split(S)</code> <code>For i = LBound(M) To UBound (M)</code> <code>MsgBox M(i)</code> <code>Next i</code>
<i>Join</i>	Преобразует массив в строку
<i>String</i>	Возвращает строку из указанного числа одного символа
<i>StrComp</i>	Возвращает результат сравнения двух строк
<i>StrReverse</i>	Возвращает строку в обратном порядке
<i>InStr, InStrRev</i>	Возвращает позиции первого и последнего вхождения одной строки внутри второй строки (<i>InStr (n, S1, S2,0)</i>)
<i>Replace</i>	Находит и заменяет одну подстроку другой подстрокой.

Лабораторная работа 3

Пример 1. Зашифровать текст, используя метод подстановки. Выпишем все буквы алфавита, а под ними те же буквы, но перемешанные. Шифрование заключается в том, чтобы каждая буква алфавита из текста заменялась на перемещённую соответствующую букву.

Листинг 3.1. Процедура *Perevod* и *DeCode*.

```
' Шифрование методом подстановки.
Private n As Long
Const ABC As String = "йцукенгшщзхфывапрлджэячсмитбю 1234567890"
Const CBA As String = "1268бтимс 90чяэждлз345рпаювыф7йцеукнгшщзх"

Private Sub Perevod()
Dim A1 As String, A2 As String, Rab As String
A1 = "Офисное программирование. Офис 2007. "
' Шифровка
Debug.Print A1
DeCode A1, Rab, ABC, CBA
Debug.Print Rab
' Расшифровка
DeCode Rab, A2, CBA, ABC
Debug.Print A2
End Sub

' Процедура шифровки
Private Sub DeCode(StrIn, StrOut, OldKey, NewKey)
Dim i As Long, n As Long
Dim tmp As String
n = Len(StrIn)
StrOut = Space(n)
For i = 1 To n
tmp = Mid(StrIn, i, 1)
k = Instr(1, OldKey, tmp)
If k = 0 Then
Mid(StrOut, i, 1) = tmp
Else
Mid(StrOut, i, 1) = Mid(NewKey, k, 1)
End If
Next i
End Sub
```

Здесь используются константы ABC и CBA, в которые записывается, какие символы будут заменяться и на какие. Также используются некоторые функции для обработки строк.

Самостоятельная работа 3

Задание 1. Задан текст, состоящий из букв русского алфавита. Записать каждый символ текста буквами латинского алфавита.

Тема 4. Создание макросов с помощью макрорекордера

Для быстрого получения чернового варианта программного кода, используемого в создаваемом приложении, отлично подходит запись макрокоманд – макрорекордер. Это транслятор, создающий программу – макрос на языке VBA, которая является результатом перевода на язык VBA действий пользователя с момента запуска макроса и до окончания записи макроса.

Процесс создания макроса состоит из трёх этапов:

- Активация режима записи и присвоения макросу имени;
- Выполнение действий, которые надо записать;
- Остановка записи макроса.

Лабораторная работа 4

Пример 1. Создать макрос, выполняющий решение системы линейных алгебраических уравнений $Ax = b$. Здесь A – матрица коэффициентов, x – искомый, b – заданный векторы. Для решения запишем данные на Лист1 Excel.

	A	B	C	D	E	F
1	Матрица A					
2	2	4	3		3	
3	1	5	7		2	
4	8	6	9		7	
5						

Создадим макрос. Для этого выполним следующую последовательность действий:

Макрос/Сервис/ Начать запись

В открывшемся окне диалога Запись макроса, введём имя создаваемого макроса, например, *Решение*. Назначим комбинацию клавиш для запуска макроса. Напишем комментарий.

Щёлкаем на *ОК*. На Лист1 выделим диапазон G1:G3. В строке формул напишем формулу:

$= \text{Мумнож}(\text{Мобр}(A2:C4); E2:E4)$

Выполним команду нажатием трёх клавиш: *Ctrl + Shift + Enter*. Щёлкаем по кнопке Остановить запись. Макрос готов.

Выполняем команды

Сервис / Макрос / Изменить

Появится окно редактора VBA и в нём следующий код:

```
Sub Решение()
'
' Решение Макрос
' Макрос записан 09.09.12 Бибикова Р.И.
'
' Сочетание клавиш: Ctrl + p
'
Range ("G1:G3").Select
Selection.FormulaArray = "=MMULT(MINVERSE(RC[-6]:C[-4]),RC[-2]:R[2]C[-2])"
```

End Sub

Изменим программу следующим образом:

```
Sub Решение()
Dim a, b, x
a = InputBox("Введи адрес диапазона a")
b = InputBox("Введи адрес диапазона b")
x = InputBox("Введи адрес диапазона x")
Range (x).Select 'Выделяет область для ответа
Selection.FormulaArray = "=MMULT(MINVERSE(" & a & ")," & b & ")"
```

End Sub

Получилась универсальная процедура, решающая систему линейных алгебраических уравнений любого порядка, записанную в любом месте.

Самостоятельная работа 4

Задание 1. Предприятие выпускает три вида продукции из сырья трёх типов.

	A	B	C	D	E	F
1	Расход сырья по видам продукции					запас
2 Вид сырья	1	2	3			сырья
3 1	6	4	5			2400
4 2	4	3	1			1450
5 3	5	2	3			1550
6						
7						

Требуется определить возможный объём выпуска каждой продукции при заданных запасах сырья.

Задание 2.

При решении системы линейных алгебраических уравнений в случае, когда число уравнений и число неизвестных в уравнении $Ax = b$ совпадает, применяют предложенный выше метод для решения. Пусть количество неизвестных n , а количество уравнений m и $n \neq m$. Тогда при $m < n$, если система является совместной, то она имеет бесчисленное множество решений. В случае, когда $m > n$ и система совместна, удобно использовать метод наименьших квадратов. Для этого обе части уравнения

$$Ax = b$$

надо умножить на транспонированную матрицу A^t .

$$A^t Ax = A^t b$$

Затем обе части уравнения надо умножить на $(A^t A)^{-1}$:

$$(A^t A)^{-1} A^t Ax = (A^t A)^{-1} A^t b.$$

Т.к. $(A^t A)^{-1} A^t A = E$ (единичная матрица), то имеем:

$$Ex = (A^t A)^{-1} A^t b.$$

Отсюда

$$x = (A^t A)^{-1} A^t b.$$

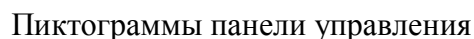
Написать процедуру для решения системы линейных алгебраических уравнений методом наименьших квадратов для системы вида

$$3x + 2y = 7$$

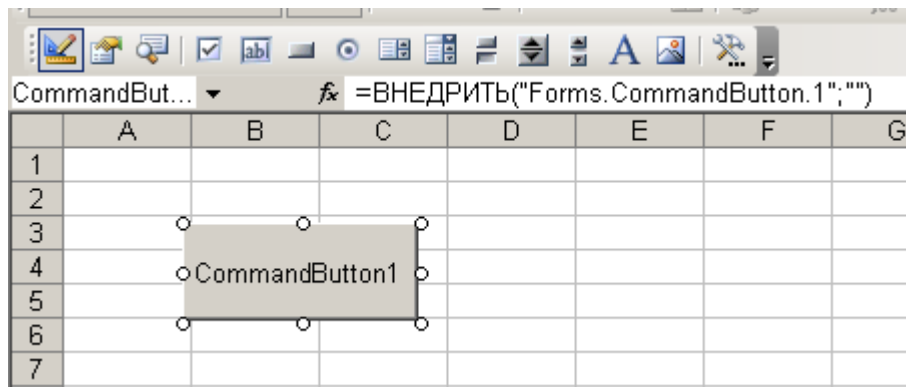
$$4x - 5y = 40$$

$$3x + 3y = 3.$$

Первая кнопка на полученной панели управления называется Режим конструктора. Вытаскивание элементов управления на лист Excel можно делать при включённом режиме конструктора. Другой режим называется режимом работы или режимом выполнения программы. Чтобы перейти в этот режим, надо отжать кнопку Режим конструктора. Чтобы записать программу в редакторе VBA, надо войти в режим конструктора и дважды щелкнуть на элемент управления, для которого пишется программа.



Перед тем, как начать какие-либо действия с какими-либо элементами ActiveX, его надо поместить на рабочий лист. Для этого надо сделать активным Режим конструктора. При наведении курсора на пиктограмму нужного элемента ActiveX, в качестве подсказки появится название элемента. Чтобы перенести этот элемент на рабочий лист, необходимо щёлкнуть на панели инструментов по его пиктограмме левой кнопкой мыши и, далее, переместить курсор мыши на лист. Курсор мыши примет вид, аналогичный математическому знаку $+$. Теперь можно нажать левую кнопку мыши и, не отпуская её, переместить курсор мыши в нужное место на листе, затем отпустить кнопку. В результате на рабочем листе мы увидим изображение появившегося элемента, окружённого маркерами.



Размещение командной кнопки на рабочем листе

После размещения элемента ActiveX на рабочем листе активизируется кнопка Режим конструктора. В этом режиме можно работать с элементами ActiveX для создания необходимого интерфейса на рабочем листе. Также в режиме конструктора пользователю предоставлена возможность создавать программные разработки для каждого элемента ActiveX, размещённого на рабочем листе. Дважды щёлкнув в режиме конструктора на созданный на рабочем листе объект (текстовое поле, кнопка, переключатель, список, поле со списком, ...) мы попадаем в среду VBA и имеем возможности писать соответствующий элементу код.

Другой режим – режим выполнения программы или режим работы. Чтобы перейти в этот режим, необходимо выйти из режима конструктора, т.е. отжать кнопку Режим конструктора. После этого можно пощёлкать созданными кнопками, выполняя соответствующие кнопкам программные коды.

Лабораторная работа 5

1. Свойства элемента управления Кнопка

У каждого программного объекта есть различные свойства, которые можно просмотреть (и, если нужно, изменить значения свойств).

Для просмотра свойств объекта, его необходимо сначала выделить в режиме конструктора (щёлкнуть на нём мышью), а затем с помощью пиктограммы Свойства (или щёлкнув по объекту правой кнопкой мыши и, вызвав контекстное меню, выбрать Свойства), открыть окно свойств.

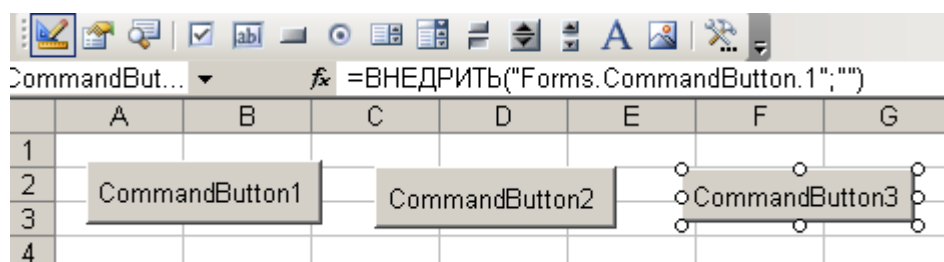
В окне свойств отображается таблица – набор строк в два столбца. Левый столбец отводится под названия свойств, правый – значения этих свойств. Содержание левого столбца нельзя изменить, а содержание правого столбца изменить можно.

Для ряда свойств возможны только два значения - *False* -ложь и *True* – истина.

Рассмотрим некоторые свойства объекта Кнопка, общие для многих других элементов конструктора.

<i>Visible</i>	Определяет видимость объекта на экране. Возможные значения <i>True</i> (видимый) и <i>False</i> (невидим, видим только в режиме Конструктора.)
<i>Enable</i>	Определяет доступ к объекту в режиме выполнения. Значение <i>False</i> – объект недоступен (доступен только в режиме Конструктора), значение <i>True</i> – объект доступен
<i>AutoSize</i>	Позволяет осуществлять автоматическую подгонку размеров кнопки под размер текста, расположенного на ней. Такое можно при значении <i>True</i> в качестве значения данного свойства
<i>Width</i>	Ширина
<i>Height</i>	Высота
<i>Top</i>	Координата верхней границы элемента, начиная от верхней части листа;
<i>Left</i>	- координата левой границы элемента, начиная от левой части листа;
<i>BackColor</i>	Определяет цвет заливки на кнопке
<i>ForeColor</i>	Определяет цвет текста на кнопке
<i>Caption</i>	Определяет надпись на кнопке
<i>Font</i>	Определяет шрифт текста на кнопке
<i>Picture</i>	Позволяет разместить на кнопке рисунок из графического файла. Если хотите убрать рисунок, то следует воспользоваться клавишей <i>Delete</i> .
<i>Name</i>	Имя, основное свойство любого объекта. Каждому объекту даётся имя по умолчанию. Это имя можно изменить и дать своё имя (идентификатор). Система формирования имени по умолчанию заключается в комбинации типа элемента управления и числа (порядкового номера).

Создадим на Лист1 три командные кнопки.



Их свойства имеют те же значения. Изменяя значения свойств *Caption* каждой кнопки и записав соответственно Ввод, Вывод, Печать, на листе 1 имеем:



2. Элементы управления Надпись и Текстовое поле

Элемент управления Надпись имеет пиктограмму *A* и используется для отображения на экране поясняющего текста. Ввод информации с клавиатуры в Надпись невозможен, данные в Надпись вводятся только из Свойства *Caption*. *Name* надписи по умолчанию *Label*.

Элемент управления Текстовое поле имеет пиктограмму *ab* и *Name* – *TextBox*. Этот элемент предназначен для ввода текста с клавиатуры.

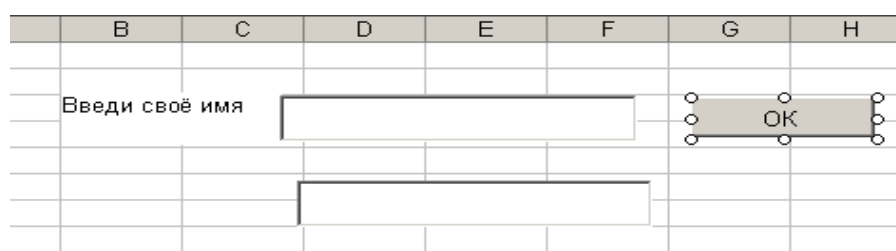
Для иллюстрации работы этих элементов приведём несложные примеры.

Пример 1. На листе Excel в режиме конструктора создать надпись, два текстовых поля и кнопку. Мы хотим, чтобы компьютер запросил имя и приветствовал нас.

Для решения задачи в режиме конструктора подберём следующие значения:

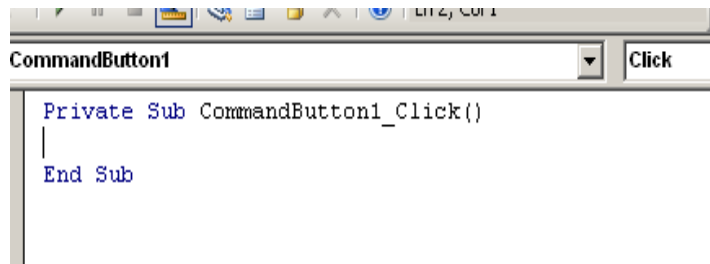
Объект	Name	Caption
Надпись	Label1	Введи своё имя
Текстовое поле	TextBox1	
Текстовое поле	TextBox2	
Кнопка	CommandButton1	ОК

Перетащим нужные элементы с панели ActiveX на лист Excel.



В режиме конструктора дважды щёлкнем по кнопке ОК и тем самым перейдём в среду написания кода (текста) программы. Перед нами автоматически открывается новое окно редактора VBA. В его правой части расположен раздел, предназначенный для написания программного кода, где мы и будем писать строки программы. Здесь в верхней части присутствуют два списка со стрелками ▼ (каждый открывается щелчком по стрелке). Левый список содержит перечень объектов, которые нам доступны при программировании, правый список включает события, связанные с объектом, который предварительно выбран в левом списке.

Таким образом, мы будем работать с событиями, используя технологию событийного программирования.



Наше событие – щелчок по кнопке *OK*, поэтому мы видим команды
Private Sub CommandButton1_Click()
End Sub

Здесь *CommandButton1_Click* – имя процедуры, которое автоматически формируется из двух составляющих: имя объекта командной кнопки *CommandButton1* и *Click* (щелчок по кнопке *OK*).

Подобные процедуры называют предопределёнными – они объективно существуют и выполняются всегда при совершении определённого действия. Продолжим разработку: напишем программный код, который в текстовое поле *TextBox2* выведет текст «Здравствуйте,» и выведет имя. Это команда

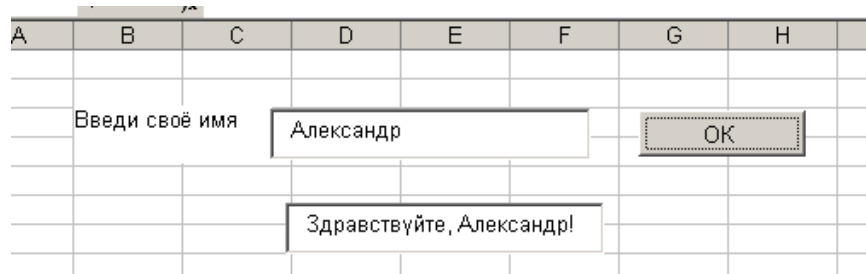
TextBox2.Text = “Здравствуйте, ” & TextBox1.Text & “!”

Листинг1. Щелчок по кнопке *OK*.

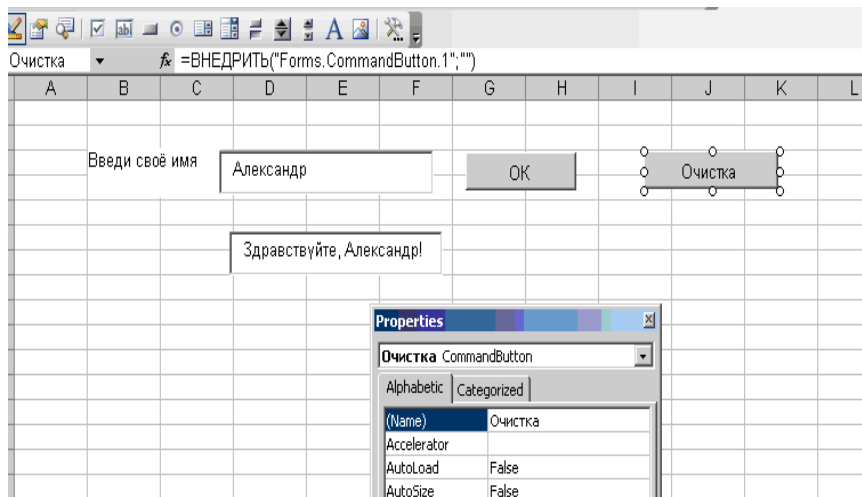
Private Sub CommandButton1_Click()
TextBox2.Text = “Здравствуйте, ” & TextBox1.Text & “!”
End Sub

Здесь символ *&* объединяет символьные строки в одну строку.

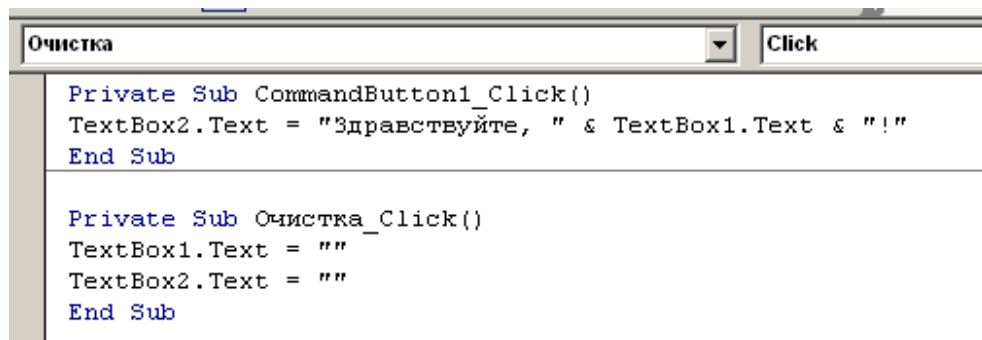
Вернёмся на Лист1. Уберём режим конструктора. В текстовом поле *TextBox1* наберём своё имя и щёлкнем кнопку *OK*. В текстовом поле *TextBox2* появится запись.



Можно добавить кнопку с *Name* –Очистка и с *Caption* – Очистка. Щелчок по этой кнопке должен очистить оба текстовых поля.



Здесь командной кнопке CommandButton2 дано имя Очистка, и значение Очистка. Ниже показаны обе процедуры в модуле редактора VBA.



Щёлкая по кнопке *OK*, получаем приветствие от компьютера, щёлкая по кнопке *Очистка*, очищаем оба текстовых поля

Примечание. Значение свойства *Text* определяет информацию, которая присутствует в текстовом поле.

Для автоматического переноса длинного текста со строки на строку нужно для свойства *MultiLine* установить значение *True*. Альтернативным является значение *False*, когда текст помещается в одной строке и может быть видна только одна его часть.

Самостоятельная работа 5

Задание 1. Пусть на рабочем листе Лист2 имеются данные фирмы по продаже недвижимости за определённый срок: фамилии менеджеров по продажам, их имена и отчества, какая недвижимость продана, на какую сумму и дата продажи. Также имеется список процента комиссионных от определённых сумм продаж. Создать надпись «Введите фамилию менеджера» и в текстовое поле ввести фамилию менеджера. Подсчитать его комиссионные.

Результат получить в виде:

Здравствуйте, (имя, отчество).
 Сегодня (дата) .
 Ваши комиссионные от (дата начала периода) до (дата
 конца периода) составляют . . .руб.

Тема 6. Элементы управления *Поле со списком* и *Список*

Поле со списком (ComboBox) используется для работы со списком (набором) значений и его особенность заключается в том, что в этом элементе управления до его активации всегда отображается только одна строка. После активации (щелчок по стрелке в поле со списком) список значений раскрывается и в нём с помощью мыши и автоматически появляющейся линейки прокрутки можно выбрать необходимый элемент.

У поля со списком кроме свойства *Name* (по умолчанию *ComboBox*) есть несколько характерных свойств:

.ListIndex - даёт номер выделенного элемента (нумерация начинается с нуля).

.ListCount - содержит число элементов, внесённых в список;

.Text - соответствует содержанию выделенного текста;

.List (номер) - осуществляет доступ к элементу с указанным номером.

Все свойства *Поля со списком* можно посмотреть при создании поля со списком в режиме конструктора.

Кроме свойств, у Поля со списком есть методы:

.Clear - удаляет все элементы из *Поля со списком* (очищает *Поле со списком*;

.AddItem - добавляет новый элемент в *Поле со списком*;

.RemoveItem - удаляет из *Поля со списком* указанный элемент. Для работы требуется удалить выделение (напр., *ComboBox1.ListIndex = -1*)

Лабораторная работа 6

Пример 1. На листе Excel создадим три поля со списком (*ComboBox1*, *ComboBox2*, *ComboBox3*), кнопку Щёлкни сюда и текстовое поле. В столбце I запишем числа от 1 до 31, в столбце J запишем месяцы от января до декабря, а в столбце K запишем годы, например, от 1973 до 2013. В окне свойств каждого *ComboBox* свойству *ListFillRange* присвоим соответственно значения I1:I31, J1:J12, K1:K40. Под кнопку написать программу, которая находит да-

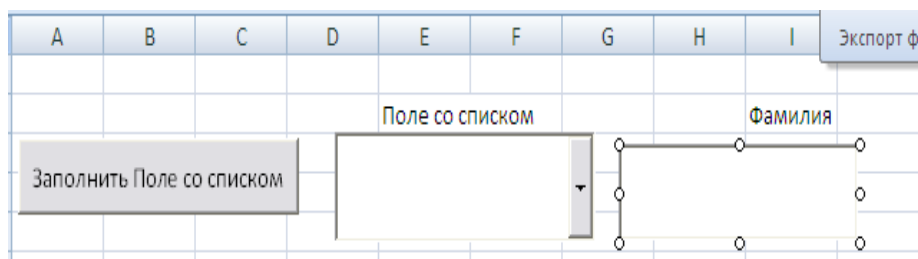
ту рождения, подсчитывает Ваш возраст и в зависимости от полученного результата в текстовое поле выдаёт какой-либо комплемент.

Листинг 1. Процедура щелчка по командной кнопке.

```
Private Sub CommandButton1_Click()
Private Sub CommandButton1_Click()
god = CLng(ComboBox3.Text)
mes = CLng(ComboBox2.ListIndex) + 1
den = CLng(ComboBox1.Text)
vozrast = 2013 - god
Select Case vozrast
Case 0 To 15
TextBox1.Value = "Вы родились " & den & "." & mes & "." & god _
& ". Вам " & 2013 - god & " лет. У вас всё впереди!"
Case 16 To 25
TextBox1.Value = "Вы родились " & den & "." & mes & "." & god _
& ". Вам " & 2013 - god & " лет. Вы молоды и прекрасны!"
Case Else
TextBox1.Value = "Вы родились " & den & "." & mes & "." & god _
& ". Вам " & 2013 - god & " лет. У вас прекрасный возраст!"
End Select
End Sub
```

Пример 2. На Лист1 Excel создадим *Поле со списком* (ComboBox1), текстовое поле (TextBox1) и командную кнопку (CommandButton1).

По щелчку на командную кнопку *Поле со списком* заполняется фамилиями сотрудников (например Иванов, Петров, Сидоров,), по щелчку на *Поле со списком* выбранная фамилия переносится в текстовое поле.



Расположение управляющих элементов на Лист1

Листинг 2. Процедура заполнения *Поля со списком* по щелчку на командной кнопке «Заполнить Поле со списком»

```
Private Sub CommandButton1_Click()
    ComboBox1.Clear 'Очистка Поля со списком
    TextBox1.Text = "" 'Очистка текстового поля
    ComboBox1.AddItem "Иванов"
    ComboBox1.AddItem "Петров"
    ComboBox1.AddItem "Сидоров"
    ComboBox1.AddItem "Сахаров"
End Sub
```

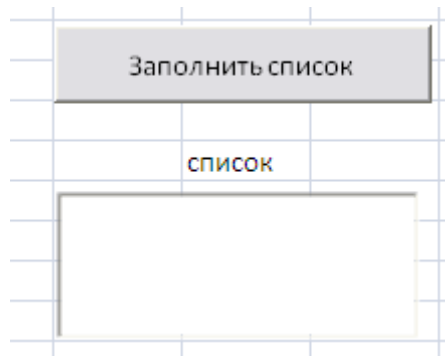
Листинг 1.2. Процедура щелчка на *Поле со списком* на символе ▼, выборе фамилии и записи её в текстовое поле

```
Private Sub ComboBox1_Click()
    Dim k As Long
    k = ComboBox1.ListIndex
    TextBox1.Value=ComboBox1.List(k)
End Sub
```

Здесь учитывается, что *ComboBox1.ListIndex* выдаёт номер выбранного в *Поле со списком* элемента, а свойство *List(k)* с этим номером осуществляет доступ к указанному элементу.

Кроме *Поля со списком* имеется другой элемент управления, который называется *Список(ListBox)*. У него своя пиктограмма, а свойства те же, что и у *Поля со списком*. Элемент управления *Список* занимает на рабочем листе фиксированный размер.

Пример 3. Пусть фамилии сотрудников записаны на Лист2. На Лист3 создадим командную Кнопку Заполнить *список* и элемент управления *Список*. *Список* должен заполняться данными с Лист2.



Листинг 2.1. Процедура заполнения Списка по щелчку на кнопке Заполнить список

```
Private Sub CommandButton1_Click()
Dim n As Long
ListBox1.Clear 'Очистка списка
'Подсчёт количества фамилий на Лист 2
n = 0
While Worksheets("Лист2").Cells(n + 2, 1).Value <> 0
n = n + 1
Wend
'Занесение фамилий в список
For i = 1 To n
ListBox1.AddItem Worksheets("Лист2").Cells(i + 1, 1).Value
Next i
End Sub
```

Здесь использован оператор цикла типа While-Wend, который считает, сколько фамилий находится в первом столбце Лист2 и записывает это значение в переменную n.

В операторе цикла типа For-Next данные с Лист2 переносятся в список ListBox1.

Самостоятельная работа 6

Задание 1. На рабочем листе Лист2 имеется прайс – лист, содержащий наименование и цену товаров компьютерного магазина. На листе Лист1 создать кнопку с названием Создать списки, создать Поля со списками по каждому товару и из выбранных товаров организовать список Заказа. Создать кнопку Печать и по щелчку по ней напечатать список заказа и сумму заказа.

Создать списки

Печать

Компьютеры

Ноутбуки

Сумки

Заказ

Задание 2. К заданию1 добавить пункт: при переносе заказа из Поля со списком в список Заказа, убрать название товара из соответствующего Поля со списком.

Задание 3. На листе *Прайс-лист* имеются данные, содержащие наименование товаров и их цены, на листе *Кнопки* создать надпись в виде «Введи наименование товара» и текстовое поле, куда надо ввести запрос в виде молочное, колбаса, конфеты, зефир и т.д. По запросу составить список, в который переносятся товары из *Прайс-листа* с указанием веса, цены и т.д. По щелчку на выбранный товар из списка его данные переносятся в список заказа. Под списком заказа найти итоговую стоимость выбранных товаров.

	А	В	С	Д	Е	Ф
1	Введи наименование			Список		Заказ
2	товара					
3						
4						
5						
6						
7						
8					сумма =	
9						
10						

Задание 4. Создать автоматизированную книгу для учёта материальных ценностей. Рабочий лист Номенклатура содержит справочник товаров.

Рабочий лист Поступление корректирует поступление товаров. Лист Номенклатура скрытый и служит лишь для главного менеджера. Вводить в него данные можно только при знании пароля. Пароль задаётся в виде текстового поля. На рабочем листе Поступление выведены следующие элементы управления: 4 текстовых поля и кнопка. По щелчку на кнопку Ввести новый

товар данные нового товара (название, цена, количество) вводятся на рабочий лист Новый товар. По щелчку на кнопку Ввод в Номенклатуру, данные из листа Новый товар переписываются на лист Номенклатура.

Новый товар

название

цена

количество

дата

Ввод в Номенклатуру

пароль

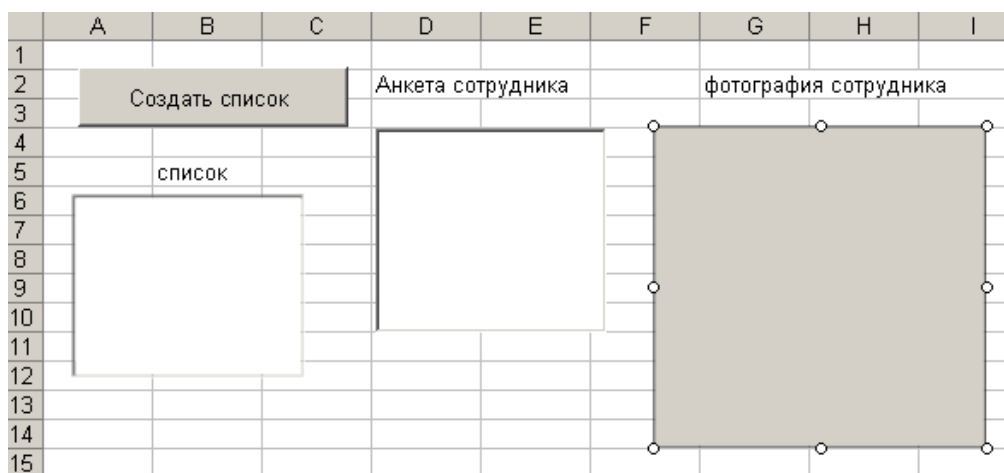
Ввести новый товар

Тема 7. Работа с изображениями

Кроме обычных свойств у Изображения есть свойство Picture. При выборе свойства Picture в окне свойств изображения, автоматически открывается диалоговое окно для выбора файла. В этом случае надо указать один из файлов с графическим изображением с расширением .bmp, .jpg, .jpeg, .gif и т.д..

Лабораторная работа 7

Создадим электронную таблицу для сотрудников фирмы. Пусть на Анкеты записаны анкетные данные на каждого сотрудника, а на диске C: в папке с именем fotki имеются фотографии каждого сотрудника (имя файла с фото совпадает с фамилией сотрудника, а расширение файла - .JPG). На листе Электронная анкета в режиме Конструктора выведем кнопку Создать список, список, текстовое поле и изображение



Листинг 1.1. Процедура заполнения списка по щелчку на кнопке Создать список.

```
Private Sub CommandButton1_Click()
    ' Очистки Списка и текстового поля
    ListBox1.Clear
    TextBox1.Value = ""
    ListBox1.AddItem "Ivanov"
    ListBox1.AddItem "Petrova"
    ListBox1.AddItem "Sidorov"
    ListBox1.AddItem "Klimov"
    ListBox1.AddItem "Andreeva"
    ListBox1.AddItem "Olimov"
    ListBox1.ListIndex = -1
End Sub
```

Листинг 1.1. Процедура выбора фотографии и анкеты по щелчку на элемент из списка

```
Private Sub ListBox1_Click()
    Dim k As Long
    k = ListBox1.ListIndex
    Select Case k
    Case 0
        Image1.Picture = LoadPicture("C:\fotki\Ivanov.JPG")
        TextBox1.Value = Worksheets("Лучм2").Range("A1").Value
    Case 1
        Image1.Picture = LoadPicture("C:\fotki\Petrova.JPG")
        TextBox1.Value = Worksheets("Лучм2").Range("A2").Value
    Case 2
        Image1.Picture = LoadPicture("C:\fotki\Sidorov.JPG")
        TextBox1.Value = Worksheets("Лучм2").Range("A3").Value
    Case 3
```

```
Image1.Picture = LoadPicture("C:\fotki\Klimov.JPG")
TextBox1.Value = Worksheets("Лист2").Range("A4").Value
Case 4
```

```
Image1.Picture = LoadPicture("C:\fotki\Andreeva.JPG")
TextBox1.Value = Worksheets("Лист2").Range("A5").Value
Case 5
```

```
Image1.Picture = LoadPicture("C:\fotki\Olimov.JPG")
TextBox1.Value = Worksheets("Лист2").Range("A6").Value
End Select
End Sub
```

Примечание. Если фотографии были записаны на диске C: в папке *fotki* под тем же именем (как в нашем случае), что и в списке, то можно было написать

```
Image1.Picture = LoadPicture("C:\fotki\" + ListBox1.Text + ".JPG")
```

Самостоятельная работа 7

Задание 1. На Листе 1 находится прайс – лист со списком компьютеров и ценами на них, а также указанием адреса файла с фотографией компьютера. На Листе 2 создать кнопку Заполнить список, текстовое поле Заказ и текстовое поле Цена. В эти поля переносятся из прайс - листа данные выбранного компьютера. В Изображение перенести фото выбранного компьютера, используя адрес, указанный в прайс – листе.

	А	В	С	Д	Е	Ф	Г
1				Заказ		Цена	
2	Заполнить список						
3							
4							
5				Фото компьютера			
6							
7							
8							
9							
10							
11							
12							
13							
14							

Тема 8. Работа с переключателями и флажком

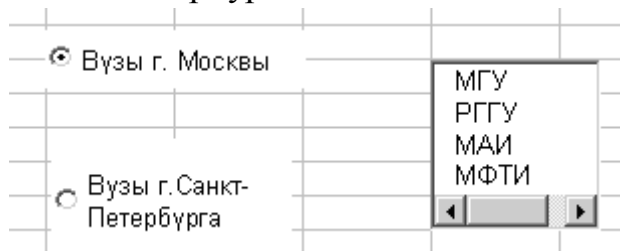
Переключатель используется для того, чтобы установить определённую опцию либо отображать конкретный набор элементов управления. По умолчанию переключатель имеет имя `OptionButton`. Если он включён, то его значение – `True`, иначе его значение – `False`.

Элемент управления Флажок имеет имя `CheckBox`. Флажок предоставляет пользователю возможность выбора. Флажок имеет два состояния: установлен (включён) и сброшен (выключен), имеет те же основные свойства, что и переключатель (`Value`, `Caption`). Основным событием элемента управления Флажок является событие `Change`.

Работу переключателя и флажка рассмотрим на примерах.

Лабораторная работа 8

Пример 1. Пусть имеются два переключателя и список. По щелчку на первом переключателе список должен заполниться именами вузов г. Москвы, при щелчке на втором переключателе список должен заполниться именами вузов г. Санкт Петербурга.



Листинг 1.1. Щелчок по первому переключателю.

```
Private Sub OptionButton1_Click()
    ListBox1.Clear
    ListBox1.AddItem "МГУ"
    ListBox1.AddItem "РГГУ"
    ListBox1.AddItem "МАИ"
    ListBox1.AddItem "МФТИ"
End Sub
```

Листинг 1.2. Щелчок по второму переключателю.

```
Private Sub OptionButton2_Click()
    ListBox1.Clear
    ListBox1.AddItem "СПбГУ"
    ListBox1.AddItem "РГПУ"
    ListBox1.AddItem "ЛЭТИ"
End Sub
```

Пример 2. Вычислить цену заказанного стеклопакета, если её квадратный метр стоит 1500 руб., цена квадратного метра москитной сетки стоит 300 руб., фиксатор стоит 200 руб., а микропроветриватель стоит 600 рублей. На листе Excel создать два текстовых поля для ширины и высоты стеклопакета, три флажка для москитной сетки, фиксатора и микропроветривателя, командную строку для расчёта и текстовое поле для результата расчёта.

The screenshot shows an Excel spreadsheet with the following elements:

- Input field for width: "ширина (см.)" with value "200".
- Input field for height: "высота (см.)" with value "300".
- Checkbox "Москитная сетка" (checked).
- Checkbox "Фиксатор" (unchecked).
- Checkbox "Микропроветриватель" (unchecked).
- Button "Расчёт" (Calculation).
- Result field: "Цена стеклопакета" with value "Цена стеклопакета = 10800 руб".

```
Private Sub CommandButton1_Click()
    t = CDbl(TextBox1.Text) * CDbl(TextBox2.Text)
    s = t * 1500 / 10000
    If CheckBox1.Value = True Then
        s = s + t * 300 / 10000
    End If
    If CheckBox2.Value = True Then
        s = s + 200
    End If
    If CheckBox3.Value Then
        s = s + 600
    End If
    If CheckBox1.Value = True And CheckBox2.Value = True And CheckBox3.Value = True Then
        s = s + s * 0.1
    End If
    TextBox3.Value = "Цена стеклопакета = " & CStr(s) & " руб."
End Sub
```

Самостоятельная работа 8

Задание 1. Составить электронный справочник по предприятиям Вологодской и Ярославской области. Создать два переключателя с текстами Вологодская обл. и Ярославская обл. При щелчке на одном переключателе создаётся список городов этой области, при щелчке на другом переключателе создаётся список городов другой области. Для каждого выбранного города создать список предприятий этого города.

☐ Вологодская область ☐ Ярославская область

Список городов

Список предприятий города

Примечание. При выборе Вологодской области в текстовое поле Список городов записывается текст «Вологодская область», а при выборе Ярославской области в текстовое поле Список городов записывается текст «Ярославская область».

Аналогично, при выборе определённого города из списка в текстовое поле Список предприятий города записывается текст с названием города, например «Рыбинск»

Задание 2. На листе Кадры имеются анкетные данные всех сотрудников фирмы. На листе Поиск создать 3 переключателя под названиями Фамилии, Должность, Пол и кнопку Заполнить список, щелчок на которую заполняет список из листа Кадры с данными о сотрудниках выбранной переключателем категории. Для выбранного сотрудника вызвать фото и анкетные данные.

☐ ФИО	☐ Должность	☐ Пол
Заполнить список		
Фото	Анкетные данные	

Задание 3. Создать электронную анкету для приёма на работу.

Фамилия		Место жительства	
Имя		☐ г.Вологда	
Отчество		☐ Другой город	
Дата рождения			
☐ Студент		☐ Специалист	
Место учёбы		Место работы	
Курс		Примечание	
☐ Хорошее знание английского языка ☐ Водительские права ☐ Знание информационных технологий			

После заполнения анкеты перенести данные на Лист2.

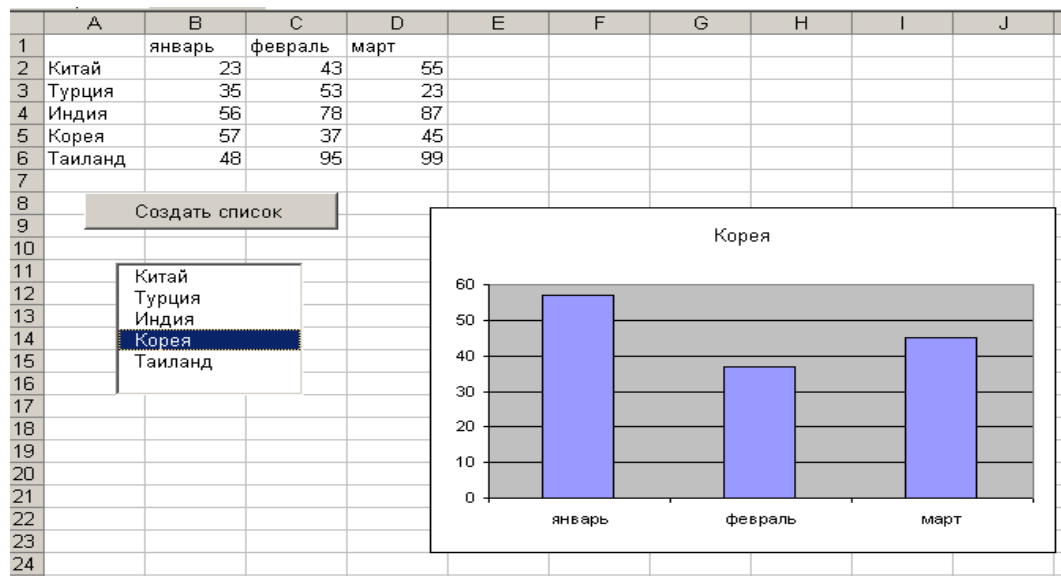
Тема 9. Работа с диаграммами

В Excel имеется богатейший набор гистограмм, диаграмм, функций. Чтобы использовать диаграммы в VBA, удобнее составить их в Excel в виде макроса, а затем изменить по потребностям программу в VBA.

Лабораторная работа 9

Задание 1. Предположим, что вы менеджер в некоторой фирме. Вам надо составить отчёт о результатах покупок в разных странах за период с января по март этого года. Кроме того, объём покупок по каждой стране должен быть представлен в виде диаграммы. Всего имеется пять стран, а создавать по отдельной диаграмме для каждой страны было бы неудобно. Более подходящим выглядит вариант существования переменной диаграммы. А именно, из списка выбирается страна, и на диаграмме отражается динамика закупок по выбранной стране. Закупки указываются в условных единицах.

На листе Лист1 имеются данные по закупкам. Мы хотим, чтобы на этом листе был список с названиями стран и диаграмма для выбранной из списка страны.



Листинг 9.1 Заполнение списка.

```
Private Sub CommandButton1_Click()
    ListBox1.Clear
    k = 1
    While Worksheets("Лист1").Cells(k + 1, 1).Value <> 0
        k = k + 1
    Wend
    For i = 2 To k + 1
        ListBox1.AddItem Worksheets("Лист1").Cells(i, 1).Value
    Next i
End Sub
```

Листинг 9.2. Процедура создания диаграммы по выбору из списка страны

```
Private Sub ListBox1_Click()
    'удаляем ранее созданную диаграмму
    ChartObjects.Delete
    'узнаём номер выделенной строки из списка
    i = ListBox1.ListIndex
    'семейство диаграмм
    Charts.Add
    'задаёт тип диаграммы
    ActiveChart.ChartType = xlColumnClustered
    'задаём диапазон данных для выбранной страны
    ActiveChart.SetSourceData Source:= _
        Sheets("Лист1").Range(Cells(i + 2, 2), _
            Cells(i + 2, 4)), PlotBy:=xlRows
    ' задаём диапазон для оси абцисс
    ActiveChart.SeriesCollection(1).XValues = _
        Sheets("Лист1").Range(Cells(1, 2), Cells(1, 4)).Value
    'задаём заголовок
    ActiveChart.SeriesCollection(1).Name = _
        Sheets("Лист1").Cells(i + 2, 1).Value
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Лист1"
    ' легенда отсутствует
    ActiveChart.HasLegend = False
End Sub
```

Самостоятельная работа 9

Задание 1. На Лист1 имеются данные продаж по сети магазинов за несколько месяцев. На этом листе создать два переключателя, один список, который задаёт или список магазинов, или список месяцев. По выбору из списка построить диаграмму по продажам в выбранном магазине или по продажам магазинов за определённый месяц.

	А	В	С	Д	Е
1	Магазин	январь	февраль	март	апрель
2	Форум	23000	25000	34000	43000
3	Мармелад	14000	23000	32000	29000
4	РИО	34000	56000	66000	75000
5	Невский	68000	82000	69000	71000

☐ Создать список
магазинов

☐ Создать список
месяцев

Список

Задание 2. К заданию 1 добавить флажок. При включенном состоянии флажка выбирается гистограмма, при выключенном состоянии выбирается круговая диаграмма.

Тема 10. Работа со счётчиком. Финансовые задачи

Элемент управления Счётчик применяется для установки числового значения, причём только целого неотрицательного значения. Основным событием элемента управления Счётчик является событие Change, генерируемое при изменении значения счётчика.

Основными свойствами Счётчика являются:

- Name, которое по умолчанию имеет имя SpinButton.
- Value, которое возвращает или устанавливает текущее значение счётчика;
- Min и Max, которые возвращают или устанавливают минимальное и максимальное значения счётчика.

В VBA имеется много встроенных функций для проведения финансовых расчётов. Они существенно облегчают разработку финансовых приложений. Ниже на примерах покажем, как можно автоматизировать работу с финансовыми функциями.

В Приложении 2 можно посмотреть список финансовых функций VBA.

Лабораторная работа 10

Решим следующую задачу: Нас просят дать в долг 200000 руб. и обещают возвращать по 20000 руб в течение 6 лет. Выгодна ли будет эта сделка при годовой ставке 6%?

Задачу решим несколькими способами.

1. Будем запрашивать годовую ставку процента в банке, размер ссуды, срок и сумму возврата. Код набирается в основном модуле (Редактор VBA/Insert/модуль) и имеет вид:

```
Private Sub Dolg()
Dim P As Single, S As Long, C As Single, V As Single, vivod As String
Dim P1 As String
P1 = InputBox("Введи годовую ставку процента, напр, в виде 6%")
P = Val(P1) / 100
S = InputBox("Введи размер ссуды")
C = InputBox("Введи срок")
V = InputBox("Введи сумму возврата")
B = C * V
D = S * (1 + P) ^ C
If D > B Then
vivod = " Выгодно класть в банк"
Else
vivod = " Выгодно дать в долг"
End If
MsgBox "При проценте в банке P= " & P & vivod
End Sub
```

2. Поместим данные на Лист 1 и против процентной ставки установим Счётчик. Мы хотим, чтобы значение годовой ставки можно было бы вводить как в ячейку листа с клавиатуры, так и Счётчиком. Причём, оба способа должны быть синхронизированы таким образом, чтобы при вводе числа в ячейку с клавиатуры автоматически изменялось и значение Счётчика.

	А	В
1	Размер ссуды	100000
2	Срок (год)	6
3	Ежегодно возвращаемые деньги	20000
4	Годовая процентная ставка	6%
5	Чистый текущий объём вклада	
6	Вывод	

Установим Счётчик, и в ячейку B5 занесём формулу

B5 fx =БС(В4;В2;;В1)		
	A	B
1	Размер ссуды	100000
2	Срок (год)	6
3	Ежегодно возвращаемые деньги	20000
4	Годовая процентная ставка	1%
5	Чистый текущий объём вклада	-106152
6	Вывод	Выгодно дать в долг

а в ячейку B6 занесём формулу

B6 fx =ЕСЛИ(-B5<B3*B2;"Выгодно дать в долг";"Выгодно положить в банк")					
	A	B	C	D	E
1	Размер ссуды	100000			
2	Срок (год)	6			
3	Ежегодно возвращаемые деньги	20000			
4	Годовая процентная ставка	1%			
5	Чистый текущий объём вклада	-106152			
6	Вывод	Выгодно дать в долг			

Теперь, изменяя значение Счётчика или значение в ячейке B4, получим соответствующий вывод.

- Улучшим проект, вводя переключатели, которые будут устанавливать, когда в банке начисляются проценты, в начале месяца или в конце месяца.

	A	B	C
1	Размер ссуды	100000	
2	Срок (год)	6	
3	Ежегодно возвращаемые деньги	20000	
4	Годовая процентная ставка	6%	
5	Чистый текущий объём вклада	-141852	
6	Вывод	Выгодно класть в банк	
7			
8		Когда начисляются проценты?	
9		☐ в начале периода	
10		☒ в конце периода	
11			
12			

Листинг процедур, устанавливающих Счётчик.

```
Private Sub SpinButton1_Change()
Worksheets(«Лист1»).Range(«B4»).Value = 0.01 * SpinButton1.Value
End Sub

Private Sub Worksheet_Change(ByVal Target As Range)
If Target.Address = «$B$4» Then
SpinButton1.Value = CInt(100 * _ Worksheets(«Лист1»).Range(«B4»).Value)
End If
End Sub
```

Листинг для переключателей.

```
Private Sub OptionButton1_Click()
Worksheets("Лист1").Range("B5").FormulaLocal = "=BC(B4;B2;;B1)"
End Sub

Private Sub OptionButton2_Click()
Worksheets("Лист3").Range("B5").FormulaLocal = "=BC(B4;B2;;B1;1)"
End Sub
```

Пример 2. Стоимость инвестиционных проектов.

Рассмотрим финансовую функцию ПС(), в которой в качестве значения возвращается стоимость инвестиций, приведённая к текущему моменту времени. Этот показатель определяет сумму, которую нужно было бы вложить при данном значении процентной ставки, чтобы получить эквивалентный доход. У функции пять аргументов, которые указываются в такой последовательности:

1. Процентная ставка за период между выплатами.
2. Количество периодов выплаты.
3. Постоянная сумма выплат за каждый период.
4. Будущая стоимость инвестиций (или остаток суммы задолженности к окончанию выплат).
5. Аргумент, который принимает значения 1, если выплаты производятся в начале периода, и 0, если в конце периода.

Рассмотрим простой пример, в котором требуется оценить стоимость двух инвестиционных проектов. Каждый из проектов рассчитан на 5 лет. В первом подразумевается получение ежегодных дивидендов в размере 20000 рублей. Второй подразумевает получение суммы 100000 рублей по окончании проекта (т.е. через 5 лет). Процентная ставка равна 4%.

Формула первого проекта:

B7	▼	f_x =ПС(B3;B4;B5)		
	A	B	C	
1	Стоимость инвестиционных проектов			
2				
3	Процентная ставка	4%		
4	Период	5		
5	Ежегодные поступления	20000		
6	Конечная стоимость	100000		
7	Первый проект	-89 036,45р.		
8	Второй проект	-82 192,71р.		
9				

Формула второго проекта:

B8	▼	f_x	=ПС(B3;B4;;B6)	
	A	B	C	D
1	Стоимость инвестиционных проектов			
2				
3	Процентная ставка	4%		
4	Период	5		
5	Ежегодные поступления	20000		
6	Конечная стоимость	100000		
7	Первый проект	-89 036,45р.		
8	Второй проект	-82 192,71р.		
9				

Как видим из результатов, первый проект оказался предпочтительней.

Простую автоматизацию можно сделать, поставив, например, счётчик на период и написав необходимый код.

	A	B	C
1	Стоимость инвестиционных проектов		
2			
3	Процентная ставка	4%	
4	Период	4	
5	Ежегодные поступления	20000	
6	Конечная стоимость	100000	
7	Первый проект	-72 597,90р.	
8	Второй проект	-85 480,42р.	
9			

Листинг кода Счётчик:

```
Private Sub SpinButton1_Change()
Range("B4").Value = CStr(SpinButton1.Value)
End Sub
```

Счётчик может изменять содержимое ячейки B4 в сторону уменьшения или увеличения. Как видим, если период поставить равным 4 года, то второй проект предпочтительней первого.

Пример 3. Выплаты по займам.

Финансовая функция Excel ПЛТ() полезна в тех случаях, когда требуется вычислять периодические платежи по обслуживанию всевозможных займов. Функция имеет 5 аргументов, которые указываются в такой последовательности:

- Процентная ставка для выплат по займу.
- Количество периодов выплаты.
- Приведённая стоимость займа.
- Остаток средств после последней выплаты.
- Аргумент, который принимает значения 1, если выплаты производятся в начале периода, и 0, если в конце периода.

Финансовая функция ПРПЛТ() (в VBA – IPMT) служит для расчёта суммы, выплачиваемой по процентам. Финансовая функция ОСПЛТ() (в VBA – PPMT) даёт выплату для погашения номинальной суммы займа. Аргументы у этих функций такие же, как и у функции ПЛТ(), за тем исключением, что между первым и вторым аргументом вклинивается ещё один, определяющий порядковый номер периода выплат, для которого вычисляются соответствующие составляющие.

Сделаем полную автоматизацию выплаты по займам.

Создадим 9 Надписей (Ежемесячные выплаты по закладной, Процентная ставка, и т.д.), 8 текстовых полей, 1 Счётчик и 1 командную кнопку.

	A	B	C	D	E	F
	Ежемесячные выплаты по закладной					
	Процентная ставка	0.07	Период(месяц)	4		
	Период(лет)	30	Выплата процентов		-581,8905	
	Стоимость (руб)	100000	Выплата долга		-83,412	
	Ежемесячные выплаты	-665,3025				
			Итого		-665,3025	
		OK				

Листинг Щелчок по командной кнопке ОК.

```
Private Sub CommandButton1_Click()
Dim a As Single, b As Single, c As Single, d As Single
a = CSng(Val(TextBox1.Text)) / 12
b = CSng(Val(TextBox2.Text)) * 12
c = CSng(Val(TextBox3.Text))
d = Pmt(a, b, c)
  TextBox4.Value = CStr(d)
End Sub
```

Листинг Щелчок по кнопке Счётчик

```
Private Sub SpinButton1_Change()
Dim a As Single, b As Single
Dim c As Single, d As Single, r As Single
SpinButton1.Min = 1
a = CSng(Val(TextBox1.Text)) / 12
b = CSng(Val(TextBox2.Text)) * 12
SpinButton1.Max = b
k = SpinButton1.Value
c = CSng(Val(TextBox3.Text))
d = IPmt(a, k, b, c)
  TextBox7.Value = CStr(k)
  TextBox5.Value = CStr(d)
  r = PPmt(a, k, b, c)
  TextBox6.Value = CStr(r)
  TextBox8.Value = CStr(d + r)
End Sub
```

Самостоятельная работа 10

Задание 1. Кредит в 1000000 р. выдан на 10 лет под 13% годовых с ежеквартальными платежами. Каков размер ежеквартального платежа в случае, когда платежи вносятся в начале квартала, в конце квартала? Какая часть платежа за конец второго года идёт на погашение процентов по кредиту?

Задание 2. Вас просят дать в долг A руб. и обещают вернуть в течение N лет. При какой годовой процентной ставке эта сделка имеет смысл?

Задание 3. Вы берёте в долг P руб под готовую процентную ставку по A руб. Сколько лет займут эти выплаты?

Задание 4. Вы собираетесь вкладывать по A руб. в течение n лет при годовой ставке $i\%$. Сколько денег будет через n лет?

Задание 5. Вас просят дать в долг P руб. и обещают вернуть P_1 руб. через год, P_2 руб. через два года и т.д., P_n через n лет. При какой годовой ставке в банке это имеет смысл?

Тема 11. Организация кадрового учёта

Автоматизируем участок работы, связанной с кадровым учётом организации. Пусть на листе Штатное расписание каждая строка информирует о конкретной должности в определённом подразделении.

Структурное подразделение	Должность	кол-во шт. ед-ц	Оклад	Всего в месяц	Занятость ставок
Админ-ый отдел	Ген.директор	1	50000	50000	1
Админ-ый отдел	Секретарь	2	20000	40000	1
Админ-ый отдел	Курьер	2	8000	16000	1
Админ-ый отдел	Переводчик	1	15000	15000	1
Админ-ый отдел	Юрист	1	24000	24000	1
Админ-ый отдел	Завхоз	1	20000	20000	1
Технический отдел	Тех.директор	1	30000	30000	1
Технический отдел	Инженер	5	25000	125000	4
Технический отдел	Энергетик	3	20000	60000	2
Технический отдел	менеджер по ар.	2	20000	40000	2
Технический отдел	менеджер по пр.	2	23000	46000	2
Технический отдел	менеджер по разв	1	25000	25000	1
Коммерческий отдел	директор	1	40000	40000	1
Коммерческий отдел	маркетолог	5	25000	125000	4
Коммерческий отдел	директор по мар.	1	40000	40000	1
Коммерческий отдел	директор по пр.	1	40000	40000	1
Финансовый отдел	главбух	1	45000	45000	1
Финансовый отдел	зам.бух.	1	35000	35000	1
Финансовый отдел	бухгалтер	1	30000	30000	1
Финансовый отдел	фин. контролёр	2	32000	64000	1
		итого		910000	

Лист Штатное расписание

Лист Справочный содержит в каждом столбце перечень данных, нужных в различных списках (для исключения неоднозначности).

Пол	Вид работы	Договор	Структурное подразделение	Должность
муж	Основная	Трудовой	Технический отдел	Ген.директор
жен	По совместительству	Договор услуг	Админ-ый отдел	Секретарь
			Коммерческий от.	Курьер
			Финансовый отдел	Переводчик
				Юрист
				Завхоз
				Тех.директор
				Инженер
				Энергетик
				менеджер по ар.
				менеджер по пр.
				менеджер по разв
				директор
				маркетолог
				директор по мар.
				директор по пр.
				главбух
				зам.бух.
				бухгалтер
				фин. контролёр

Лист Справочный

Лист Основной содержит базу данных сотрудников.

Лист Управление содержит кнопки Принять и Уволить.

Приём нового сотрудника
Увольнение

Принять

Уволить

ЛистУправление

Кнопка — Принять- это зачисление нового сотрудника, а Уволить – это увольнение сотрудника.

Листинг щелчка по кнопке Принять

```
Private Sub CommandButton1_Click()
```

WorkSheets(“Принятие”).Activate

Здесь появляется рабочий лист Принятие и заполняются все ComboBox1, ComboBox2, Данные берутся с листа Справочный.

Для создания листа Принятие заранее расположим на нём все кнопки и заголовков. Как любой графический объект, элементы управления можно рисовать при нажатой клавише Shift, что обеспечит ему пропорциональное изменение размера, либо при зажатой клавише Alt, для привязки элемента управления к сетке рабочего листа.

Принятого работника вписать в лист Основной.

Лист Принятие содержит 8 текстовых полей, 5 полей со списками и кнопку. Как любой графический объект, элементы управления можно рисовать при зажатой клавише Shift, что обеспечит ему пропорциональное изменение размера, либо при зажатой клавише Alt, для привязки элемента управления к сетке рабочего листа.

Принятие нового сотрудника

Табельный номер	Дата приёма	Подразделение	
Фамилия	Вид работы	Должность	
Имя	Вид договора	Пол	
Отчество	Информация о приказе		
	Дата	Номер	Оклад
Записать			

Лист Принятие

Аналогичная работа делается с уволенным сотрудником. Создаётся лист Увольнение, которое должно отразить, кого увольняют, по какому приказу, напечатать обходной лист, сделать изменения в других листах.

Тема 12. Работа со строками, цветом, ячейками.

Создание бегущей строки

Свойство Font объекта Range или объекта Cells представляет собой шрифт. Объект Font имеет следующие свойства:

Name – возвращает описание строки, указывающее имя шрифта,

FontStyle- задаёт стиль шрифта:

Regular –обычный,

Bold – полужирный,

Italic-курсив,

BoldItalic-полужирный курсив.

Size- задаёт размер шрифта

Color задаёт цвет шрифта

Пример.

Worksheets(“Лист1”).Range(“A1:D4”).Font.Size=16

Worksheets(“Лист1”).Range(“A1:D4”).FontStyle=Bold

Коды цветов в VBA задаются числами в форме 16-ричной системы, функциями и QBColor или константами цвета.

Функция RGB (Red, Green, Blue) позволяет получить любой цвет, смешивая красную, зелёную и синюю краски различной интенсивности. Red, Green, Blue целые числа от 0 до 15.

Значения параметров для получения стандартных цветов					Константы, задающие цвет
Цвет	Red	Green	Blue		
Чёрный	0	0	0		vbBlack
Синий	0	0	255		vbBlue
Зелёный	0	255	0		vbGreen
Голубой	0	255	255		vbCyan
Красный	255	0	0		vbRed
Розовый	255	0	255		vbMagenta
Жёлтый	255	255	0		vbYellow
Белый	255	255	255		vbWhite

Лабораторная работа 12

Пример 1. Создание бегущей строки

```
Dim i1 As Integer
'kol-vo probelov i nachale stroki
Sub Start()
  ' Установка начального количества пробелов
  i1 = 10
  MovingString
End Sub
'Первый вызов бегущей строки
Sub MovingString()
If i1 >= 0 Then
  ' Отображение строки
  Range("A1").Value = Space(i1) & "Privet!!"
  i1 = i1 - 1
  ' Указание, что данную строку нужно вызвать через 1 сек.
  Application.OnTime Now + TimeValue("00:00:01"), "MovingString"
End If
End Sub
```

После запуска этого кода, записанного в модуле Лист1, в ячейке A1 будет отображаться бегущая строка с текстом Privet!!.

Пример 2. Создание бегущей картинки.

Заставим двигаться надпись «Привет!!», повернутой на 30 градусов и написанной полужирным шрифтом.

```
'Бегущая картинка
Sub MovingImage()
Dim I As Integer
Dim image As Object
' Создание изображения (в ячейке A1)
With Range("A1")
  ' Формирование значения в ячейке:
  ' Текст
  .Value = "privet!"
  ' полужирный шрифт
  .Font.Bold = True
  ' цвет
  .Font.Color = RGB(233, 133, 229)
  ' размер шрифта
  .Font.Size = 16
  ' угол наклона
  .Orientation = 30

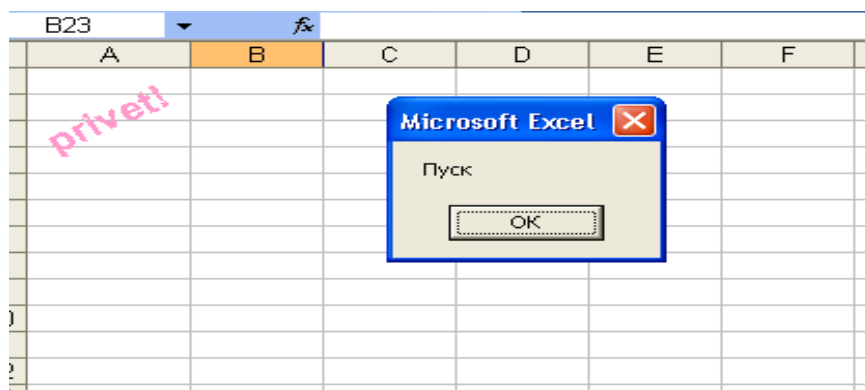
```

```

' Отображение текста полностью
.EntireColumn.AutoFit
' Копирование в буфер обмена
.Copy
' Создание самостоятельного изображения

Set image = ActiveSheet.Pictures.Paste(Link:=False)
' Содержимое ячейки больше не нужно
.Clear
End With
' Начальное положение – левый верхний угол
With image
.Top = 0
.Left = 0
End With
MsgBox " Пуск"
With image
' Перемещение изображения по диагонали
For I = 0 To 100
.Top = I
.Left = I
Next
' Удаление изображения
.Delete
End With
' Удаление ссылки на изображение
Set image = Nothing
End Sub
Запуск бегущей картинки

```



Пример 3. Мигающая ячейка. Нам необходима ячейка, которая 10 раз каждые 5 секунд будет изменять цвет фона с красного на зелёный. Запишем следующий код:

```

Sub BlinkingCell()
Static intCalls As Integer ' Счётчик количества миганий

```

```

' Если ячейка мигала менее 10 раз, то изменим её цвет
If intCalls < 10 Then
intCalls = intCalls + 1
' определение, какой цвет надо установить
If Range("A1").Interior.Color <> RGB(255, 0, 0) Then
' если цвет не красный, назначим красный
Range("A1").Interior.Color = RGB(255, 0,0)
Else
'Назначим зелёный цвет
Range("A1").Interior.Color = RGB(0,255, 0)
End If
' Эту процедуру необходимо вызвать через 5 секунд
Application.OnTime Now + TimeValue("00:00:05"), "BlinkingCell"
Else
'Хватит мигать
Range("A1").Interior.ColorIndex = xINone
intCalls = 0
End If
End Sub

```

После запуска кода будет мигать ячейка A1. Путём внесения соответствующих изменений в код, можно регулировать продолжительность цикла и изменять цвета фона по своему усмотрению.

Пример 4. Создание калькулятора. Создадим простой калькулятор, предназначенный для вычисления значений арифметических выражений.

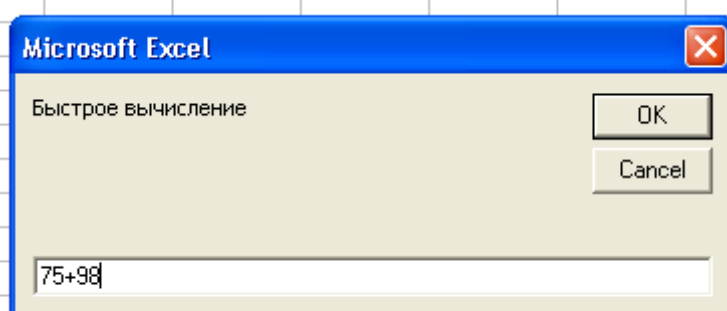
```

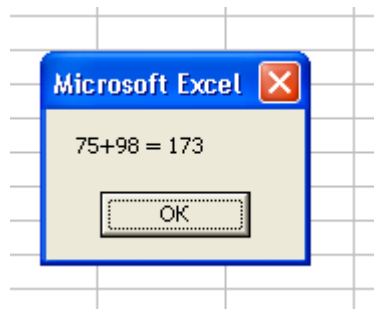
Sub MoyKalkulator()
Dim s1 As String
s1 = InputBox(«Быстрое вычисление»)
MsgBox s1 & " = " & Application.Evaluate(s1)

End Sub

```

После выполнения макроса появится окно:





Пример 5. Вызов таблицы цветов. При необходимости можно вывести на экран таблицу цветов и соответствующих значений свойства ColorIndex (данное свойство задаёт индексированный цвет заливки в соответствии с палитрой цветов).

Sub ShowColorTable()

Dim intColor As Integer

' Формирование заголовка таблицы

Range("A1").Value = "цвет"

Range("B1").Value = "Значение свойства ColorIndex"

' Вывод таблицы

Range("A2").Select

For intColor = 1 To 56

' Окрашиваем ячейку столбца A в текущий цвет

With ActiveCell.Interior

.ColorIndex = intColor

.Pattern = xlSolid

.PatternColorIndex = xlAutomatic

End With

' В ячейку B вносим индекс текущего цвета

ActiveCell.Offset(0, 1).Value = intColor

' Переходим на следующую строку

ActiveCell.Offset(1, 0).Activate

Next

' Покажем ячейку A1 (начало таблицы)

Range("A1").Select

ActiveWindow.ScrollRow = 1

End Sub

Пример 6. Вращающиеся автофигуры. Составим код, который вращает с одновременным перемещением сразу две автофигуры.

Sub RotatingAutoShapes()

Static fRunning As Boolean

' Проверка, выполнялся ли этот макрос

If fRunning Then

' При повторном запуске останавливаем все запущенные макросы

fRunning = False

```

End
End If
' Укажем, что макрос запущен
fRunning = True

Dim Sell As Range
Dim intLeftBorder As Long
Dim intRightBorder As Long
Dim intTopBorder As Long
Dim intBottomBorder As Long
Dim alngVertSpeed(1 To 2) As Long
Dim alngHorzSpeed(1 To 2) As Long

Dim ashShapes(1 To 2) As Shape
Dim i As Integer
' Заполнение массива автофигур
Set ashShapes(1) = ActiveSheet.Shapes(1)
Set ashShapes(2) = ActiveSheet.Shapes(2)
' Заполнение массива скоростей для первой фигуры
alngVertSpeed(1) = 3
alngHorzSpeed(1) = 3
' Заполнение массива скоростей для второй фигуры
alngVertSpeed(2) = 4
alngHorzSpeed(2) = 4
' Получение границ рабочей ячейки
Set cell = Range("B2")
intLeftBorder = cell.Left
intRightBorder = cell.Left + cell.Width
intTopBorder = cell.Top
intBottomBorder = cell.Top + cell.Height
' Выполнение вращения и перемещения фигур
Do
' Изменение положения каждой фигуры
For i = 1 To 2
With ashShapes(i)
' Контроль достижения правой фигуры
If .Left + .Width + alngHorzSpeed(i) > intRightBorder Then
' Корректировка положения
.Left = intRightBorder - .Width
' Изменение направления
alngHorzSpeed(i) = -alngHorzSpeed(i)
End If
' Контроль достижения левой границы
If .Left + alngHorzSpeed(i) < intLeftBorder Then
' Изменение направления

```

```

.Left = intLeftBorder
' Корректировка положения
alngHorzSpeed(i) = -alngHorzSpeed(i)
End If
' Контроль достижения нижней границы
If .Top + .Height + alngVertSpeed(i) > intBottomBorder Then
' Корректировка положения
.Top = intBottomBorder - .Height
' Изменение направления
alngVertSpeed(i) = -alngVertSpeed(i)
End If
' Контроль достижения верхней границы
If .Top + alngVertSpeed(i) < intTopBorder Then
.Top = intTopBorder
' Изменение направления
alngVertSpeed(i) = -alngVertSpeed(i)
End If
.Left = .Left + alngHorzSpeed(i)
.Top = .Top + alngVertSpeed(i)
' Перемещение автофигуры
.IncrementRotation alngVertSpeed(i)
' Команда обработать пользовательский код

DoEvents
End With
Next
Loop
End Sub

```

Теперь в ячейке B2 нужно создать любые две автофигуры (перед этим ячейку B2 следует существенно увеличить, сделав её размером примерно с четверть экрана, чтобы было место для вращения автофигур).

Самостоятельная работа. Создать свой логотип.

Тема 13. Применение технологии при решении макроэкономических задач

Известно, что рациональное функционирование многоотраслевого хозяйства предполагает соблюдение баланса между отраслями. Каждая отрасль многоотраслевого хозяйства, с одной стороны, является производителем определённой продукции, а с другой, –потребителем продукции, выпускаемой другими отраслями. Макроэкономика функционирования многоотраслевого хозяйства требует, чтобы соблюдался баланс по производству и потреблению между отдельными отраслями.

Балансовый принцип связи различных отраслей состоит в том, что валовой выпуск i -й продукции должен быть равен сумме объёма потребления. В простейшей форме балансовые соотношения имеют вид:

$$x_i = x_{i1} + x_{i2} + \dots + x_{in} + y_i, \quad i = 1, 2, \dots, n,$$

где x_i - общий объём выпускаемой продукции i -й отрасли;

x_{ij} - объём продукции i -й отрасли, потребляемой j -й областью при производстве объёма продукции x_j ;

y_i - объём продукции i -й отрасли конечного потребления (для реализации в непроизводственной сфере).

Для производства продукции j -й отрасли объёмом x_j нужно использовать продукцию i -й области объёмом $a_{ij}x_j$, где a_{ij} - постоянное число, характеризующее прямые затраты. Это допущение позволяет представить модель многоотраслевой экономики (модель Леонтьева) в виде системы линейных уравнений, которая в матричной форме имеет вид:

$$\bar{x} = A\bar{x} + \bar{y}$$

где $\bar{x}$ - вектор валового выпуска;

$\bar{y}$ - вектор объёма продукции конечного потребления;

A - матрица коэффициентов прямых затрат.

Приведённая система уравнений может быть представлена в виде

$$(E - A)\bar{x} = \bar{y},$$

где E - единичная матрица.

Если существует обратная матрица $(E - A)^{-1}$ (матрица полных затрат), то существует единственное решение системы

$$\bar{x} = (E - A)^{-1} \bar{y}.$$

Из экономической теории известно несколько критериев продуктивности матрицы A :

- матрица A продуктивна тогда и только тогда, когда матрица $(E - A)^{-1}$ существует и её элементы неотрицательны;
- матрица A с неотрицательными элементами продуктивна, если сумма элементов по любому её столбцу (строке) не больше единицы, причём хотя бы для одного столбца (строки) строго меньше единицы.

Замечание. Неотрицательная матрица $A \geq 0$ называется продуктивной, если существует хотя бы один вектор $X > 0$, такой, что $X > AX$ или в преобразованном виде $(E - A)X > 0$.

Экономический смысл этого определения состоит в следующем: неотрицательная матрица $A \geq 0$ продуктивна, если существует такой положительный вектор-столбец объёмов производства отраслей $X > 0$, что каждая отрасль может произвести некоторое количество $Y > 0$ конечной продукции.

Рассмотрим пример решения макроэкономической задачи на применение модели Леонтьева.

	A	B	C	D	E	F	G	H	I
1	№			Потребление				Конечный	Валовой
2	п/п	Отрасль	1	2	3	4	5	продукт	выпуск, ден.ед.
3	1	Станкостроение	15	12	24	23	16	10	100
4	2	Энергетика	10	3	35	15	7	30	100
5	3	Машиностроение	10	5	10	10	10	5	50
6	4	Автомобильная промышленность	10	5	10	5	5	15	50
7	5	Добыча и переработка углеводородов	7	15	15	3	3	50	100
8									

Требуется найти матрицу коэффициентов прямых затрат и определить её продуктивность, объёмы конечного продукта.

В приведённой таблице в первых пяти столбцах (группа Потребление) содержатся значения x_{ij} , а в последнем столбце – элементы вектора объёма конечного потребления $\bar{y}$.

Решение в Excel. Рассмотрим матрицу X, расположенную в диапазоне C3:G7, вектор $\bar{x}$, расположенный в диапазоне H3:H7, вектор $\bar{y}$, расположенный в диапазоне I3:I7. Составим макрос для решения поставленной задачи на применение модели Леонтьева.

Матрица коэффициентов прямых затрат A вычисляется путём деления i-го столбца матрицы Потребления X на строку транспонированного вектора $\bar{y}$. Это вычисление можно выполнить, используя формулу деления матрицы Потребление на транспонированный вектор валового выпуска $\bar{y}$.

. На листе выделим область C11:G15 и в строке формул введём команду:

=C3:G7/ТРАНСП(I3:I7)

Эта команда выполняется нажатием трёх клавиш: <Ctrl> + <Shift> + <Enter> (матричная операция). Получим значения матрицы A.

C11	{=C3:G7/ТРАНСП(I3:I7)}												
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	№			Потребление				Конечный	Валовой				
2	п/п	Отрасль	1	2	3	4	5	продукт	выпуск				
3	1	Станкостроение	15	12	24	23	16	10	100		96		
4	2	Энергетика	10	3	35	15	7	30	100		97		
5	3	Машиностроение	10	5	10	10	10	5	50		48		
6	4	Автомобильная промышленность	10	5	10	5	5	15	50		48		
7	5	Добыча и переработка углеводородов	7	15	15	3	3	50	100		91		
8													
9													
10			Матрица коэффициентов прямых затрат										
11			0,15	0,12	0,48	0,46	0,16		1				
12			0,1	0,03	0,7	0,3	0,07			1			
13			0,1	0,05	0,2	0,2	0,1				1		
14			0,1	0,05	0,2	0,1	0,05					1	
15			0,07	0,15	0,3	0,06	0,03						1

Просуммируем значения элементов каждого столбца полученной матрицы коэффициентов прямых затрат.

C16		=СУММ(C11:C15)												
A	B	C	D	E	F	G	H	I	J	K	L	M		
№			Потребление				Конечный	Валовой						
п/п	Отрасль	1	2	3	4	5	продукт	выпуск						
1	Станкостроение	15	12	24	23	16	10	100			96			
2	Энергетика	10	3	35	15	7	30	100			97			
3	Машиностроение	10	5	10	10	10	5	50			48			
4	Автомобильная промышленность	10	5	10	5	5	15	50			48			
5	Добыча и переработка углеводородов	7	15	15	3	3	50	100			91			
		Матрица коэффициентов прямых затрат												
1		0,15	0,12	0,48	0,46	0,16			1					
2		0,1	0,03	0,7	0,3	0,07				1				
3		0,1	0,05	0,2	0,2	0,1					1			
4		0,1	0,05	0,2	0,1	0,05						1		
5		0,07	0,15	0,3	0,06	0,03						1		
6	сумма	0,52	0,4	1,88	1,12	0,41								

Вычислим значения элементов полных затрат:

C19		=МОБР(I11:M15-C11:G15))									
	A	B	C	D	E	F	G	H	I	J	K
1	№			Потребление				Конечный	Валовой		
2	п/п	Отрасль	1	2	3	4	5	продукт	выпуск		
3	1	Станкостроение	15	12	24	23	16	10	100		96
4	2	Энергетика	10	3	35	15	7	30	100		97
5	3	Машиностроение	10	5	10	10	10	5	50		48
6	4	Автомобильная промышленность	10	5	10	5	5	15	50		48
7	5	Добыча и переработка углеводородов	7	15	15	3	3	50	100		91
8											
9											
10			Матрица коэффициентов прямых затрат								
11			0,15	0,12	0,48	0,46	0,16		1		
12			0,1	0,03	0,7	0,3	0,07			1	
13			0,1	0,05	0,2	0,2	0,1				
14			0,1	0,05	0,2	0,1	0,05				
15			0,07	0,15	0,3	0,06	0,03				
16		сумма	0,52	0,4	1,88	1,12	0,41				
17											
18			Матрица полных затрат								
19			1,709	0,4916	2,0684	1,538	0,61				
20			0,571	1,3371	1,9867	1,21	0,458				
21			0,374	0,2358	1,9839	0,732	0,321				
22			0,324	0,1995	0,8434	1,546	0,234				
23			0,347	0,3275	1,1222	0,62	1,259				

Проанализируем полученные в результате расчётов данные. Матрица полных затрат $(E-A)^{-1}$ существует, все её элементы положительны. Следовательно, необходимое и достаточное условие продуктивности матрицы A выполняется.

Приложения

Приложение 1

1. Значения параметра *B (Button)* процедуры *MsgBox*, определяющие отображаемые кнопки в диалоговом окне.

Константа		Отображаются кнопки
vbOKOnly	0	ОК
vbOKCancel	1	ОК, Отмена
vbAbortRetryIgnore	2	Стоп, Повтор, Пропустить
vbYesNoCancel	3	Да, Нет, Отмена
vbYesNo	4	Да, Нет
vbRetryCancel	5	Повтор, Отмена

2. Значения параметра *B (Button)* процедуры *MsgBox*, определяющие отображаемые информационные значки в диалоговом окне.

Константа	Значение	Значок сообщения
vbCritical	16	⊗
vbQuestion	32	?
vbExclamation	48	!
vbInformation	64	ⓘ

3. Значения параметра *B (Button)* процедуры *MsgBox*, определяющие основную кнопку диалоговом окне

Константа	Значение	Номер основной кнопки
vbDefaultButton1	0	1
vbDefaultButton2	256	2
vbDefaultButton3	512	3
vbDefaultButton4	768	4

4. Значения, возвращаемые функцией *MsgBox*

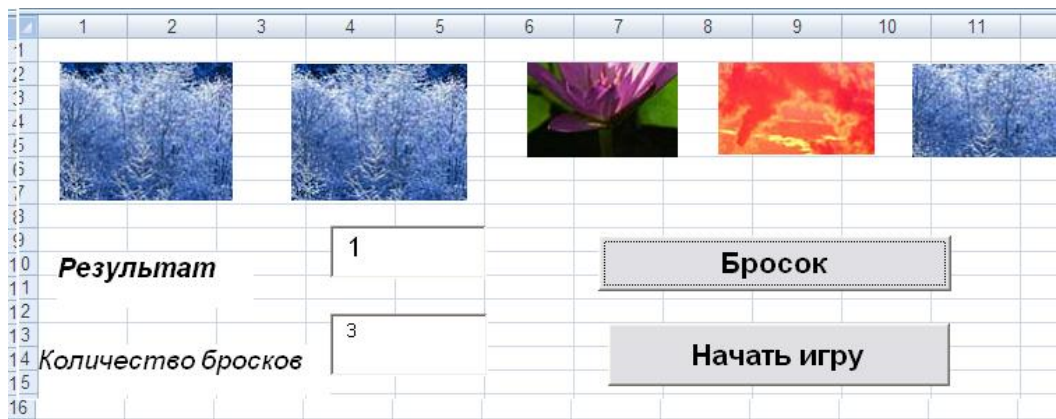
Константа	Значение	Кнопка
vbOK	1	ОК
vbCancel	2	Отмена (Cancel)
vbAbort	3	Прервать (Abort)
vbRetry	4	Повторить (Retry)
vbIgnore	5	Пропустить (Ignore)
vbYes	6	Да (Yes)
vbNo	7	Нет (No)

Встроенные в VBA функции финансовых расчётов

Функции в VBA	Функции в Excel	Возвращаемое значение
DDB	ДДОБ	Возвращает величину амортизации имущества для указанного периода, используя метод двукратного учёта амортизации
FV	БС	Возвращает будущее значение вклада на основе постоянных платежей, осуществляемых через определённые промежутки времени, и постоянной процентной ставки. Функция FV подходит для расчёта итогов накоплений при ежемесячных банковских взносах.
IPmt	ПРПЛТ	Возвращает платежи по процентам за данный период на основе постоянных выплат, осуществляемых через определённые промежутки времени, и постоянной процентной ставки
IRR	ВСД	Возвращает внутреннюю скорость оборота для ряда последовательных операций с наличными, представленными числовыми значениями. Объёмы операций не должны быть регулярными, как в случае ренты. Внутренняя скорость оборота – это процентная ставка дохода, полученного от инвестиций, состоящих из выплат (отрицательные значения) и поступлений (положительные значения), которые происходят в регулярные периоды времени.
MIRR	МВСД	Возвращает модифицированную внутреннюю скорость оборота средств для ряда последовательных периодических операций с наличными. Функция MIRR учитывает, как скорость инвестиции, так и доход, получаемый от реинвестирования.
NPer	КПЕР	Вычисляет общее количество периодов выплаты для данного вклада на основе периодических постоянных выплат и постоянной процентной ставки.
NPV	ЧПС	Возвращает чистый текущий объём вклада, вычисляемый на основе ряда последовательных поступлений наличных и нормы амортизации
Pmt	ПЛТ	Вычисляет величину постоянной периодической выплаты ренты (например, регулярных платежей по займу) при постоянной процентной ставке.
PPmt	ОСПЛТ	Возвращает величину выплаты на данный период на основе постоянных платежей, осуществляемых через определённые промежутки времени и постоянной процентной ставки.
PV	ПС	Возвращает текущий объём вклада на основе периодических платежей, осуществляемых через определённые промежутки времени. Аналогична функции NPV, разница в том, что денежные взносы постоянны и взносы можно вносить в начале или конце периода.
Rate	СТАВКА	Вычисляет процентную ставку за один период, необходимую для накопления определённой суммы путём постоянных взносов.
SLN	АПЛ	Возвращает величину амортизации имущества за один период времени, используя метод равномерной амортизации.
SVD	АЧС	Возвращает амортизацию имущества для указанного периода.

Игра 1.

Разработаем следующую игру «Бросание картинок». На рабочем листе следует разместить пять окон для изображений. При этом два окна будут основными – расположены слева и большие по размеру. Другие 3 окна имеют меньшие размеры и расположены в правой части рабочего листа (они будут содержать эталонные изображения). Также на рабочем листе расположены 2 кнопки с надписями Бросок и Начать игру снова, и два текстовых поля. В одно текстовое поле пишется сумма баллов от результата броска, в другое – количество бросков. Смысл игры в следующем. Предварительно в левые три окна вводятся разные картинки (Режим конструктора\Свойства). При щелчке на кнопку Бросок в каждое из двух основных окон для изображений должна попасть случайным образом одна из картин из набора эталонных. Если картинки в основных (двух левых окнах) окнах совпадают, то в текстовое поле добавляется 1 балл, а содержимое другого текстового поля (количество бросков) при каждом броске увеличивается на 1.



Листинг программы для процедуры, работающей по щелчку на кнопку Бросок

```
Public s As Single, n As Long
Private Sub CommandButton1_Click()
Randomize
a = CInt(Rnd * 3 + 0.5)
b = CInt(Rnd * 3 + 0.5)
If a = 1 Then
    Image1.Picture = Image3.Picture
ElseIf a = 2 Then
    Image1.Picture = Image4.Picture
Else
    Image1.Picture = Image5.Picture
End If
If b = 1 Then
    Image2.Picture = Image3.Picture
ElseIf b = 2 Then
    Image2.Picture = Image4.Picture
```

```

Else
Image2.Picture = Image5.Picture
End If
If a = b Then
s = s + 1
End If
n = n + 1
TextBox1.Text = CInt(Format(s, "Fixed"))
TextBox2.Text = CInt(Format(n, "Fixed"))
End Sub

```

Листинг процедуры, работающей по щелчку на кнопку Начать игру

```

Private Sub CommandButton2_Click()
s = 0
n = 0
TextBox1.Text = CInt(Format(s, "Fixed"))
TextBox2.Text = CInt(Format(n, "Fixed"))
End Sub

```

Игра 2.

Программа для игры Орёл – Решка. Создадим два переключателя и кнопку Бросание монеты.

A	B	C	D	E	F
	☐ Орёл				
			Бросание монеты		
	☐ Решка				

Листинг процедуры под Щелчка по кнопке Бросание монеты

```

CommandButton1 Click
Public s As Long, k As Long
Private Sub CommandButton1_Click()
    'Моделирование бросания монеты: 1 - орёл, 0 - решка
    Randomize
    If OptionButton1.Value = False And OptionButton2.Value = False Then
        MsgBox "Выбери Орёл или Решка"
        GoTo M1
    End If
    Moneta = Int(2 * Rnd)
    s = s + 1
    'Сравнение результата бросания монеты компьютером,
    'когда игрок загадал "орёл"
    If OptionButton1.Value = True Then
        If Moneta = 0 Then
            MsgBox "Не угадал!"
        Else
            MsgBox "Угадал!"
            k = k + 1
        End If
    Else
        'Сравнение результата бросания монеты компьютером,
        'когда игрок загадал "решка"
        If Moneta = 1 Then
            MsgBox "Не угадал!"
        Else
            MsgBox "Угадал!"
            k = k + 1
        End If
    End If
    OptionButton1.Value = False
    OptionButton2.Value = False
    If s = 5 Then
        s = 0
        MsgBox "Произведено 5 бросаний " & " Угадали " & k & " раза"
        k = 0
    End If
M1: End Sub

```

Перед процедурой CommandButton1_Click описателем Public описаны переменные s и k. В переменную s накапливается, сколько раз бросали монету, а в переменной k накапливается, сколько раз отгадали правильно, Орёл или Решка.

Если монета брошена 5 раз, то печатается, что произведено 5 бросаний и сколько раз угадали при 5 бросаниях будет Орёл или Решка. Функция Randomize –включает счётчик случайных чисел. Функция Rnd – выдаёт случайное число со значением от 0 до 1, функция Int преобразует число в целый тип, отбрасывая дробную часть. Таким образом, результат выражения INT(2* Rnd()) будет или 0 или 1.

Библиографический список

1. Информатика для экономистов: учебник /под общ. ред. В.М. Матюшка. – М.: ИНФРА-М, 2009. – 879 с.
2. Экономическая информатика: учеб, пособие/ под ред. Д.В.Чистова. – М.: Кнорус, 2009. –512 с.
3. Кашаев, С. Офисные решения с использованием Microsoft Excel 2007 и VBA / С.Кашаев. – СПб. Питер, 2009. – 352с. ил. . (+CD)
4. Гарнаев, А. EXCEL, VBA, INTERNET в экономике и финансах /А.Гарнаев. – СПб.: БХВ-Петербург, 2005. –796 с.
5. Васильев, А.Н. EXCEL 2007 + видеокурс на примерах / А.Н. Васильев. – СПб.: БХВ Петербург, 2007. – 651 с.
6. Васильев, А.Н. Фининсовое моделирование и оптимизация средствами Excel 2007 / А.Н. Васильев. – СПб.: Питер, 2009. – 320 с., ил. (+CD)
7. Пикуза. В. Экономические и финансовые расчеты в EXCEL / В. Пикуза. А. Гаращенко. – СПб.: Питер, 2003. – 396 с.
8. Глушаков, С.В. Microsoft Excel 2007 / С.В. Глушаков, А.С.Сурядный. – М.: АСТ:, 2009. – 512 с.– (Эффективное руководство).

Оглавление

Тема 1. Введение в VBA	3
Знакомство с редактором VBA	4
Основы программирования VBA	5
Функции преобразования типов.....	6
Операторы, выражения, операции	7
Лабораторная работа №1.	9
Самостоятельная работа 1.	12
Тема 2. Встроенные диалоговые окна.....	13
Лабораторная работа 2.....	15
Самостоятельная работа 2.	17
Тема 3. Встроенные функции для обработки строк	17
Лабораторная работа 3.....	18
Самостоятельная работа 3	19
Тема 4. Создание макросов с помощью макрорекордера.	19
Лабораторная работа 4.....	19
Самостоятельная работа 4	21
Тема 5. Режим конструктора и элементы ActiveX	22
Лабораторная работа 5.....	23
Элементы управления Надпись и Текстовое поле	25
Самостоятельная работа 5	27
Тема 6. Элементы управления <i>Поле со списком и Список</i>.....	28
Лабораторная работа 6.....	28
Самостоятельная работа 6	31
Тема 7. Работа с изображениями	33
Лабораторная работа 7.....	33
Самостоятельная работа 7	35
Тема 8. Работа с переключателями и флажком.	36
Лабораторная работа 8.....	36
Самостоятельная работа 8	37
Тема 9. Работа с диаграммами.....	39
Лабораторная работа 9.....	39
Самостоятельная работа 9	41
Тема 10. Работа со счётчиком. Финансовые задачи.....	41
Лабораторная работа 10.....	42
Самостоятельная работа 10	47
Тема 11. Организация кадрового учёта.....	48
Тема 12. Работа со строками, цветом, ячейками. Создание бегущей строки	51
Лабораторная работа 12.....	52
Тема 13. Применение технологии при решении макроэкономических задач.....	57
Приложения.....	61

Подписано в печать 6.03.2014.	Усл. печ. л. 4,2	Тираж	экз.
Печать офсетная.	Бумага писчая.	Заказ № ____.	

Отпечатано: РИО ВоГУ, г. Вологда, ул. Ленина, 15