

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)  
Кафедра экономической математики, информатики и статистики (ЭМИС)

КЛАССЫ И ОБЪЕКТЫ В VISUAL BASIC FOR APPLICATION  
Отчет по практической работе по дисциплине «Информационные  
технологии»

Студент группы 549



Баулин С.К.

«\_\_» \_\_\_\_\_ 2020 г.

Старший преподаватель  
кафедры ЭМИС

\_\_\_\_\_ Афанасьева И. Г.

«\_\_» \_\_\_\_\_ 2020 г.

Томск 2020

## **Практическая работа № 8**

### **Классы и объекты в Visual Basic for Application**

#### **Цель работы**

Знакомство с объектами VBA. На практике рассмотреть основные операции и функции.

#### **Теоретический материал, для освоения темы**

##### **Объектная модель**

Объектная модель MS Excel представляет собой иерархию объектов, подчиненных одному объекту Application, который соответствует самому приложению Office. Многие из этих объектов собраны в библиотеке объектов конкретного приложения, но некоторые из них, например, объект Assistant, входят в библиотеку объектов Office, которая является общей для всех офисных приложений.

##### **Полная и неявная ссылка на объект**

Полная ссылка на объект состоит из ряда имен вложенных последовательно друг в друга объектов. Разделителями имен объектов в этом ряду являются точки, ряд начинается с объекта Application и заканчивается именем самого объекта. Например, полная ссылка на ячейку **A1** рабочего листа **Продажи** рабочей книги с именем **Архив** имеет вид:

`Application.Workbooks("Архив").Worksheets("Продажи").Range("A1")`

Приводить каждый раз полную ссылку на объект совершенно не обязательно. Обычно достаточно ограничиться только неявной ссылкой на объект. В неявной ссылке, в отличие от полной, объекты, которые активны в данный момент, как правило, можно опускать. В рассмотренном случае, если ссылка на ячейку **A1** дана в программе, выполняемой в среде Excel, то ссылка

на объект Application может быть опущена, т.е. достаточно привести относительную ссылку:

```
Workbooks("Архив").Worksheets("Продажи").Range("A1")
```

Если в этом примере ссылки рабочая книга **Архив** является активной, то ссылку можно еще сократить:

```
Worksheets("Продажи").Range("A1")
```

Если и рабочий лист **Продажи** активен, то в относительной ссылке вполне достаточно ограничиться упоминанием только диапазона **A1**:

```
Range("A1")
```

## Основные объекты VBA

Объект Application (приложение) является главным в иерархии объектов Excel и представляет само приложение Excel. Он имеет более 120 свойств и методов и методов. Эти свойства и методы предназначены для установки общих параметров приложения Excel. Кроме того, объект Application позволяет вызывать более 400 встроенных функций рабочего листа при помощи конструкции вида:

```
Application.ФункцияРабочегоЛиста (Аргументы)
```

Например, для определения постоянных периодических платежей при постоянной процентной ставке можно воспользоваться следующей функцией:

```
Application.Pmt (Аргументы)
```

## Свойства объекта Application

ActiveWorkbook	Возвращает активный объект: рабочую книгу, лист, ячейку, диаграмму диалоговое окно. Свойство ActiveCell содержится в ActiveSheet, а свойства ActiveChart и ActiveDialog в ActiveWorkbook. Например: ActiveCell.Value = "Привет!" Здесь в активную ячейку вводится фраза <b>Привет!</b>
ActiveSheet	
ActiveCell	
ActiveChart	
ActiveDialog	

ThisWorkbook	Возвращает рабочую книгу, содержащую выполняющийся в данный момент макрос.
Calculation	Устанавливает режим вычислений
Caption	Возвращает текст в строке имени главного окна Excel. Установка свойства равным Empty возвращает заголовок, используемый по умолчанию
DisplayAlerts	Значения: True – отображаются встроенные предупреждения о работе программы, False – не отображаются предупреждения
DisplayFormulaBar	Значения: True – строка формул выводится в окне Excel, False – не выводится
DisplayScrollBars	Значения: True – полоса прокрутки выводится в окне Excel, False – полосы прокрутки не отображаются
Height	Высота приложения в пунктах
Width	Ширина приложения в пунктах
Right	Расстояние в пунктах от правой границы окна приложения до правого края окна
Left	Расстояние в пунктах от левой границы окна приложения до левого края окна
Top	Расстояние в пунктах от верхней границы приложения
WindowState	Устанавливает размер окна

### Методы объекта Application

Calculate	Вызывает принудительное вычисление во всех открытых рабочих книгах.
Run	Запускает на выполнение программу или макрос.
OnKey	Устанавливает сочетание клавиш для выполнения выбранной процедуры. OnKey(Key, Procedure)
Quit	Закрывает приложение.
OnTime	Назначает выполнение процедуры на определенное время.

### События объекта Application

NewWorkbook	При создании новой рабочей книги
WorkbookActivate	При активизации рабочей книги
WorkbookBeforeClose	Перед закрытием рабочей книги
WorkbookBeforePrint	Перед печатью рабочей книги.
WorkbookBeforeSave	Перед сохранением рабочей книги.
WorkbookNewSheet	При добавлении нового листа
WorkbookOpen	При открытии рабочей книги

## Объект Workbook (книга)

### Свойства объекта Workbook

ActiveSheet	Возвращает активный лист книги.
ActiveDialog	Возвращает активное диалоговое окно
ActiveChart	Возвращает активную диаграмму Например: <b>MsgBox “Название диаграммы” &amp; ActiveChart.Name</b> выводит в диалоговом окне имя активной диаграммы
Saved	True – если не производились изменения в документе со времени его последнего сохранения, False – в противном случае. Например: <b>If WorkBook.Saved=True Then</b> <b>MsgBox “ удачно сохранились! “</b> <b>End If</b>
WriteReserved	True – если книгу закрыть для записи, False – в противном случае

### Методы объекта Workbook

Activate	Активизирует рабочую книгу
Add	Создает новую книгу
Protect	Защищает рабочую книгу от внесения в неё изменений. <b>Protect(Password, Structure, Windows)</b> Например: <b>ActiveWorkbook.Protect Password:= “Не влезай!”</b>
UnProtect	Снятие защиты с листа <b>UnProtect(Password)</b>
Close	Закрытие рабочей книги
Open	Открытие существующей книги
Save	Сохранение рабочей книги
SaveAs	Сохранение книги с другими параметрами
PrintPreview	Предварительный просмотр

### События объекта Workbook

BeforeClose	При закрытии книги
BeforePrint	Перед печатью рабочей книги
BeforeSave	Перед сохранением рабочей книги
NewSheet	При добавлении нового листа.
Open	При открытии рабочей книги.
SheetActivate	При активизации любого рабочего листа.
SheetDeactivate	Когда рабочий лист теряет фокус.

## Объект Worksheet (лист)

### Свойства объекта Worksheet

Name	Возвращает имя рабочего листа.
Visible	Отображает «видимость» или «невидимость» листа (True, False – соответственно)
ActiveCell	Возвращает активную ячейку активного листа.
StandartHeight	Возвращает стандартную высоту всех строк рабочего листа
UsedRange	Возвращает диапазон (объект Range), который содержит данные.
Union	Возвращает диапазон, являющийся объединением нескольких диапазонов.

### Методы Worksheet

Activate	Активизирует указанный рабочий лист. <b>Worksheets(1).Activate</b>
Add	Создает новый рабочий лист <b>Add(Before, After, Count, Type)</b>
Delete	Удаляет рабочий лист
Protect	Защищает рабочий лист от внесения в него изменений
Unprotect	Снятие защиты с рабочего листа
Copy	Копирование рабочего листа в другое место <b>Copy (Before, After). Допустимо использование только одного аргумента</b>
Move	Перемещение рабочего листа в другое место
Evaluate	Преобразует выражение в объект или значение. Используется при вводе формул и ячеек из диалоговых окон

### Объекты Range (диапазон)

При работе с объектом Range необходимо помнить, как в Excel ссылаются на ячейку рабочего листа (см. относительные и абсолютные ссылки).

Так как ячейка является частным случаем диапазона, состоящим только из единственной ячейки, объект Range также позволяет работать с ней. Объект Cells(ячейки) – это альтернативный способ работы с ячейкой. Например, ячейка A2 как объект описывается Range(“A2”) или Cells(1,2).

## Свойства объекта Range

Value	Возвращает значение из ячейки или в ячейки диапазона. Например: <b>h=Range ("C1").Value</b>
Name	Возвращает имя диапазона.
WrapText	Позволяет переносить текст при вводе в диапазон. <b>With Range ("F3")</b> <b>.Value= "Привет всем!"</b> <b>.WrapText=True</b> <b>End With</b>
Comment	Добавление комментария к данным в ячейки.
Font	Возвращает объект Шрифт с указанными параметрами.
Formula	Возвращает формулу в формате A1.
Text	Возвращает содержание диапазона в текстовом формате.

## Наиболее часто используемые методы Range

Clear	Очистка диапазона. <b>Range ("A1:B1").Clear</b>
Copy	Копирует диапазон в другой или буфер обмена.
Cut	Копирует диапазон в другой или буфер обмена с удалением.
Delete	Удаляет диапазон.
Insert	Вставка ячейки или диапазона ячеек. <b>Worksheets ("Лист1").Rows(4).Insert</b>
Select	Выделение диапазона.

Заполнение произвольного диапазона данных по столбцам можно осуществить следующим образом для перебора адреса ячейки с A3 по A5:

```
For i = 1 To 3
```

```
    A = "A" & i + 2
```

```
    Range([A]).Value = InputBox("Введи данные " & i)
```

```
Next
```

## Задания на лабораторную работу

**Задание 1.** Создать приложение в VBA, позволяющее определить была ли сохранена открытая рабочая книга.

## Задание 2.

Создать приложение в VBA, позволяющее произвести расчет амортизации различными методами. Данные для расчета брать с одного листа текущей книги, а выводить таблицу начисления амортизации в отдельную сохраненную книгу.

### I. Применение функции

АПЛ(нач\_стоимость;ост\_стоимость;время\_эксплуатации)

Стоимость актива составляет 10 000 руб., срок эксплуатации актива – 5 лет, а ликвидационная стоимость – 2000 руб. Создайте таблицу начисления износа, в которой бы указывались суммы ежегодной амортизации и значения остаточных стоимостей в конце каждого года, применяя линейный метод.

### II. Применение функции

АСЧ(нач\_стоимость;ост\_стоимость;время\_эксплуатации; период)

Стоимость актива составляет 46 000 руб., срок эксплуатации актива – 7 лет, а ликвидационная стоимость – 9000 руб. Создайте таблицу начислений износа, в которой бы указывались суммы ежегодной амортизации и значения остаточных стоимостей в конце каждого года, применяя метод «суммы (годовых) чисел».

Результат самостоятельной работы представлен ниже.

1 задание представлено на скриншотах 1.1, 1.2.

```
Sub WorkBooksSaved()  
,  
' WorkBooksSaved Макрос  
,  
' Сочетание клавиш: Ctrl+Shift+O  
,  
    If ActiveWorkbook.Saved = True Then  
        MsgBox "Книга была сохранена"  
    Else  
        MsgBox "Книга не была сохранена"  
    End If  
  
End Sub
```

Рисунок 1.1 – Скриншот кода программы для задания 1



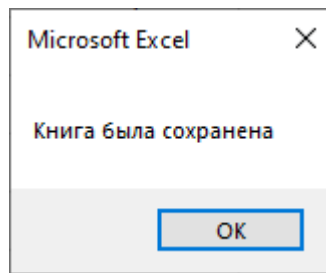


Рисунок 1.2 – Скриншот результата работы макроса для задания 1

Задание 2 представлено на рисунках 2.1, 2.2.

```

Sub CountAm()
    Dim AllCost1, LiqCost1, AllCost2, LiqCost2 As Currency
    Dim Time1, Time2 As Integer
    Dim a, b, c As String
    Workbooks.Open (ThisWorkbook.Path & "\macros_practica8.xlsm")
    AllCost1 = Range("B1").Value
    LiqCost1 = Range("B2").Value
    Time1 = Range("B3").Value
    AllCost2 = Range("E1").Value
    LiqCost2 = Range("E2").Value
    Time2 = Range("E3").Value
    If Dir(ThisWorkbook.Path & "\Расчет амортизации.xlsx") = "" Then
        Workbooks.Add
        ActiveWorkbook.SaveAs ThisWorkbook.Path & "\Расчет амортизации.xlsx"
        MsgBox "Книга была добавлена"
    Else
        Workbooks.Open (ThisWorkbook.Path & "\Расчет амортизации.xlsx")
    End If
    'АПЛ
    Range("A1").Value = "Линейный способ"
    Range("A2").Value = "Год"
    Range("A:A").EntireColumn.AutoFit
    Range("B2").Value = "Ежегодная амортизация"
    Range("B:B").EntireColumn.AutoFit
    Range("C2").Value = "Остаточная стоимость"
    Range("C:C").EntireColumn.AutoFit
    For i = 1 To Time1
        a = "A" & i + 2
        b = "B" & i + 2
        c = "C" & i + 2
        Range([a]).Value = i
        Range([b]).FormulaLocal = "=АПЛ(" & AllCost1 & "," & LiqCost1 & "," & Time1 & ")"
        Range([c]).FormulaLocal = "=" & AllCost1 & "-СУММ(B3:B" & i + 2 & ")"
    Next i
    'АСЧ
    Range("E1").Value = "Метод суммы годовых чисел"
    Range("E2").Value = "Год"
    Range("E:E").EntireColumn.AutoFit
    Range("F2").Value = "Ежегодная амортизация"
    Range("F:F").EntireColumn.AutoFit
    Range("G2").Value = "Остаточная стоимость"
    Range("G:G").EntireColumn.AutoFit
    For i = 1 To Time2
        a = "E" & i + 2
        b = "F" & i + 2
        c = "G" & i + 2
        Range([a]).Value = i
        Range([b]).FormulaLocal = "=АСЧ(" & AllCost2 & "," & LiqCost2 & "," & Time2 & "," & i & ")"
        Range([c]).FormulaLocal = "=" & AllCost2 & "-СУММ(F3:F" & i + 2 & ")"
    Next i
    ActiveWorkbook.Save
End Sub

```

Рисунок 2.1 – Скриншот кода макроса для задания 2

A		B		C	D	E		F	G
1	Линейный способ					Метод суммы годовых чисел			
2	Год	Ежегодная амортизация	Остаточная стоимость		Год		Ежегодная амортизация	Остаточная стоимость	
3	1	1 600,00 Р	8 400,00 Р			1	9 250,00 Р	36 750,00 Р	
4	2	1 600,00 Р	6 800,00 Р			2	7 928,57 Р	28 821,43 Р	
5	3	1 600,00 Р	5 200,00 Р			3	6 607,14 Р	22 214,29 Р	
6	4	1 600,00 Р	3 600,00 Р			4	5 285,71 Р	16 928,57 Р	
7	5	1 600,00 Р	2 000,00 Р			5	3 964,29 Р	12 964,29 Р	
8						6	2 642,86 Р	10 321,43 Р	
9						7	1 321,43 Р	9 000,00 Р	
10									
11									
12									
13									
14									
15									
16									

Рисунок 2.2 – Скриншот результата работы макроса для задания 2

Вывод: проведена работа с объектами VBA. Рассмотрены основные операции и функции.