

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Федеральное государственное образовательное
бюджетное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М. А. БОНЧ-БРУЕВИЧА»**

Е. В. Стригина

***Программирование
IT сервисов предприятия***

**Методические указания
к выполнению лабораторных работ**

СПб ГУТ)))

**САНКТ-ПЕТЕРБУРГ
2013**

УДК 004.65(075.8)
ББК 32.973.26–018.2я73
C85

Рецензент
доктор технических наук, профессор СПбГУТ
А. Д. Сотников

*Рекомендованы к печати
Редакционно-издательским советом СПбГУТ*

Стригина, Е. В.
C85 Программирование ИТ сервисов предприятия: методические указания к выполнению лабораторных работ / Е. В. Стригина. ; РИЦ СПбГУТ. СПб., 2013. – 40 с.

Рассмотрены возможности использования языка Visual Basic for Applications для программирования приложений MS Excel 2007. Задания к лабораторным работам дают возможность познакомиться с основными конструкциями и операторами языка, выполнить обработку табличных данных и создать интерфейс пользователя с помощью форм.

Предназначены для студентов, обучающихся по направлению (специальности) 080500.62 «Бизнес-информатика».

**УДК 004.65(075.8)
ББК 32.973.26–018.2я73**

© Стригина Е. В., 2013
© Федеральное государственное
образовательное бюджетное учреждение
высшего профессионального образования
«Санкт-Петербургский государственный
университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича», 2013

СОДЕРЖАНИЕ

| | |
|--|----|
| Лабораторная работа 1. Знакомство с методами и свойствами объектов Workbook и Worksheet. Создание ссылок на объекты | 4 |
| Лабораторная работа 2. Знакомство с операторами ветвления If и Select Case | 6 |
| Лабораторная работа 3. Знакомство с операторами циклов For...Next и Do...Loop | 8 |
| Лабораторная работа 4. Знакомство со способами отладки программ в редакторе VBA | 10 |
| Лабораторная работа 5. Работа с текстом и датами | 13 |
| Лабораторная работа 6. Знакомство со свойствами и методами объекта Range | 15 |
| Лабораторная работа 7. Знакомство с использованием простой формы для ввода данных в таблицу | 20 |
| Лабораторная работа 8. Проект создания «Базы данных» средствами VBA в Excel 2007 (1). Создание структуры таблицы и экранной формы для взаимодействия с таблицей | 21 |
| Лабораторная работа 9. Проект создания «Базы данных» средствами VBA в Excel 2007 (2). Добавление данных в таблицу с проверкой их корректности | 24 |
| Лабораторная работа 10. Проект создания «Базы данных» средствами VBA в Excel 2007 (3). Просмотр и удаление записей | 27 |
| Лабораторная работа 11. Проект создания «Базы данных» средствами VBA в Excel 2007 (4). Модификация формы. Использование функций даты и времени | 29 |
| Лабораторная работа 12. Знакомство с методами обработки данных с помощью сортировки | 31 |
| Лабораторная работа 13. Знакомство с методами обработки данных с помощью фильтрации. Графическое представление данных | 32 |
| Лабораторная работа 14. Знакомство с процедурами с передачей параметров | 36 |
| Список литературы | 38 |

Лабораторная работа 1

ЗНАКОМСТВО С МЕТОДАМИ И СВОЙСТВАМИ ОБЪЕКТОВ WORKBOOK И WORKSHEET. СОЗДАНИЕ ССЫЛОК НА ОБЪЕКТЫ

1. Создайте новый файл в Excel 2007 (Excel 2010) в папке, путь к которой проследите в проводнике, и сделайте его по возможности коротким (например, E:\Petrov\vba1.xlsx).
2. Откройте редактор VBA.
3. Познакомьтесь с созданием ссылки на объект Workbook и с его методами.
 - 3.1. Откройте область редактирования кода Лист1, щелкнув дважды на Лист1 в окне проектов.
 - 3.2. Вставьте в область кода процедуру, используя команду меню Insert – Procedure...
 - 3.3. Введите код процедуры add_book.

```
Public Sub add_book()
    'описание типа ссылки wb
    Dim wb As Workbook
    'создание новой книги и ссылки на нее
    Set wb = Workbooks.Add
    'сохранение книги под именем mybook (без расширения - добавляется автоматически)
    wb.SaveAs Filename:="mybook"
    'закрытие этого файла с сохранением изменений
    wb.Close SaveChanges:=True
End Sub
```

- 3.4. Дополните имя файла mybook, указав полный путь к созданной Вами папке (расширениеxlsx добавляется автоматически).
- 3.5. Выполните процедуру, выбрав команду Run – Run Sub/UserForm или нажав клавишу F5.
- 3.6. Убедитесь, что в Вашей папке находится рабочая книга mybook.xlsx.
- 3.7. Предъявите результат преподавателю
4. Познакомьтесь с созданием ссылки на объект Worksheet и с его методами и свойствами.
 - 4.1. Введите код процедуры add_sheet.

```
Public Sub add_sheet()
```

```

'описание типов ссылок wb1 и ws
Dim wb1 As Workbook
Dim ws As Worksheet

Set wb1 = Workbooks.Open("mybook.xlsx")
'создание нового листа и ссылки на него
Set ws = Worksheets.Add
' изменение имени листа
ws.Name = "Total"
'удаление Лист2
wb1.Worksheets("Лист2").Delete

'заполнение данными ячеек A1:A3
ws.Range("A1").Value = 25
ws.Range("A2").Value = 30
ws.Cells(3, 1) = "=sum(a1:a2)"
'сохранение книги
wb1.Save
End Sub

```

4.2. Выполните процедуру в пошаговом режиме, выбрав команду Debug – Step Into или нажимая клавишу F8.

4.3. Откройте созданную книгу и проанализируйте ее содержание.

4.4. Предъявите результат преподавателю

5. Введите код процедуры close_wb и проанализируйте его.

```

Public Sub close_wb()
    Dim wb As Workbook
    Set wb = Workbooks("mybook.xlsx")
    wb.Close SaveChanges:=True
    'или просто без ссылок
    'Workbooks("mybook.xlsx").Close SaveChanges:=True
End Sub

```

5.1. Выполните процедуру, выбрав команду Run – Run Sub/UserForm или нажав клавишу F5.

5.2. Убедитесь в наличии в Вашей папке закрытой книги mybook.xlsx и удалите ее.

6. Создайте процедуру Run_all, которая будет выполнять последовательно процедуры add_book и add_sheet.

6.1. Создайте в листе элемент управления – кнопку и присвойте ей

созданную процедуру.

- 6.2. Выполните процедуру Run_all.
- 6.3. Добавьте в нее обращение к процедуре close_wb и убедитесь в ее работоспособности.
7. Перечислите используемые в процедурах методы и свойства объектов Workbook и Worksheet.
8. Предъявите результат преподавателю

Лабораторная работа 2

ЗНАКОМСТВО С ОПЕРАТОРАМИ ВЕТВЛЕНИЯ IF И SELECT CASE

1. Создайте в Вашей папке файл под именем If_case_Имя.xlsx, где Имя – Ваше имя.

2. В ячейки A2, B2 и C2 введите фамилию, пол и возраст человека. В зависимости от его возраста и пола определите статус: школьник – от 7 до 17 лет, студент – от 17 до 22 лет, работающий – до 55 или 60 лет, для женщин и мужчин, соответственно, и пенсионер – старше 55 или 60 лет также в зависимости от пола. В клетке D2 отобразите полученный статус.

2.1. В поле кода для первого листа этой книги, создайте процедуру, код которой приведен в процедуре Data_if и реализует приведенную выше задачу. Проанализируйте код процедуры.

```
Public Sub Data_If()
    Dim status As String
    Dim age As Byte
    Dim sex As String * 1

    Sheets("Лист1").Activate
    Range("a2").Value = InputBox("Введите имя")
    sex = InputBox("Введите пол")
    Range("b2").Value = sex
    age = InputBox("Введите возраст")
    Range("c2").Value = age

    If age > 7 And age < 17 Then
        status = "школьник"
    ElseIf age > 18 And age < 22 Then
        status = "студент"
    ElseIf age > 22 And ((age < 55 And sex = "ж") Or (age < 60 And sex = "м")) Then
        status = "работающий"
    Else
        status = "пенсионер"
    End If

    Range("d2").Value = status
End Sub
```

```

    status = "работающий"
    ElseIf (age > 55 And sex = "ж") Or (age > 60 And sex = "м") Then
        status = "пенсионер"
    End If
    Range("d2").Value = status
End Sub

```

2.2. Выполните процедуру и убедитесь в ее работоспособности.

2.3. Модифицируйте процедуру так, чтобы появились еще два статуса: для возраста менее 7 лет – «дошкольник», а для возраста, большего 90 лет – «долгожитель» и сохраните ее под другим именем.

2.4. Убедитесь в работоспособности процедуры.

3. В упрощенных условиях предыдущего примера используйте конструкцию **Select Case**. Для простоты будем считать статусом людей старше 55 лет – «пенсионер», людей, которым исполнилось 100 лет – «долгожитель», а детей до 7 лет – «дошкольник». В клетке D2 отобразите полученный статус.

3.1. В поле кода для второго листа этой книги, создайте процедуру, код которой приведен в процедуре **Data_case**. Проанализируйте код процедуры.

```

Public Sub Data_case()
    Dim status As String
    Dim age As Byte

    Sheets("Лист2").Activate
    Range("a2").Value = InputBox("Введите фамилию")
    age = InputBox("Введите возраст")
    Range("c2").Value = age
    Select Case age
        Case Is<7
            status = "дошкольник"
        Case 7 To 17
            status = "школьник"
        Case 18 To 22
            status = "студент"
        Case 23 To 55
            status = "работающий"
        Case Is > 55
            status = "пенсионер"
        Case 100
    End Select
    Range("d2").Value = status
End Sub

```

```

status = "должжитель"
Case Else
    status = "ошибка ввода"
End Select
Range("d2").Value = status
End Sub

```

3.2. Убедитесь в работоспособности процедуры.

3.3. Модифицируйте процедуру так, чтобы возраст классифировался по трем статусам – «молодой человек» - до 30 лет, «средний возраст» - до 45 лет и «пожилой человек» - старше 45 лет.

3.4. Убедитесь в работоспособности процедур и предъявите результаты преподавателю.

Лабораторная работа 3

ЗНАКОМСТВО С ОПЕРАТОРАМИ ЦИКЛОВ For...Next и Do...Loop

1. Создайте в Вашей папке файл под именем For1_Имя.xlsx, где Имя – Ваше имя.

2. В поле кода для первого листа этой книги, создайте процедуру, содержащую два цикла For...Next для формирования заголовков строк и столбцов таблицы. Заголовки столбцов должны располагаться в диапазоне B1:D1 и содержать номера годов 2010, 2011 и 2012, а заголовки строк должны располагаться в диапазоне A2:A5 и содержать номера кварталов: Кв.1, Кв.2, Кв.3, Кв.4.

2.1. Сравните ее с приведенной в практикуме процедурой SetUpTable.

```

Sub SetUpTable()
    'формирование заголовков столбцов и строк
    Worksheets("лист1").Activate
    For год = 2 To 4
        Cells(1, год).Value = 2008 + год
    Next год

    For квартал = 1 To 4
        Cells(квартал + 1, 1).Value = "Кв." & квартал
    Next квартал
End Sub

```

2.2. Выполните процедуру в пошаговом режиме (F8) и убедитесь в ее работоспособности.

2.3. Создайте вторую процедуру для заполнения таблицы данными. Обратите внимание на вложенный цикл (двойной цикл), который формирует номера строки и столбца ячеек таблицы.

2.4. Сравните ее с приведенной в практикуме процедурой InputData.

```
Public Sub InputData()
    Dim строка as byte, столбец as byte
    ' ввод данных в таблицу
    Worksheets("лист1").Activate
    For строка = 2 To 5
        For столбец = 2 To 4
            Cells(строка, столбец).Value = InputBox("Введите значение")
        Next столбец
    Next строка
End Sub
```

2.5. Выполните процедуру в пошаговом режиме (F8) и убедитесь в ее работоспособности.

2.6. Модифицируйте процедуры так, чтобы число годов в таблице увеличилось до пяти, а заполнение ячеек происходило числами натурального ряда от 1 до 5 по каждому кварталу. Убедитесь в работоспособности программы и предъявите результат преподавателю.

2.7. Заполните ячейки таблицы случайными числами из диапазона 0 – 100 и округлите их до двух значащих чисел. (Используйте функцию RND для получения случайных чисел в диапазоне от 0 до 1 и функцию Application.Round(округляемое число, количество знаков после запятой)).

2.8. Найдите максимальное значение данных таблицы и поместите его в ячейку A10. Убедитесь в работоспособности программы и предъявите результат преподавателю.

3. Создайте в Вашей папке файл под именем Do_Loop_Имя.xlsx, где Имя – Ваше имя.

3.1. Используйте *четыре* варианта циклов Do...Loop для ввода своего имени в ячейки A1:A4, соответственно.

3.2. Используйте бесконечный цикл Do While ...Loop для ввода в ячейку A10 своего имени. **Выход из цикла должен быть предусмотрен оператором Exit Do.** Выполните данную процедуру в **пошаговом** режиме.

3.3. Предъявите результат преподавателю.

4. Создайте на Лист1 новой книги заголовки строк и столбцов таблицы так, как показано на рис. 1 и сохраните файл под именем For2_Имя.xlsx, где Имя – Ваше имя.

4.1. Заполните таблицу случайными числами в диапазоне 0 - 150, используя функцию RND() и Round(), округляя полученные значения до целых

значений (0 знаков после запятой) (рис. 1).

4.2. Выделите значения в ячейках таблицы, превышающие 90, применив к ним полужирное начертание - Cells(I,j).Font.Bold=True и установив для них фоновое выделение - Cells(I,j).Interior.ColorIndex=35 (рис. 1).

4.3. Вычислите количество месяцев для каждой фирмы, в которых значения в ячейках превышают 90, и поместите результат вычислений в первый свободный столбец справа от таблицы в виде, показанном на рис. 1. Предусмотрите очистку формата при многократном выполнении программы.

| A | янв | фев | мар | апр | май | июн | H |
|-------------|-----|-----|-----|-----|-----|-----|-------|
| 1 | | | | | | | |
| 2 O'Кей | 70 | 45 | 93 | 97 | 40 | 42 | Num=2 |
| 3 Карусель | 124 | 124 | 88 | 148 | 137 | 34 | Num=4 |
| 4 Лента | 104 | 147 | 37 | 80 | 16 | 150 | Num=3 |
| 5 Пятерочка | 101 | 2 | 86 | 15 | 15 | 120 | Num=2 |
| 6 | | | | | | | |

Рис. 1

4.4. Предъявите результат преподавателю.

5. В новой книге (или на другом листе предыдущей) создайте процедуру, которая будет отображать цвета фона, устанавливаемые с помощью свойства ColorIndex. Известно, что значения этого свойства лежат в диапазоне от 1 до 56. Результат работы процедуры показан на рис. 2.

6. Предъявите результат преподавателю.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
|---|---|----|----|----|----|----|----|----|---|---|---|---|---|---|---|
| 1 | 1 | 8 | 15 | 22 | 29 | 36 | 43 | 50 | | | | | | | |
| 2 | 2 | 9 | 16 | 23 | 30 | 37 | 44 | 51 | | | | | | | |
| 3 | 3 | 10 | 17 | 24 | 31 | 38 | 45 | 52 | | | | | | | |
| 4 | 4 | 11 | 18 | 25 | 32 | 39 | 46 | 53 | | | | | | | |
| 5 | 5 | 12 | 19 | 26 | 33 | 40 | 47 | 54 | | | | | | | |
| 6 | 6 | 13 | 20 | 27 | 34 | 41 | 48 | 55 | | | | | | | |
| 7 | 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | | | | | | | |
| 8 | | | | | | | | | | | | | | | |

Рис. 2

Лабораторная работа 4

ЗНАКОМСТВО СО СПОСОБАМИ ОТЛАДКИ ПРОГРАММ В РЕДАКТОРЕ VBA

1. Создайте новый файл с поддержкой макросов.
2. Откройте редактор VBA и создайте поле для размещения кода, щелкнув дважды на Лист1 в окне проектов.
 - 2.1. Введите код процедуры My_debug_1

```
Public Sub My_debug_1()
    Dim wb As Workbook
```

```

Dim Fam As String, Name As String, Age As Byte
Dim Total As String

Set wb = Workbooks.Add
wb.SaveAs Filename:="debug_11"

Sheets("Лист1").Activate
Fam = InputBox("Введите фамилию")
Debug.Print Fam
ActiveSheet.Range("A2").Value = Fam

Name = InputBox("Введите имя")
Debug.Print Name
ActiveSheet.Range("B2").Value = Name

Age = InputBox("Введите возраст")
Debug.Print Age
ActiveSheet.Range("C2").Value = Age

Total = Fam & " " & Name & " " & Age
ActiveSheet.Range("A1").Value = Total
'ActiveSheet.Columns("A:C").AutoFit
End Sub

```

2.2. Дополните имя файла `debug_11`, указав полный путь к созданной Вами папке

2.3. Проанализируйте раздел процедуры с описанием переменных и ответьте на вопросы:

-переменные каких типов будут задействованы в программе и каковы их имена?

-почему переменной `wb` предшествует ключевое слово `Set`?

2.4. Выполните процедуру, выбрав команду `Run – Run Sub/UserForm` или нажав клавишу `F5`.

2.5. Убедитесь, что в Вашей папке находится рабочая книга `debug_11.xlsx`.

2.6. Предъявите результат преподавателю.

2.7. Удалите файл `debug_11.xlsx`.

2.8. Выполните процедуру `My_debug_1` в *пошаговом* режиме, выбрав команду `Debug – Step Into` и нажимая клавишу `F8`.

2.9. Убедитесь, что процедура выполнена и удалите файл `debug_11.xlsx`.

2.10. Выполните процедуру My_debug_1, установив предварительно *точки прерывания* перед вводом имени и формированием итоговой строки.

2.11. Убедитесь, что процедура выполнена и удалите файл debug_11.xlsx.

2.12. Выполните процедуру My_debug_1, открыв окно быстрого просмотра Immediate (View – Window Immediate) для просмотра значений переменных Fam, Name, Age и Total (для Total используйте выражение Debug.Print Total в процедуре).

2.13. Уберите апостроф перед последней строкой процедуры и объясните действие этой строки.

2.14. Сохраните созданный файл в своей папке и предъявите результат преподавателю.

3. Создайте в своей папке новый файл, поддерживающий макросы.

3.1. Разместите процедуру My_debug_2 в кодовом пространстве Листа2 и выполните процедуру в пошаговом режиме (F8).

```
Public Sub My_debug_2()
    Dim FinalRow As Long
    Dim i As Byte

    Sheets("Лист2").Activate
    ' Ввод чисел в ячейки A1:A3
    For i = 1 To 3
        Range("A" & i).Value = InputBox("Введите число")
    Next i

    Range("A1:A3").Select
    'Rows.Count - ЧИСЛО СТРОК В ДИАПАЗОНЕ, здесь совпадает с
    'номером последней строки, т.к. диапазон НАЧИНАЕТСЯ с A1
    FinalRow = Selection.Rows.Count

    Cells(FinalRow + 1, 1).FormulaR1C1 = "=Sum(R[-3]C:R[-1]C)"
    'форматирование - полужирный шрифт
    'Rows("1:3").Font.Bold = True
    'форматирование - красный фон
    'Range("A" & FinalRow + 1).Interior.Color = RGB(255, 0, 0)
End Sub
```

3.2. Проанализируйте полученный результат.

3.3. Удалите результаты с Листа2.

3.4. Замените объект Range в цикле For объектом Cells и выполните

процедуру, установив точку прерывания на строке Range("A1:A3").Select.

3.5. Предъявите результат преподавателю.

3.6. Удалите результаты с Листа2.

3.7. Используйте окно Watches (Debug – Add Watch) и поместите в него выражение Selection.Rows.Count и A4 и в пошаговом режиме проследите за отображением результатов работы процедуры.

3.8. Предъявите результат преподавателю.

3.9. Выполните процедуру, удалив апострофы перед строками с форматированием, и объясните действие этих строк.

3.10. Видоизмените процедуру таким образом, чтобы диапазон суммирования увеличился до пяти строк, выполните ее в обычном режиме (Run или F5) и предъявите результат преподавателю.

Лабораторная работа 5

РАБОТА С ТЕКСТОМ И ДАТАМИ

1. Создайте программный код, позволяющий вывести на лист книги таблицу кодов ASCII.

1.2. 255 символов и их порядковые номера в коде ASCII должны располагаться по 10 значений в каждом столбце. Фрагмент результата работы кода показан на рис. 3.

| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|---------------------------|-----------------|-----------------|--------|--------|---------|---------|---------|---------|---|---|---|
| 1 | 1 11 21 31 41) | 51 3 61 = | 71 G | 81 Q | 91 [| 101 e | 111 o | | | | | |
| 2 | 2 12 22 32 42 * | 52 4 62 > | 72 H | 82 R | 92 \ | 102 f | 112 p | | | | | |
| 3 | 3 13 23 33 ! | 43 + | 53 5 63 ? | 73 I | 83 S | 93] | 103 g | 113 q | | | | |
| 4 | 4 14 24 34 " | 44 , | 54 6 64 @ | 74 J | 84 T | 94 ^ | 104 h | 114 r | | | | |
| 5 | 5 15 25 35 # | 45 - | 55 7 65 A | 75 K | 85 U | 95 _ | 105 i | 115 s | | | | |
| 6 | 6 16 26 36 \$ | 46 . | 56 8 66 B | 76 L | 86 V | 96 ` | 106 j | 116 t | | | | |
| 7 | 7 17 27 37% | 47 / | 57 9 67 C | 77 M | 87 W | 97 a | 107 k | 117 u | | | | |
| 8 | 8 18 28 38 & | 48 0 | 58 : | 68 D | 78 N | 88 X | 98 b | 108 l | 118 v | | | |
| 9 | 9 19 29 39 ' | 49 1 | 59 ; | 69 E | 79 O | 89 Y | 99 c | 109 m | 119 w | | | |
| 10 | 20 30 40 (| 50 2 | 60 < | 70 F | 80 P | 90 Z | 100 c | 110 n | 120 x | | | |
| 11 | | | | | | | | | | | | |

Рис. 3

Используйте для формирования вывода счетчик по модулю 10 ($\text{mod } 10$). При $n \text{ mod } 10=0$, где n – порядковый номер символа, следует присвоить счетчику строк значение 1, а счетчик столбцов увеличить на 1 (т.е. перейти к следующему столбцу).

Используйте для формирования вывода функции работы со строками Str и Chr.

2. Предъявите результат преподавателю.

3. Создайте на листе Excel таблицу, приведенную на рис. 4.

| | A | B | C | D |
|---|-----------|-------|---|---|
| 1 | Марка | Цена | | |
| 2 | NokiaC7 | 18000 | | |
| 3 | NokiaC2 | 12000 | | |
| 4 | NokiaC4 | 20000 | | |
| 5 | MotorolaM | 16000 | | |
| 6 | SiemensC7 | 10000 | | |
| 7 | SiemensC9 | 14000 | | |
| 8 | NokiaC1 | 15000 | | |
| 9 | | | | |

Рис. 4

3.1. Замените марку NokiaNN (N – любой символ) на Nokia55, используя функции работы со строками Left и Mid, и вычислите количество произведенных замен.

Для отладки создайте копию таблицы в области рабочего листа.

4. Предъявите результат преподавателю.

5. В кодовом пространстве Лист1 новой книги Excel расположите процедуру date_1 и добавьте комментарии, поясняющие работу функций DateAdd, DatePart и DateDiff.

```

Public Sub date_1()
Dim dt As Date, dt1 As Date, dt2 As Date

    dt = Now
    dt1 = Time
    dt2 = Date
    Cells(1, 1).Value = dt
    Cells(1, 2).Value = dt1
    Cells(1, 3).Value = dt2
    Columns("a:c").AutoFit
    'добавьте комментарии к следующим строкам кода,
    'поясняющие результаты работы функций
    Cells(2, 1).Value = DateAdd("d", -3, #2/23/2011#)
    Cells(3, 1).Value = DateAdd("q", 3, Date)
    Cells(4, 1).Value = DateDiff("d", #1/25/2011#, Date)
    Cells(5, 1).Value = DatePart("w", Date, 2)
End Sub

```

5.1. В указанных функциях замените функцию Date произвольной датой и предъявите результат преподавателю.

5.2. Для произвольной даты, введенной в произвольную ячейку Листа2, определите название дня недели, соответствующего этой дате. Ис-

пользуйте функцию DatePart и оператор Select Case.

6. Предъявите результат преподавателю.

7. В кодовом пространстве Лист1 новой книги Excel расположите функцию vis и процедуру vis_proc, в которой происходит обращение к этой функции.

```
Function vis(year As String) As Boolean
    Dim d1 As String, d2 As String
    d1 = "1/1/" & year
    d2 = "12/31/" & year
    If DateDiff("d", d1, d2) = 365 Then
        vis = True
    Else
        vis = False
    End If
End Function
```

```
Public Sub vis_proc()
    Dim s As String
    s = InputBox("Ваш год рождения?")
    If vis(s) Then
        Sheets(3).Range("a10").Value = "високосный"
    Else
        Sheets(3).Range("a10").Value = "не високосный"
    End If
End Sub
```

7.1. Выполните процедуру и проанализируйте полученный результат.

7.2. Модифицируйте текст функции так, чтобы она возвращала текстовые значения «високосный» и «не високосный».

7.3. Модифицируйте текст процедуры, в которой происходит обращение к модифицированной функции.

8. Предъявите результат преподавателю.

Лабораторная работа 6

ЗНАКОМСТВО СО СВОЙСТВАМИ И МЕТОДАМИ ОБЪЕКТА RANGE

1. Заполните произвольными данными диапазон B5:D8 Лист1 новой книги.

1.1. Введите код процедуры col_row в кодовое пространство Лист1 и выполните процедуру, предварительно открыв окно Immediate.

```

Public Sub col_row()
    Dim N As Long, M As Long
    Dim N1 As Long, M1 As Long, n2 As Long
    Dim finalrow As Long, finalcol As Long
    Dim r As Range

    Range("B5:D8").ClearFormats

    ' N – номер ПЕРВОГО столбца диапазона
    ' M - номер ПЕРВОЙ строки диапазона
    N = Range("B5:D8").Column
    M = Range("B5:D8").Row
    Debug.Print N; M
    ' N1 – количество столбцов диапазона
    ' M1 - количество строк диапазона
    N1 = Range("B5:D8").Columns.Count
    M1 = Range("B5:D8").Rows.Count
    Debug.Print N1; M1

    ' обращение ко второй строке диапазона, например, для форматирования
    Range("B5:D8").Rows(2).Font.Color = RGB(255, 0, 0)

    ' обращение ко всем строкам диапазона, например, для форматирования
    Range("B5:D8").Rows.Font.Color = RGB(0, 0, 255)

    'finalrow, finalcol – номера последней строки и последнего столбца диапазона
    B5:D8
    finalrow = Cells(Rows.Count, 2).End(xlUp).Row
    finalcol = Cells(6, Columns.Count).End(xlToLeft).Column
    Debug.Print finalrow; finalcol
    End Sub

```

1.2. Запомните, ЧТО возвращают свойства **Column** и **Row** диапазона (объекта **Range**) и каковы их значения для заданного диапазона.

1.3. Запомните, ЧТО возвращают свойства **Columns.Count** и **Rows.Count** диапазона (объекта **Range**) и каковы их значения для заданного диапазона.

1.4. Запомните, как можно обратиться к конкретной строке (столбцу) диапазона и как обратиться ко всем строкам (столбцам) диапазона (объекта Range).

1.5. Как можно найти номер последней строки или столбца диапазона? Объясните строки кода, вычисляющие значения переменных finalrow и finalcol. Какому объекту принадлежат свойства Columns.Count и Rows.Count в вычислении значений этих переменных (Cells(Rows.Count,2) Cells(6, Columns.Count))?

2. Создайте таблицу с произвольными данными (рис. 5) и поместите ее на второй лист книги Excel.

2.1. Введите код процедуры range_work в кодовое пространство Листа2 и выполните процедуру. Обратите внимание на формирование ссылки на диапазон Set r1=Range("B10") и ее описание Dim r1 As Range.

```
Public Sub range_work()
    Dim r1 As Range, r2 As Range
    Cells.ClearFormats

    Set r1 = Range("B10")
    r1.Name = "Итоги"

    Names.Add Name:="Total", RefersTo:="=Лист1!$e$10"
    'Range("B3").AddComment "Текст комментария"

    ' расширение диапазона до трех столбцов и форматирование
    Set r2 = r1.Resize(, 3)
    r2.Clear
    r2.Interior.Color = vbBlue

    'использование ссылок R1C1 и занесение формулы в диапазон
    B10:D10 (variant 1)
    r2.FormulaR1C1 = "=sum(r[-7]c:r[-1]c)"

    'копирование формулы в диапазон C10:D10 (variant 2)
    'Range("Итоги").Value = "=sum(B3:B9)"
    'Range("Итоги").Copy Destination:=Range("C10:D10 ")
```

```

'копирование формулы в диапазон C10:D10 (variant 3)
'Range("Итоги").Select
'Selection.Copy
'Range("C10:D10").Select
'ActiveSheet.Paste
End Sub

```

| | A | B | C | D | E |
|----|----|-----|-----|-----|---|
| 1 | | | | | |
| 2 | | янв | фев | мар | |
| 3 | пн | 123 | 234 | 345 | |
| 4 | вт | 123 | 234 | 345 | |
| 5 | ср | 123 | 234 | 345 | |
| 6 | чт | 123 | 234 | 345 | |
| 7 | пт | 123 | 234 | 345 | |
| 8 | сб | 123 | 234 | 345 | |
| 9 | вс | 123 | 234 | 345 | |
| 10 | | | | | |

Рис. 5

2.2. Отметьте различие в присвоении имен диапазонам с помощью свойства r1.Name и метода Names.Add

2.3. Какое свойство диапазона позволяет изменить его размеры? Укажите второй вариант написания свойства диапазона r1.Resize(, 3).

2.4. Выполните три варианта заполнения формулами диапазона B10:D10.

2.5. Почему при использовании стиля ссылок R1C1 формула не нуждается в копировании?

2.6. Суммируйте значения в ячейках B10:D10 и занесите результат в клетку E10 (“Total”). Используйте стиль ссылок R1C1. Выделите ячейку форматом. Центрируйте значение (HorizontalAlignment=xlCenter).

3. Предъявите результаты преподавателю.

4. Создайте таблицу с произвольными данными (рис. 6) и поместите ее на третий лист книги Excel.

| | A | В | С | Д | Е | F | G | Н | I | J | К | L | M |
|---|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | | янв | фев | мар | апр | май | июн | июл | авг | сен | окт | ноя | дек |
| 2 | Продажа оборуд | 123 | 345 | 456 | 567 | 789 | 789 | 789 | 789 | 789 | 789 | 789 | 789 |
| 3 | Продажа По | 123 | 345 | 456 | 567 | 789 | 789 | 789 | 789 | 789 | 789 | 789 | 789 |
| 4 | Сервис | 123 | 345 | 456 | 567 | 789 | 789 | 789 | 789 | 789 | 789 | 789 | 789 |
| 5 | Торговые издержки | 123 | 345 | 456 | 567 | 789 | 789 | 789 | 789 | 789 | 789 | 789 | 789 |
| 6 | Общие расходы | 123 | 345 | 456 | 567 | 789 | 789 | 789 | 789 | 789 | 789 | 789 | 789 |
| 7 | | | | | | | | | | | | | |

Рис. 6

4.1. Введите код процедуры range_offset в кодовое пространство Лист3 и выполните процедуру.

```

Public Sub range_offset()
    Dim r As Range
    ' работа с выделенным диапазоном - Selection
    ' перед выполнением процедуры нужно ВЫДЕЛИТЬ таблицу
    ' четные строки «закрасить»
    ' вычислить среднее по строкам и суммы по столбцам
    Cells.ClearFormats

    Set r = Selection.Rows(2)

    Do While r.Cells(1, 1).Value <> ""
        r.Interior.ColorIndex = 4
        Set r = r.Offset(2, 0)
    Loop
End Sub

```

4.1. Обратите внимание на формирование ссылки на предварительно выделенный диапазон (объект Selection), в котором формируется ссылка на вторую строку Set r=Selection.Rows(2)

В цикле While новый диапазон формируется сдвигом предыдущей строки на две строки вниз Set r=r.Offset(2,0). Обратите внимание на условие продолжения цикла r.Cells(1,1).value<> «» и объясните его.

4.2. Вычислите среднее значение по строкам таблицы (функция VBA Average()) и суммы по столбцам таблицы, используя стиль ссылок R1C1.

5. Предъявите результат преподавателю.

6. Создайте заголовки столбцов таблицы в диапазоне A1:E1 и поместите их на свободный лист книги Excel так как показано ниже.

| Дата | Имя | Номер | Адрес | Тлф |
|------|-----|-------|-------|-----|
|------|-----|-------|-------|-----|

6.1. Создайте две процедуры add_rec1 и add_rec2, позволяющие *ввести данные* в таблицу с приведенными выше заголовками, в первую свободную строку таблицы.

6.2. Процедура add_rec1 должна использовать цикл, в котором параметром цикла будет номер (счетчик) столбцов (For c=1 To 5, где c – параметр цикла – счетчик столбцов).

6.3. Ввод данных должен происходить с помощью окна ввода InputBox.

6.4. Последняя заполненная строка диапазона должна быть вычислена с помощью выражения finalrow=Cells(Rows.Count,1).End(xlUp).Row

6.5. Созданная процедура должна быть использована для ввода трех строк в предлагаемую таблицу.

7. Предъявите результат преподавателю.

8. Процедура add_rec2 должна использовать ссылки на диапазоны – *заполненный* данными и *свободный* - для введения новых данных.

Заполненный диапазон определяется с помощью свойства CurrentRegion
Set r=ActiveSheet.Range("A1").CurrentRegion

Свободный диапазон определяется при смещении заполненного диапазона на количество строк в нем:

Set r1=r.Offset(r.Rows.Count,0)

8.1. Ввод данных должен происходить с помощью обозначения *позиции* ячейки в диапазоне.

r1.Cells(1).Value= #1/31/2010#

r1.Cells(2).Value=" Иванов" и т.д.

8.2. Созданная процедура должна быть использована для ввода одной строки в предлагаемую таблицу.

9. Предъявите результат преподавателю.

Лабораторная работа 7

ЗНАКОМСТВО С ИСПОЛЬЗОВАНИЕМ ПРОСТОЙ ФОРМЫ ДЛЯ ВВОДА ДАННЫХ В ТАБЛИЦУ

1. Создайте на листе1 книги Excel таблицу, приведенную на рис. 7.

| | A | B | C | D | E | F |
|---|----------------------|-----|-----|-----|-----|---|
| 1 | | янв | фев | мар | апр | |
| 2 | продажи оборудования | | | | | |
| 3 | продажи по | | | | | |
| 4 | сервис | | | | | |
| 5 | | | | | | |
| 6 | | | | | | |

Рис. 7

2. Создайте код процедуры, позволяющей ввести данные в ячейки таблицы B2:E4, пользуясь обращением к ячейкам таблицы Cells(строка, столбец) и диалоговым окном InputBox.

2.1. В клетку B5 запишите формулу суммирования по месяцам, пользуясь свойством Resize.

3. Создайте код процедуры, позволяющий ввести данные в ячейки таблицы B2:E4, пользуясь обращением к ячейкам таблицы Cells(pos) и диалоговым окном InputBox.

4. Создайте форму, приведенную на рис. 8.

4.1. Создайте код процедуры обработки события Click кнопки Ввод, позволяющей вводить данные в таблицу из полей формы, приведенной на рис. 8.

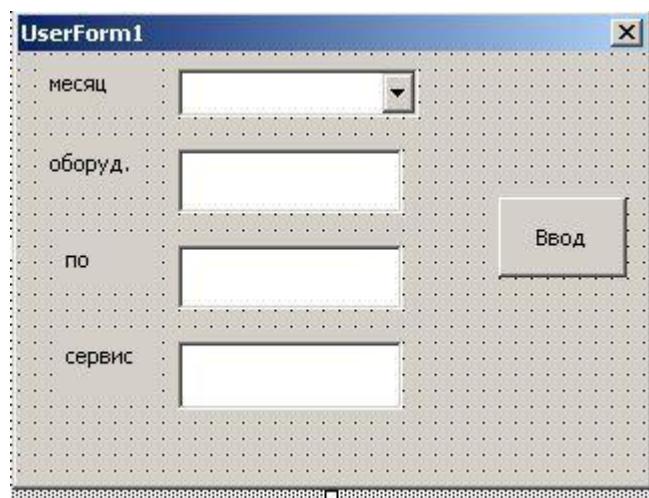


Рис. 8

5. Предъявите работу преподавателю.

Лабораторная работа 8

ПРОЕКТ СОЗДАНИЯ «БАЗЫ ДАННЫХ» СРЕДСТВАМИ VBA В EXCEL 2007 (1).

СОЗДАНИЕ СТРУКТУРЫ ТАБЛИЦЫ И ЭКРАННОЙ ФОРМЫ ДЛЯ ВЗАИМОДЕЙСТВИЯ С ТАБЛИЦЕЙ

1. Создайте структуру таблицы - введите в диапазон A1:E1 заголовки полей и сохраните книгу под именем Имя_project1.xlsx, где Имя – Ваше имя или фамилия:

Дата ФИО Индекс Район Платеж

Поле **Дата** содержит данные типа **Дата и время**.

Поле **Индекс** представляет собой шестизначную комбинацию цифр.

Поля **ФИО** и **Район** содержат данные типа **Текст**.

Поле **Платеж** содержит числовые данные.

Данные вводятся в таблицу программным путем с помощью формы!!!

Пример таблицы приведен на рис. 9.

| Дата | ФИО | Индекс | Район | Платеж |
|------------|----------------------------|--------|-------------|--------|
| 09.04.2010 | Пупкин Васисуалий Петрович | 123456 | Центральный | 5600 |

| | | | | |
|------------|------------------------------------|--------|------------------|-------|
| 05.05.1985 | Фудзиямова Киота Токиевна | 589789 | Василеостровский | 7000 |
| 30.04.1999 | Снегирев Андрей Геннадьевич | 112233 | Центральный | 10000 |
| 07.07.2007 | Шерстянкин Никанор Дормидон-тovich | 777777 | Петроградский | 25000 |
| 31.03.2011 | Никитина Нина Никитична | 190000 | Приморский | 15000 |
| 05.05.2010 | Зеленкин Пафнутий Михайлович | 876543 | Василеостровский | 8000 |

Рис. 9

2. Создайте экранную форму, приведенную на рис. 10.

Рис. 10

2.1. Экранная форма должна иметь заголовок **Добавление, просмотр и удаление записи**, иметь имя (свойство Name) - **UserForm1** и содержать следующие элементы управления.

2.1.1. Текстовые поля **TextBox** для ввода даты, ФИО, индекса и платежа. Текстовое поле для ввода ФИО должно предусматривать возможность ввода нескольких строк текста. (Свойство **Multiline** должно иметь значение **True**). Имена текстовых полей (свойство Name):**txtdate**, **txtfio**, **txtindex**, **txtpl**, соответственно.

2.1.2. Раскрывающийся список **ComboBox** для выбора района. Имя – **cmbdis**.

2.1.3. Командная кнопка **Отмена**, нажатие на которую закрывает форму. Имя кнопки – **cmdcancel**.

2.1.4. Командная кнопка **Далее**, нажатие на которую позволяет ввести данные в таблицу из формы и очистить поля формы для дальнейшего ввода данных. Кнопка **Далее** выбирается активной по умолчанию (свойство **Default** должно иметь значение **True**). Имя кнопки – **cmdnext**.

2.1.5. Командная кнопка Готово (OK), нажатие на которую позволяет ввести данные в таблицу и закрыть форму. Имя кнопки – cmddone.

2.1.6. Панели Frame, которые используются для создания секции Добавить данные для кнопок Далее и Готово (OK) и для создания секции Просмотр для просмотра записей таблицы в прямом (с первой записи до последней) и обратном (с последней записи до первой) порядке. Панели Frame используются для группирования элементов. **Для размещения элементов в секции, первоначально должна быть создана секция, а затем элементы, ее содержащие!**.

2.1.7. Командная кнопка Удалить, нажатие на которую позволяет удалить выбранную запись. Имя кнопки – cmddelete.

2.1.8. Полоса значений SpinButton используется для просмотра записей таблицы в прямом (с первой записи до последней) и обратном (с последней записи до первой) порядке. Имя элемента – SpinButton1.

3. Создайте код инициализации формы

Код инициализации формы выполняется при загрузке формы. Его задача загрузить список ComboBox (имя – cmbdis) с названиями районов города (достаточно ввести названия пяти – шести районов). Код инициализации содержится в процедуре обработки события Initialize для формы UserForm. Код помещается в кодовую область формы.

```
Private Sub UserForm_Initialize()
    'загружает список с названиями районов
    cmbdis.AddItem "Центральный"
    cmbdis.AddItem "Приморский"
    cmbdis.AddItem "Василеостровский"
    cmbdis.AddItem "Выборгский"
    cmbdis.AddItem "Фрунзенский"
    cmbdis.AddItem "Петроградский"
End Sub
```

4. Создайте процедуру очистки формы. Перед записью в форму новых данных она должна быть очищена. Процедура очистки полей формы приведена далее и состоит в записи «пустой» строки в поля формы.

```
Public Sub ClearForm()
    'удаляет данные из полей формы
    txtdate.Value = ""
    txtfio.Value = ""
    txtindex.Value = ""
    txtpl.Value = ""
    cmbdis.Value = ""
End Sub
```

5. Активизируйте форму с помощью кода UserForm1.Show. Макрос с этим кодом назначьте кнопке в Листе1. Проверьте работоспособность созданных процедур.

6. Предъявите результаты работы преподавателю.

Лабораторная работа 9

ПРОЕКТ СОЗДАНИЯ «БАЗЫ ДАННЫХ» СРЕДСТВАМИ VBA В EXCEL 2007 (2).

ДОБАВЛЕНИЕ ДАННЫХ В ТАБЛИЦУ С ПРОВЕРКОЙ ИХ КОРРЕКТНОСТИ

1. Создайте процедуру проверки корректности введения Индекса. Индекс должен состоять только из цифр и иметь длину шесть символов.

1.1. Для оценки данных, вводимых с клавиатуры, можно использовать событие KeyDown (нажатие на клавишу). При наступлении этого события возвращается значение аргумента KeyCode, соответствующее коду нажатой клавиши. В справочной системе VBA представлена таблица значений KeyCode, из которой следует, что клавишам 0 – 9 соответствуют коды 48 – 57. Таким образом, если процедура обработки события KeyDown получает для аргумента KeyCode значение в диапазоне 48 – 57, это свидетельствует о нажатии цифровой клавиши, в противном случае будет сформировано сообщение об ошибке. Для разрешения использования служебных клавиш (например, Backspace), которые имеют коды от 0 до 32, диапазон условий можно расширить:

(KeyCode > 32 And KeyCode < 48) Or KeyCode > 57

Код процедуры представлен далее.

```
Private Sub txtindex_KeyDown(ByVal KeyCode As MSForms.ReturnInteger, _
    ByVal Shift As Integer)
```

```
If (KeyCode > 32 And KeyCode < 48) Or KeyCode > 57 Then
```

```
    KeyCode = 0
```

```
    MsgBox "Жми цифру!"
```

```
End If
```

```
End Sub
```

2. Создайте код функции, проверяющей корректность вводимых данных.

После щелчка на кнопках Далее и Готово (OK) данные должны быть введены в таблицу только после их проверки на корректность. Это подразумевает проверку заполнения всех полей формы и проверку длины данных поля Индекс, составляющих шесть символов. Длина проверяется с помощью свойства текстового поля – TextLength. Код функции ValidateData() приведен далее.

```
Public Function ValidateData() As Boolean
```

```

If txtdate.Value = "" Then
    MsgBox "укажите дату"
    ValidateData = False
    Exit Function
End If

If txtfio.Value = "" Then
    MsgBox "укажите фио"
    ValidateData = False
    Exit Function
End If

If txtindex.Value = "" Then
    MsgBox "укажите индекс"
    ValidateData = False
    Exit Function
End If

If txtpl = "" Then
    MsgBox "укажите платеж"
    ValidateData = False
    Exit Function
End If

If cmbdis.Value = "" Then
    MsgBox "выберите район"
    ValidateData = False
    Exit Function
End If

If txtindex.TextLength <> 6 Then
    MsgBox "индекс должен состоять из шести цифр"
    ValidateData = False
    Exit Function
End If

ValidateData = True
End Function

```

3. Создайте процедуру ввода данных из полей формы в таблицу (добавление данных). В процедуре определяется заполненный данными диа-

пазон с помощью свойства CurrentRegion и смещение на количество строк, содержащихся в исходном диапазоне для нахождения свободной строки для записи данных. Для обращения к ячейкам используется свойство Cells(pos) с указанием позиции ячейки в диапазоне. Код процедуры приведен далее.

```
Public Sub EnterDataInWorksheet()

    Dim r As Range, r2 As Range

    Set r = Worksheets("Лист1").Range("A1").CurrentRegion
    Set r2 = r.Offset(r.Rows.Count, 0)

    r2.Cells(1).Value = txtdate.Value
    r2.Cells(2).Value = txtfio.Value
    r2.Cells(3).Value = txtindex.Value
    r2.Cells(4).Value = cmbdis.Value
    r2.Cells(5).Value = txtpl.Value
    Worksheets("Лист1").Columns("A:E").AutoFit

End Sub
```

4. Создайте процедуры, выполняющиеся при нажатии командных кнопок (процедуры обработки события Click этих кнопок).

4.1. Нажатие кнопки Отмена приводит к очистке формы (процедура ClearForm) и ее закрытию (Me.Hide). Процедуру Private Sub cmdcancel_Click() напишите самостоятельно.

4.2. Нажатие кнопки Далее приводит к выполнению процедур записи данных в таблицу EnterDataInWorksheet и очистки формы (ClearForm) при условии, что функция проверки корректности ввода данных (ValidateData) возвратит значение True. Процедуру Private Sub cmdnext_Click() напишите самостоятельно.

4.3. Нажатие кнопки Готово (OK) приводит к тем же действиям, что и при нажатии кнопки Далее, но кроме процедур ввода данных в таблицу и очистки формы, выполняется закрытие формы. Эти три действия происходят при условии, что функция проверки корректности ввода данных (ValidateData) возвратит значение True. Процедуру Private Sub cmddone_Click() напишите самостоятельно.

5. Проверьте работоспособность процедур и предъявите результат преподавателю.

Лабораторная работа 10

ПРОЕКТ СОЗДАНИЯ «БАЗЫ ДАННЫХ» СРЕДСТВАМИ VBA В EXCEL 2007 (3). ПРОСМОТР И УДАЛЕНИЕ ЗАПИСЕЙ

1. Создайте процедуры просмотра записей таблицы.

1.1. Процедуры просмотра формируются как процедуры обработки событий SpinDown и SpinUp элемента управления SpinButton. События возникают при нажатии на изображения стрелок элемента управления. Каждое нажатие должно приводить к просмотру одной записи или в порядке возрастания (от первой до последней), или в порядке убывания (от последней к первой). Поэтому обращение к процедурам должно фиксироваться с помощью ГЛОБАЛЬНОЙ переменной (в проекте – i), которая должна быть описана вне всех процедур - Public i As Long.

Переменная r1, которая ссылается на просматриваемую запись и должна быть доступна для процедур просмотра и удаления записей, также должна быть объявлена как ГЛОБАЛЬНАЯ, и ее объявление вынесено вне области процедур - Public r1 As Range.

В каждой из процедур необходимо вначале определить номер последней строки finalRow и заполненный диапазон с помощью свойства CurrentRegion:

```
finalRow = Range("A1").Cells(Rows.Count, 1).End(xlUp).Row  
Set r = Worksheets("Лист1").Range("A1").CurrentRegion
```

1.2. В процедуре SpinButton1_SpinDown, в которой просмотр происходит от первой до последней записи, в качестве исходной строки берется первая строка диапазона (т.е. строка с заголовками полей). При каждом обращении к этой процедуре диапазон (строка) смещается вниз - Set r1 = r1.Offset(i) на значение переменной i. Таким образом, при первом обращении смещение исходной строки активизирует вторую строку, при втором – третью и так до достижения переменной i значения finalRow. При достижении значения finalRow переменная i обнуляется. Код процедуры приведен далее.

```
Private Sub SpinButton1_SpinDown()  
    Dim r As Range  
    Dim finalRow As Long  
    ClearForm  
    finalRow = Range("A1").Cells(Rows.Count, 1).End(xlUp).Row  
    Set r = Worksheets("Лист1").Range("A1").CurrentRegion  
  
    Set r1 = r.Rows(1)
```

```

'i – счетчик обращений к процедуре
i = i + 1
If i < finalRow Then
    Set r1 = r1.Offset(i)
    txtdate.Value = r1.Cells(1).Value
    txtfio.Value = r1.Cells(2).Value
    txtindex.Value = r1.Cells(3).Value
    cmbdis.Value = r1.Cells(4).Value
    txtpl.Value = r1.Cells(5).Value
Else
    i = 0
End If
End Sub

```

1.3. В процедуре `SpinButton1_SpinUp`, где просмотр происходит от последней до первой записи, в качестве исходной строки берется строка, следующая за строкой `finalRow` (т.е. строка за диапазоном). При каждом обращении к этой процедуре диапазон (строка) смещается вверх - `Set r1 = r1.Offset(-i)` на значение переменной `i`. Таким образом, при первом обращении смещение исходной строки активизирует строку `finalRow`, при втором – следующую и так до достижения переменной `i` значения `finalRow`. При достижении значения `finalRow` переменная `i` обнуляется. Код процедуры приведен далее.

```

Private Sub SpinButton1_SpinUp()
    Dim r As Range
    Dim finalRow As Long
    ClearForm
    finalRow = Range("A1").Cells(Rows.Count, 1).End(xlUp).Row
    Set r = Worksheets("Лист1").Range("A1").CurrentRegion
    Set r1 = r.Rows(finalRow + 1)
    'i – счетчик обращений к процедуре
    i = i + 1
    If i < finalRow Then
        Set r1 = r1.Offset(-i)
        txtdate.Value = r1.Cells(1).Value
        txtfio.Value = r1.Cells(2).Value
        txtindex.Value = r1.Cells(3).Value
        cmbdis.Value = r1.Cells(4).Value
        txtpl.Value = r1.Cells(5).Value
    Else
        i = 0
    End If

```

```
End Sub
```

2. Создайте процедуру удаления записи.

Процедура удаления записи выполняется при щелчке на кнопке Удалить и должна иметь доступ к просматриваемым записям. Поэтому переменная r1, которая ссылается на просматриваемую запись, должна быть объявлена как ГЛОБАЛЬНАЯ, и ее объявление вынесено вне области процедур - Public r1 As Range. Код процедуры приведен далее.

```
Private Sub cmddelete_Click()
    r1.EntireRow.Delete
    ClearForm
End Sub
```

3. Проверьте работоспособность созданных процедур и всей «базы данных» в целом.

4. Предъявите результаты преподавателю.

Лабораторная работа 11

ПРОЕКТ СОЗДАНИЯ «БАЗЫ ДАННЫХ» СРЕДСТВАМИ VBA В EXCEL 2007 (4).

МОДИФИКАЦИЯ ФОРМЫ. ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ ДАТЫ И ВРЕМЕНИ

1. Модифицируйте форму, созданную в Лабораторной работе 8 так как показано на рис. 11.

1.1. Удалите текстовое поле Дата (имя - txtdate) и создайте три новых текстовых поля для отображения составляющих даты: дня, месяца и года с именами txtday, txtmonth и txtyear.

1.2. Добавьте три элемента управления SpinButton (полоса значений) с именами sbday, sbmonth и sbyear. Для обеспечения их функционирования задайте свойства Min и Max в соответствии со значениями, которые может принимать день, месяц и год. (Для дня – от 1 до 30).

1.3. Удалите секцию Просмотр и создайте две новые кнопки Вперед и Назад с именами cmdforward и cmdback.

2. Внесите изменения в коды формы.

2.1. Процедуры просмотра записей используйте как процедуры обработки события Click кнопок Вперед и Назад. (Имена кнопок - cmdforward и cmdback), соответственно.

2.2. В процедурах просмотра записей занесение даты в текстовое поле txtdate замените кодом выделения из даты ее составляющих – дня, месяца и года и записью этих составляющих в соответствующие поля: txtday = DatePart("d", r1.Cells(1).Value)

```
txtmonth = DatePart("m", r1.Cells(1).Value)
txtyear = DatePart("yyyy", r1.Cells(1).Value)
```



Рис. 11

2.3. Замените код, относящийся к текстовому полю txtdate. В процедуре очистки полей формы вместо строки `txtdate.value=""` предусмотрите очистку трех текстовых полей txtday, txtmonth и txtyear, проверив предварительно, что эти имена (свойство Name) заданы соответствующим текстовым полям.

2.4. В функции ValidateData замените проверку заполнения текстового поля txtdate на проверку заполнения трех текстовых полей txtday, txtmonth и txtyear.

2.5. В процедурах ввода данных в таблицу и изменения (редактирования) данных предусмотрите возможность объединения составляющих даты в порядке: месяц, день, год, которые должны быть соединены с помощью операции & и отделены друг от друга с помощью знака /. `txtmonth.Value & "/" & txtday.Value & "/" & txtyear.Value`.

2.6. Заполните таблицу. Число записей должно быть не менее десяти. Скопируйте таблицу на второй лист в качестве резервной копии

2.7. Проверьте работоспособность всех элементов управления формы

3. Предъявите результат работы преподавателю.

Лабораторная работа 12

ЗНАКОМСТВО С МЕТОДАМИ ОБРАБОТКИ ДАННЫХ С ПОМОЩЬЮ СОРТИРОВКИ

1. Отсортируйте табличные данные, созданные в проекте, приведенном в предыдущих лабораторных работах.
 - 1.1. Создайте процедуру sort1, в которой отсортируйте табличные данные по полю ФИО в алфавитном порядке.
 - 1.2. Создайте процедуру sort2, в которой отсортируйте табличные данные по полю ФИО в обратном алфавитном порядке.
 - 1.3. Создайте процедуру sort3, в которой отсортируйте табличные данные в обратном алфавитном порядке по полю Район и затем по возрастанию платежей.
 - 1.4. Выполните пункт 1.1.
2. Предъявите результат работы преподавателю.
3. Создайте форму, приведенную на рис. 12, предназначенную для вывода результатов сортировки.

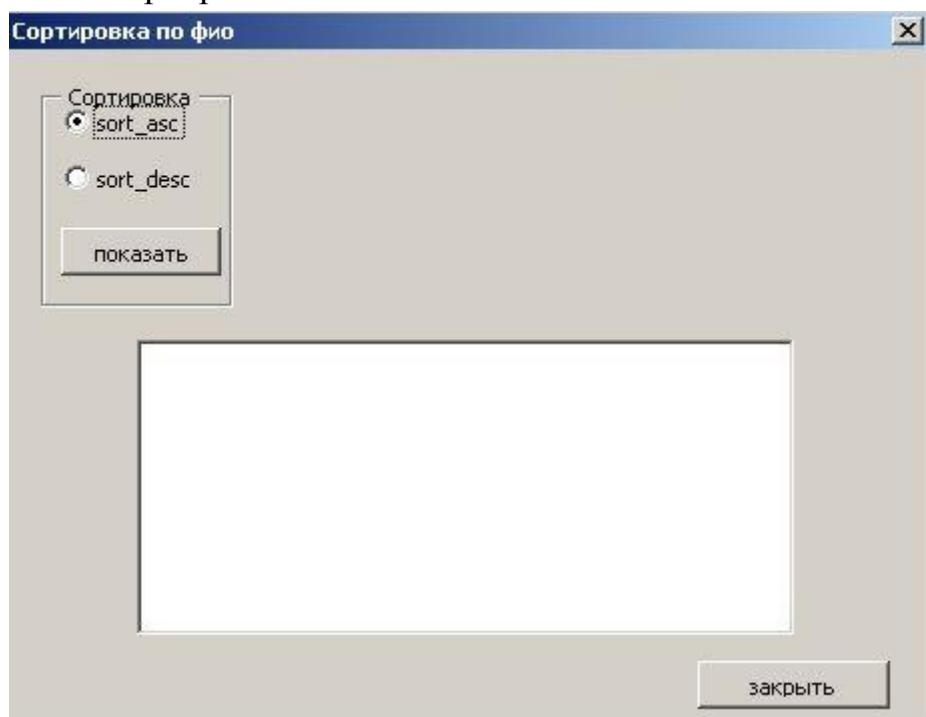


Рис. 12

3.1. Вывод отсортированной таблицы производится в элемент управления - список **ListBox**, имеющий имя **ListBox1**.

3.1.1. В этот элемент управления можно вывести произвольное количество полей таблицы, управляя их шириной. Свойство **ColumnWidths**, определяющее ширину полей в пунктах, может быть установлено в панели

свойств элемента в редакторе VBA. Установите значение этого свойства следующим образом: 0,120,0,0,0. После получения результата вывода установите ширину поля Платеж, отличную от нуля.

3.1.2. В свойствах элемента управления ListBox установите также значения свойств ColumnCount, равное 5 и ColumnHeads, равное True.

3.1.3. Свойство RowSource, определяющее диапазон вывода, укажите в кодах процедур сортировки :ListBox1.RowSource = r.Address, где r – диапазон сортируемой таблицы, а свойство Address возвращает адрес этого диапазона, например a22:e44, если таблица находится в этом диапазоне.

3.1.4. Вывод результатов сортировки в ListBox должен происходить с помощью процедуры обработки события Click кнопки Показать. Процедура анализирует состояние переключателей (элементы управления OptionButton) и при нажатии пользователем соответствующего переключателя (при этом переключатель возвращает значение True - OptionButton1.value=True) выполняются соответствующие процедуры сортировки.

4. Проверьте работоспособность процедур формы.

5. Предъявите результат преподавателю.

Лабораторная работа 13

ЗНАКОМСТВО С МЕТОДАМИ ОБРАБОТКИ ДАННЫХ С ПОМОЩЬЮ ФИЛЬТРАЦИИ.

ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ ДАННЫХ

1. Отфильтруйте табличные данные, созданные в проекте, приведенном в предыдущих лабораторных работах.

1.1. С помощью автофильтра найдите платежи Центрального и Приморского районов.

1.2. С помощью автофильтра найдите данные о трех максимальных платежах.

1.3. С помощью автофильтра найдите записи, относящиеся к фамилиям, начинающимся с букв от Г до С.

1.4. С помощью расширенного фильтра найдите список уникальных (неповторяющихся районов) и посчитайте их количество.

1.5. С помощью расширенного фильтра найдите записи, относящиеся к фамилиям, начинающимся с букв С или П.

2. Создайте форму, приведенную на рис. 13, предназначенную для вывода результатов расширенной фильтрации и отображения результатов фильтрации графически.

2.1. Имена элементов управления формы: frm2 – форма, OptionButton1 и OptionButton2 – переключатели, CheckBox1 – флажок, ListBox1 – список, cmdclose_frm2 – кнопка Закрыть, cmdfil – кнопка Показать результаты фильтрации, cmdch – кнопка Показать диаграмму.

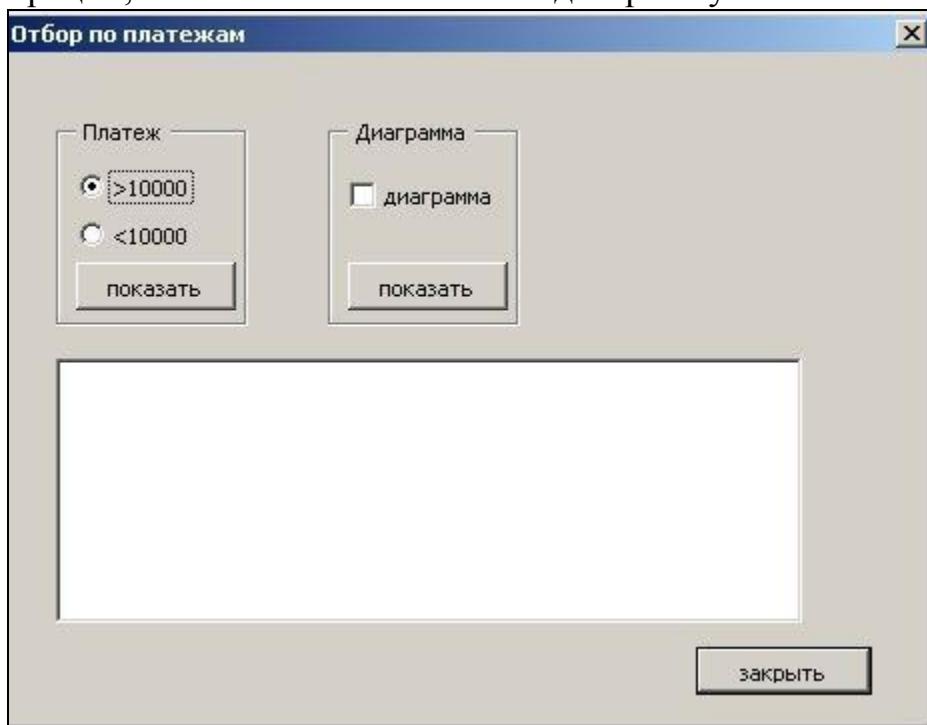


Рис. 13

3. Расширенная фильтрация должна обеспечить вывод записей таблицы по критериям поля Платежи.

3.1. Первый критерий расположите в области i1:i2. Он предназначен для отбора записей, в которых платежи ≥ 10000 , второй (платежи < 10000) расположите в области j1:j2.

3.2. Внесите код процедур ad_filter_1 и ad_filter_2 в кодовую область формы. (Отличие процедур только в определении диапазона критериев Crange).

```
Public orange As Range  
Public orange1 As Range  
Public i As Byte
```

```
Private Sub ad_filter_1()  
Dim Irange As Range, Crange As Range  
Dim finalRow As Long
```

```

i = i + 1
If i > 1 Then
orange1.Clear
CheckBox1.Value = False
End If

finalRow = Cells(Rows.Count, 1).End(xlUp).row
'определение исходного диапазона
Set lrange = Worksheets("Лист1").Range("A1:e" & finalRow)
'определение верхней левой ячейки выходного диапазона
Set orange = Range("A" & finalRow + 2)
'определение диапазона критериев
Set Crange = Range("i1:i2")
'фильтрация
lrange.AdvancedFilter Action:=xlFilterCopy, CopyToRange:=orange,
CriteriaRange:=Crange
'определение диапазона результатов (полного)
Set orange1 = orange.CurrentRegion
' отображение результата фильтрации в ListBox1
With ListBox1
.ColumnCount = 5
.ColumnWidths = "0,120,0,50,20"
.ColumnHeads = False
.RowStyle = orange1.Address
End With
End Sub

```

В процедуре ведется подсчет числа обращений к ней. При всех обращениях к процедуре, кроме первого, необходимо очистить область вывода результатов. Счетчик обращений обозначен глобальной переменной *i*, которая описана вне всех процедур.

Таким же образом описаны диапазоны вывода *orange* (верхняя левая ячейка диапазона) и диапазон отфильтрованных данных *orange1*, который формируется с помощью свойства *CurrentRegion* относительно диапазона (ячейки) *orange*.

Вывод результата фильтрации происходит в элемент управления *ListBox*, все свойства которого заданы в процедуре в отличие от кода, примененного в Лабораторной работе 12, где часть свойств заданы в панели

свойств элемента **ListBox**. Выбор способа задания свойств остается за программистом.

3.3 Внесите в кодовую область формы код процедуры обработки щелчка мыши на кнопке Показать секции Платежи.

```
Private Sub cmdfil_Click()
If OptionButton1.Value = True Then
    Worksheets("Лист1").Activate
    ad_filter_1
ElseIf OptionButton2.Value = True Then
    Worksheets("Лист1").Activate
    ad_filter_2
End If
End Sub
```

Выполнение той или иной процедуры фильтрации зависит от нажатия пользователем соответствующего переключателя. Активизация рабочего листа с таблицей необходима для повторения фильтрации после построения диаграммы, когда активным рабочим листом становится лист диаграммы.

3.4. Внесите в кодовую область формы код процедуры кнопки Закрыть и проанализируйте его самостоятельно.

```
Private Sub cmdclose_frm2_Click()
ActiveWorkbook.Save
Worksheets("Лист1").Activate
orange1.Clear
CheckBox1.Value = False
Me.Hide
End Sub
```

3.5. Внесите в кодовую область формы код процедуры построения диаграммы **Chart_1**.

```
Private Sub Chart_1()
Dim ch As Chart
Dim src As Range

orange1.Copy Destination:=Worksheets("Лист1").Range("aa40")
Set src = Worksheets("Лист1").Range("aa40").CurrentRegion

Set ch = ActiveWorkbook.Charts.Add
```

```

ch.SetSourceData Source:=src, PlotBy:=xlColumns
ch.ChartType = xlColumnClustered 'xlBar xlPie xlColumn xlPyramid
xlLineMarkers
ch.HasTitle = True
ch.ChartTitle.Caption = "платежи"
ch.SetElement msoElementDataLabelCenter
ThisWorkbook.Save
End Sub

```

Для построения диаграммы используется диапазон отфильтрованных данных, который копируется в свободную область для сохранения построенных диаграмм, поскольку диапазон orange1 очищается для последующих фильтраций. Проанализируйте код построения диаграммы.

3.6. Внесите в кодовую область формы код процедуры кнопки Показать диаграммы и проанализируйте его самостоятельно.

```

Private Sub Cmdch_Click()
If CheckBox1.Value = True Then
    Chart_1
    ThisWorkbook.Save
End If
End Sub

```

3.7. Внесите в кодовую область формы код процедуры инициализации формы и проанализируйте его самостоятельно.

```

Private Sub UserForm_Initialize()
ListBox1.Clear
OptionButton1.Value = True
OptionButton2.Value = False
CheckBox1.Value = False
End Sub

```

4. Проверьте работоспособность кодов.

5. Предъявите результат преподавателю.

Лабораторная работа 14

ЗНАКОМСТВО С ПРОЦЕДУРАМИ С ПЕРЕДАЧЕЙ ПАРАМЕТРОВ

1. Внесите изменения в коды процедур `ad_filter_1()` и `cmdfil_Click()`, приведенных в Лабораторной работе 13 для реализации процедуры фильтрации с помощью передачи параметра.

1.1. Измените имя процедуры на `ad_filter()`. Значение параметра `Crange` (диапазона критериев) должно быть передано в процедуру `ad_filter()`. Заголовок этой процедуры в измененном виде показан далее

```
Private Sub ad_filter(ByVal Crange As Range)
```

Значение `Crange` представляет собой формальный параметр процедуры `ad_filter()`. Фактическое значение этого параметра будет передано из процедуры `cmdfil_Click()`.

1.2. Удалите фрагмент процедуры, 'определение диапазона критериев
`Set Crange = Range("i1:i2")`

1.3. Код измененной процедуры `ad_filter()` представлен далее.

```
Private Sub ad_filter (ByVal Crange As Range)
Dim lrange As Range
Dim finalRow As Long

i = i + 1
If i > 1 Then
    orange1.Clear
    CheckBox1.Value = False
End If

finalRow = Cells(Rows.Count, 1).End(xlUp).row
'определение исходного диапазона
Set lrange = Worksheets("Лист1").Range("A1:e" & finalRow)
'определение верхней левой ячейки выходного диапазона
Set orange = Range("A" & finalRow + 2)
'фильтрация
lrange.AdvancedFilter Action:=xlFilterCopy, CopyToRange:=orange,
CriteriaRange:=Crange
'определение диапазона результатов (полного)
Set orange1 = orange.CurrentRegion
' отображение результата фильтрации в ListBox1
With ListBox1
    .ColumnCount = 5
    .ColumnWidths = "0,120,0,50,20"
    .ColumnHeads = False
    .RowSource = orange1.Address
```

```
End With
```

```
End Sub
```

1.4. Фактическое значение параметра Crange будет передано в процедуру ad_filter() из процедуры cmdfil_Click(), текст которой показан далее.

```
Private Sub cmdfil_Click()
If OptionButton1.Value = True Then
    Worksheets("Лист1").Activate
    Set rr = Range("i1:i2")
    ad_filter (rr)
ElseIf OptionButton2.Value = True Then
    Worksheets("Лист1").Activate
    Set rr = Range("j1:j2")
    ad_filter (rr)
End If
End Sub
```

Фактический параметр (rr) имеет в первом случае значение Range("i1:i2"), а во - втором - Range("j1:j2").

Обратите внимание на использование одной и той же процедуры ad_filter() для разных диапазонов критериев.

1.5. Удалите процедуру ad_filter_2().

1.6. Проверьте работоспособность пользовательской формы (рис. 13) с измененными процедурами.

2. Предъявите результаты преподавателю.

СПИСОК ЛИТЕРАТУРЫ

1. Истомин, Е. П.. Информатика и программирование : Pascal и VBA. учебник для вузов : / [ред. Л. С. Слесарева] ; Рос. гос. гидрометеорологический ун-т : СПб. : Андреевский издательский дом, 2010. - 293 с.

2. Стригина, Е. В. Программирование IT сервисов предприятия. : учебное пособие. / Е. В. Стригина. – СПб.: Издательство СПбГУТ, 2013. - 68 с.

3. Стригина, Е. В. Программирование : конспект лекций : [Электронный ресурс] / Е. В. Стригина. – 2012. - <http://www.fem-sut.spb.ru>.

4. Стригина, Е. В., Средства обработки и анализа данных.: методические указания к выполнению лабораторных работ. / Е. В. Стригина. – СПб.: Издательство СПбГУТ, 2012. - 51 с.

5. Вольфсон, М. Б. , Средства обработки и хранения данных. : учебное пособие. / Вольфсон М. Б. ,Стригина, Е. В. . – СПб.: Издательство СПбГУТ, 2012.- 54 с.

6. Слепцова, Л. Д. Программирование на VBA в Microsoft Office 2007 / Л. Д. Слепцова. – М. : ООО «Диалектика, Вильямс», 2007. – 688 с.

7. Джелен, Б. VBA и макросы в Microsoft Office Excel 2007 / Б. Джелен, Т. Сирстал ; пер.с англ. –М. : ООО «И.Д. Вильямс», 2008. –212 с.
8. Информатика: учебник / Б. В. Соболь, А. Б. Галин, Ю. В. Панов и др. – Ростов н/Д : Феникс, 2006. – 446 с.
9. Уокенбах, Дж. Microsoft Office Excel 2007 : Библия пользователя / Дж. Уокенбах. – М. : ИД «Вильямс», 2009. – 816 с.

Стригина Елена Владимировна

ПРОГРАММИРОВАНИЕ ИТ сервисов предприятия

**Методические указания
к выполнению лабораторных работ**

Редактор Л. А. Медведева

План 2013 г., п. 122
Подписано к печати 27.03.2013
Объем 3 усл.-печ. л. Тираж 25 экз. Заказ 300

РИЦ СПбГУТ. 191186 СПб., наб. р. Мойки, 61
Отпечатано в СПбГУТ

Е. В. Стригина

***Программирование
IT сервисов предприятия***

**Методические указания
к выполнению лабораторных работ**

**САНКТ-ПЕТЕРБУРГ
2013**