

Министерство труда и социальной защиты
Республики Беларусь

Государственное учреждение образования
«Республиканский институт повышения квалификации
и переподготовки работников Министерства труда
и социальной защиты Республики Беларусь»

И. В. ВИНОГРАДОВА, С. С. ЩУПАК

ПРИМЕНЕНИЕ МАКРОСОВ ДЛЯ АВТОМАТИЗАЦИИ ПРИЛОЖЕНИЙ ПОЛЬЗОВАТЕЛЕЙ ПК

Практикум
для слушателей переподготовки

Минск
«Минский государственный ПТК полиграфии»
2016

УДК 378.147.091.313(075.9)

ББК 74.58

В49

Рекомендовано к изданию Советом Государственного учреждения образования «Республиканский институт повышения квалификации и переподготовки работников Министерства труда и социальной защиты Республики Беларусь»

Р е ц е н з е н т ы :

старший преподаватель кафедры «Проектирование дорог» факультета
транспортных коммуникаций Белорусского национального
технического университета *П. В. Нарыжнов*;
к.т.н., доцент кафедры защиты информации Учреждения образования
«Белорусский государственный университет информатики и
радиоэлектроники» *С. Н. Петров*

Практикум содержит задания по созданию макросов и программ на языке программирования Visual Basic for Application.

Предназначен для слушателей повышения квалификации и переподготовки РИПК Минтруда и соцзащиты при проведении занятий по дисциплине «Средства визуального программирования приложений» специальности «Программное обеспечение информационных систем».

ISBN 978-985-6712-67-1

© Виноградова И. В., Шупак С. С., 2016

© РИПК Минтруда и соцзащиты, 2016

© Оформление. УО «Минский государственный
ПТК полиграфии, 2016

СОДЕРЖАНИЕ

Введение.....	3
Раздел 1. Создание макросов.....	5
Раздел 2. Создание простейших пользовательских функций.....	9
Раздел 3. Создание процедур-подпрограмм.....	12
Раздел 4. Условные операторы. Встроенная процедура MsgBox	20
Раздел 5. Операторы циклов	23
Раздел 6. Массивы.....	26
Раздел 7. Тестирование приложения пользователя.....	31
Раздел 8. Создание пользовательской формы.....	33

ВВЕДЕНИЕ

Многие задачи, выполняемые в Microsoft Office, такие как введение данных о продажах за определенный день или добавление формулы, выполняются только один раз. Некоторые задачи приходится выполнять довольно регулярно, совершая при этом много однообразных действий. Чтобы не выполнять ряд действий вручную, можно записать их с помощью макроса.

Макрос – действие или набор действий, используемых для автоматизации выполнения задач, требующих периодического выполнения.

При записи макроса сохраняется информация о каждом шаге выполнения последовательности команд. Последующий запуск макроса вызывает повторение команд. Visual Basic хранит каждый записанный макрос в отдельном модуле.

Модуль – совокупность описаний, инструкций и процедур, сохраненная под общим именем.

Язык программирования Visual Basic для приложений (VBA) был разработан специалистами корпорации Майкрософт для создания приложений на базе Microsoft Windows. Средствами языка программирования Microsoft Visual Basic можно написать пользовательский макрос.

Редактор Microsoft Visual Basic – среда, позволяющая редактировать записанные макросы, а также создавать новые макросы и программы на языке Visual Basic для приложений.

Редактор Visual Basic разработан для упрощения написания и редактирования программ начинающими пользователями. Для внесения простейших изменений в макросы нет необходимости изучать программирование на языке Visual Basic. Редактор Visual Basic позволяет редактировать макросы, копировать макросы из одного модуля в другой, переименовывать модули, хранящие макросы и переименовывать макросы.

Раздел 1. СОЗДАНИЕ МАКРОСОВ

Для автоматического создания *макросов* служит *макрорекордер*. С помощью *макрорекордера* можно записывать действия, которые выполняются в программе. Например, записывается ввод и удаление текста, нажатие на кнопки вкладок, форматирование текста и так далее.

Алгоритм работы с *макрорекордером* выглядит так:

1. Включить запись *макроса*, настроив некоторые параметры.
2. Выполнить действия, которые надо автоматизировать.
3. Остановить запись.
4. Использовать записанный макрос по необходимости.

Макрорекордер, помимо написания *макросов*, помогает лучше изучить возможности VBA и тонкости объектных моделей приложений Office. Например, чтобы узнать, как именно реализовать то или иное действие программно, надо записать макрос с нужными действиями, а потом открыть его в редакторе VBA. Далее проанализировать полученный код, а, возможно, и включить его фрагменты в свое приложение.

Задача 1. Работа с макросами в MS Word:

- записать макрос для перестановки букв;
- назначить сочетание клавиш для запуска макроса;
- создать кнопку для запуска макроса.

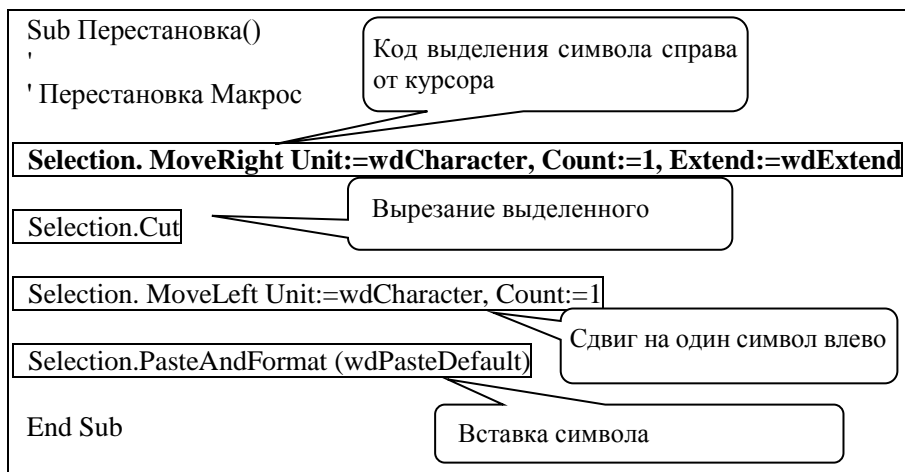
Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Word.
2. Создайте текстовый документ, содержащий текст:
Здесь находится текст для эксперимента по записи макросов.
3. Сохраните документ в своей рабочей папке с именем **Пример1**.
4. Создайте макрос **Перестановка** в документе **Пример1**.
5. Назначьте макросу сочетание клавиш **ALT+CTRL+W** в документе

Пример1.

6. Выполните запись макроса:
 - выделите символ сочетанием клавиш **Shift + →**;
 - щелкните по кнопке **Вырезать**;
 - измените положение курсора кнопкой **←**;
 - щелкните по кнопке **Вставить**;
 - щелкните по кнопке **Остановить запись**.
7. Проверьте работу макроса, нажав одновременно три клавиши **ALT+CTRL+W**.
8. Сохраните измененный документ **Пример1**.
9. Создайте на панели **Быстрого доступа** кнопку для запуска макроса **Перестановка** и сохраните ее в документе **Пример1**.
10. Проверьте работу кнопки.

11. Просмотрите VBA-код макроса **Перестановка**:



12. Закройте окно кода.
13. Сохраните изменения.
14. Завершите работу Microsoft Word.

Задача 2. Работа с макросами в MS Excel:

- создать макрос форматирования ячейки (денежный формат, цвет зеленый, шрифт – полужирный);
- назначить макросу сочетание клавиш;
- создать кнопку для вызова макроса.

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Excel.
2. Создайте таблицу следующего вида:

Валюта	Количество	Курс	Итого
USD			
EUR			
RUB			

3. Сохраните рабочую книгу с именем **Пример2**.
4. Введите данные в таблицу и выполните расчеты.
5. Активизируйте ячейку с курсом USD.
6. Создайте макрос с именем **Форматирование**.
7. Макрос должен быть сохранен в книге **Пример2**.

8. Назначьте макросу сочетание клавиш **CTRL+R** в рабочей книге

Пример2.

9. Выполните запись макроса:

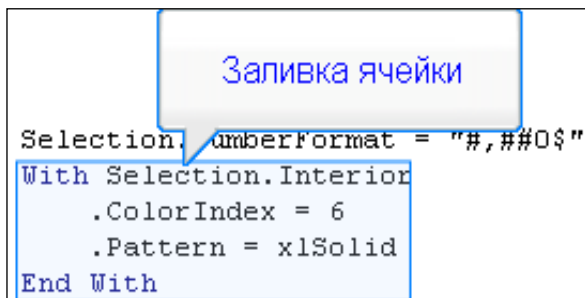
- установите **денежный формат**;
- установите число десятичных знаков **0**;
- выберите цвет для шрифта **Зеленый**;
- начертание шрифта – **полужирный**.

15. Проверьте работу макроса для ячеек с курсом EUR, нажав клавиши **CTRL+ R**.

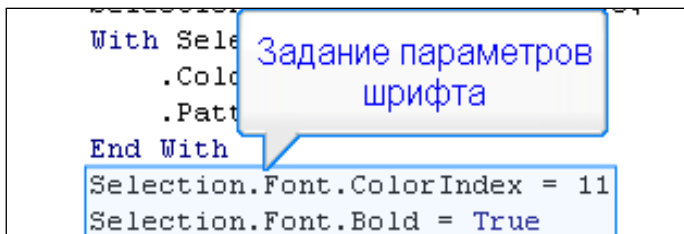
10. Создайте на панели **Быстрого доступа** кнопку для запуска макроса **Форматирование** и сохраните ее в рабочей книге **Пример2**.

11. Проверьте работу кнопки для ячеек с курсом RUB.

12. Просмотрите VBA-код макроса:



```
Selection.NumberFormat = "#,##0$"  
With Selection.Interior  
    .ColorIndex = 6  
    .Pattern = xlSolid  
End With
```



```
With Selection  
    .ColorIndex = 11  
    .Bold = True  
End With
```

13. Вернитесь в MS Excel.

14. Сохраните изменения.

15. Завершите работу MS Excel.

Задача 3. Пример использования макросов:

■ Создать макрос для оформления текста в таблицу в документе **MS Word**.

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Word.

2. Создайте документ следующего вида:

Поскольку свойство **Documents** глобальное, его можно использовать без указания **Application**, а так как **Item** – свойство или метод по умолчанию для объектов-наборов, его можно не упоминать в тексте программы.

Следовательно, оператор можно записать короче:

Documents(1).Words(1).Text = "The Bat"

Аналогично, в Microsoft Excel нужно пройти весь путь до объекта **Range**, который представляет ячейку **B3** рабочего листа **New** из рабочей книги **Sales.xls**:

Workbooks("Sales.xls").Worksheets("New").Range("B3").Value = 7

3. Сохраните документ с именем **ОБРАЗЕЦ**.

4. Выделите фрагмент текста:

Documents(1).Words(1).Text = "The Bat"

Замечание. Макрос будет работать с предварительно выделенным текстом.

5. Создайте макрос с именем **Оформление Кода** в документе **ОБРАЗЕЦ**.

6. Назначьте макросу комбинацию клавиш **ALT+CTRL+W** и сохраните ее в документе **ОБРАЗЕЦ**.

7. Выполните запись макроса:

➤ щелкните по кнопке **Вырезать**;

➤ создайте таблицу:

▪ число столбцов таблицы – **1**

▪ число строк таблицы – **1**

➤ щелкните по кнопке **Вставить**;

➤ нажмите кнопку управления курсором ↓ и комбинацию клавиш **SHIFT+↑** для выделения содержимого таблицы;

➤ выберите из списка размер шрифта **11**, шрифт **Courier New**, начертание **полужирный**;

8. Проверьте работу макроса:

➤ выделите фрагмент с кодом:

Workbooks("Sales.xls").Worksheets("New").Range("B3").Value = 7

➤ нажмите комбинацию клавиш **ALT+CTRL+W**

9. Просмотрите VBA-код макроса.

10. Вернитесь в Microsoft Word.

11. Сохраните изменения.

12. Завершите работу Microsoft Word.

Задача 4. Пример использования макросов:

- создать макрос для копирования информации с одного рабочего листа книги MS Excel на другой рабочий лист.

Раздел 2. СОЗДАНИЕ ПРОСТЕЙШИХ ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ

В редакторе Visual Basic можно создавать собственные программы. Программа представляет собой набор инструкций, которые компьютер выполняет в заданном порядке.

Редактор Visual Basic отображает информацию в окнах: окно Project (Проект), окно Properties (Свойства) и окно программы.

Окно Project (Проект)

Можно использовать окно Project (Проект) для выбора макроса, с которым требуется работать. В этом окне в виде дерева представлены все проекты и содержащиеся в них макросы.

Это представление аналогично дереву в проводнике Windows. Корневыми или базовыми объектами в древовидной структуре является текущая рабочая книга, т. е. книга, в которой работают в данный момент, плюс любые другие рабочие книги и надстройки, которые могут быть открыты.

Окно Properties (Свойства)

Окно Properties (Свойства) используется для просмотра и изменения свойств, связанных с выбранным объектом. Для модулей обычно доступно только одно свойство – Name (Имя).

Окно программы

Окно программы содержит в верхней части два поля с раскрывающимися списками.

Поле слева – Object (Объект) – используется для выбора объекта для работы. Когда выполняется работа только с кодом, в этом поле отображается объект General (Общий).

Второе поле со списком – Procedure (Процедура) – используется для выбора отдельных макросов внутри текущего модуля.

По мере добавления макросов в модуль и удаления их из модуля они также добавляются в поле Procedure (Процедура) или удаляются из него.

Задача 1. Создание пользовательских функций в MS Excel:

- **ознакомиться с настройками редактора кода;**
- **создать пользовательскую функцию для расчета налога;**
- **научиться использовать функции.**

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Excel.

2. Создайте на рабочем листе **Лист1** таблицу следующего вида:

	А	В
1	Ставка налога на 01.01.2007	
2	до 7.440.000 рублей	9 процентов
3	от 7.440.001 рубля до 18.600.000 рублей	669.600 рублей + 15 процентов с суммы, превышающей 7.440.000 рублей
4	от 18.600.001 рубля до 26.040.000 рублей	2.343.600 рублей + 20 процентов с суммы, превышающей 18.600.000 рублей
5	от 26.040.001 рубля до 33.480.000 рублей	3.831.600 рублей + 25 процентов с суммы, превышающей 26.040.000 рублей
6	от 33.480.001 рубля и выше	5.691.600 рублей + 30 процентов с суммы, превышающей 33.480.000 рублей

3. Сохраните книгу с именем **Налог**.

4. Добавьте на ленту вкладку **Разработчик**.

5. Откройте окно **Microsoft Visual Basic**:

➤ вкладка **Разработчик** / кнопка **Visual Basic**.

6. Добавьте через меню **View** окно проекта и окно свойств:

➤ **View / Project Explorer**;

➤ **View / Properties Window**.

7. Выполните настройки:

➤ **Tools / Options**, откроется окно **Options**;

➤ снимите во вкладке **Editor** флажок **Auto Syntax Check**, это позволит уменьшить количество сообщений об ошибках;

➤ выберите вкладку **Editor Format** и просмотрите параметры настройки подсветки кода в поле **Code Colors**;

➤ щелкните **OK** для завершения настройки.

8. Вставьте новый модуль проекта: **Insert / Module**, будет создан модуль с именем **Module1**.

9. Переименуйте новый модуль в **Налог**:

➤ введите в окне свойств **Properties** в поле **Name** имя **Налог**.

10. В окне кода введите текст функции с именем **Налог** и аргументом **Доход**.

В конце строк нажимайте клавишу **Enter**.

11. Для отступа в начале строк используйте клавишу табуляции.

```
Public Function Налог(Доход As Currency) As Currency
If Доход <= 7440000 Then
    Налог = Доход * 0.09
End If
If Доход > 7440000 And Доход <= 18600000 Then
    Налог = 669600 + (Доход - 7440000) * 0.15
End If
If Доход > 18600000 And Доход <= 26040000 Then
    Налог = 2343600 + (Доход - 18600000) * 0.2
End If
If Доход > 26040000 And Доход <= 33480000 Then
    Налог = 3831600 + (Доход - 26040000) * 0.25
End If
If Доход > 33480000 Then
    Налог = 5691600 + (Доход - 33480000) * 0.3
End If
End Function
```

12. Закройте окно кода.

13. На рабочем листе **Лист2** создайте таблицу вида:

	А	В	С	Д
1	Место работы	Доход	Налог	
2	БГУ	9 420 000р.		
3	БГЭУ	1 320 000р.		
4	Остальные	1 573 000р.		
5	Итого			
6		Доплата		
7				

14. Вычислите налог с использованием созданной функции **Налог**:

➤ Выберите ячейку **С2**.

➤ Щелкните кнопку **Вставка функции fx** в строке формул. Открывается диалоговое окно **Мастер функций**.

➤ Выберите в поле со списком **Категория** значение **Определенные пользователем**.

➤ Прокрутите список функций и двойным щелчком выберите функцию **Налог**. Откроется диалоговое окно **аргументы функции**.

➤ В качестве аргумента функции выберите ячейку **В2**.

➤ Щелкните по кнопке **ОК**.

15. Выберите денежный формат для подсчитанного налога.

16. Скопируйте формулу в смежные ячейки **С3** и **С4**.

17. Посчитайте итоговый доход в ячейке **В5**, используя **Автосуммирование**.

18. Вычислите налог с итоговой суммы.

19. Посчитайте сумму доплаты как разность между налогом с итоговой суммой и налогами для каждого места работы.

20. Сохраните внесенные изменения.

Задача 2. Создание пользовательских функций в MS Excel:

▪ создать пользовательскую функцию для расчета надбавки за стаж.

Раздел 3. СОЗДАНИЕ ПРОЦЕДУР-ПОДПРОГРАММ

В редакторе Visual Basic можно создавать собственные процедуры-подпрограммы. Главное отличие процедуры от функции заключается в том, что функция возвращает в точку вызова некое значение, которое, как правило, является результатом обработки переданной функции информации. А процедура лишь выполняет какие-либо действия, но ничего в точку вызова не возвращает.

Процедуры удобно использовать для сокращения объема программы, выделяя в них часто используемые блоки операторов.

Нерационально каждый раз копировать участки кода в нужное место. Во-первых, увеличивается размер текста программы, во-вторых, если вам понадобится изменить что-нибудь в часто используемом наборе операторов, придется искать все такие участки и редактировать каждый из них.

Гораздо удобнее будет выделить часто используемый участок в виде самостоятельной процедуры и каждый раз, когда он будет нужен, вызывать его с помощью одной лишь строчки кода.

Задача 1: Создание процедуры-подпрограммы в MS Word:

- записать макрос выделения символа полужирным шрифтом;
- модифицировать макрос для выделения слова полужирным шрифтом;
- создать процедуру – подпрограмму для выделения заданного количества слов;
- создать процедуру-подпрограмму для выделения первого слова в каждом абзаце.

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Word.
2. Введите следующий текст:

Данные – величины, обрабатываемые программой. Имеется три основных вида данных: константы, переменные и массивы.

Константы – это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения. Константы представляются в виде лексем, изображающих фиксированные числовые, логические, символьные или строковые значения.

Массивы – последовательности логически связанных элементов одного типа, которым присвоено одно имя. Положение элемента в массиве однозначно определяется его индексами (одним, в случае одномерного массива, или несколькими, если массив многомерный).

3. Сохраните документ в своей рабочей папке с именем **Работа_с_Word**.

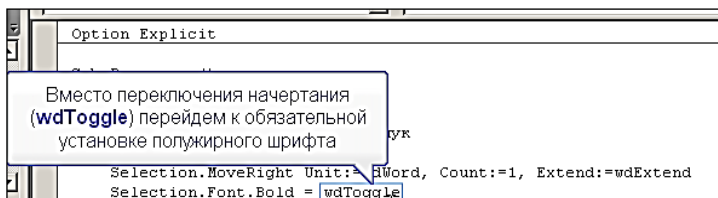
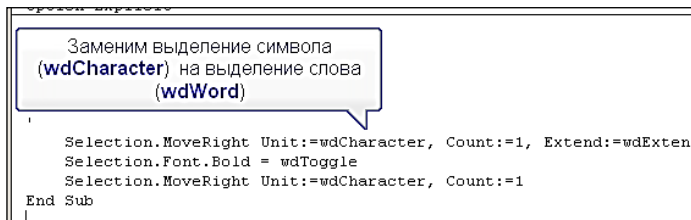
4. Создайте в этом документе макрос выделения символа полужирным шрифтом с именем **Выделение**:

- выделите один символ с помощью клавиатуры (**SHIFT + →**);
- щелкните на вкладке **Главная** по кнопке **Полужирный**;
- снимите выделение (**→**).

5. Измените созданный макрос так, чтобы он выделял целое слово:

➤ в окне **Microsoft Visual Basic** для проекта документа **Работа_с_Word** откройте папку **Modules**, которая содержит модули текущего проекта;

- откройте модуль **NewMacros** – модуль с макросом



- замените **wdCharacter** на **wdWord**, а **wdToggle** на **True**;
- закройте окно кода.

6. Выполните пробный запуск макроса:

- установите курсор перед словом, которое надо выделить;
- выполните макрос **Выделение**.

7. Выполните отмену действия.

8. Добавьте текст новой процедуры с именем **Повторы**:

- вернитесь в редактор кода, нажав **ALT + F11**;
- в окне кода введите:

```
Sub Повторы()
Dim i, количество As Integer
количество = InputBox("Введите число повторов")
For i = 1 To количество
    Выделение
Next i
End Sub
```

9. Установите курсор перед словами, которые надо выделить и выполните пробный запуск процедуры, введите число повторов.

Задача 2: Создание процедуры-подпрограммы в MS Word:

- создать процедуру-подпрограмму для выделения первого слова в каждом абзаце.

Для решения задачи надо выполнить следующие действия:

1. Добавьте процедуру для выделения первого слова в каждом абзаце.

Процедуре дайте имя **ПервыеСлова**.

```
Sub ПервыеСлова()  
Dim Абзац As Paragraph  
For Each Абзац In ActiveDocument.Paragraphs  
    Абзац.Range.Words(1).Font.Bold = True  
Next  
End Sub
```

2. Проверьте работу процедуры **ПервыеСлова**.

Задача 3: Создание процедуры-подпрограммы в MS Excel:

- записать макрос создания сводной таблицы;
- записать макрос удаления сводной таблицы;
- изменить процедуру-подпрограмму создания сводной таблицы, построенную при записи макроса;
- использовать элемент управления Кнопка для создания и удаления сводной таблицы.

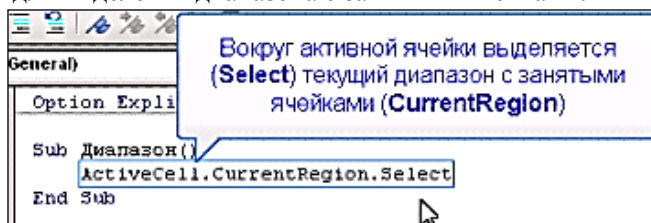
Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Excel.
2. Создайте таблицу:

	А	В	С	
1	Страна назначения	Продавец	Сумма заказа	
2	Украина	Батурин	13 200,00р.	
3	Россия	Рощин	55 902,00р.	
4	Украина	Воронова	46 578,00р.	
5	Беларусь	Крылова	19 621,80р.	
6	Беларусь	Воронова	107 937,00р.	
7	Беларусь	Крылова	43 344,00р.	
8	Россия	Батурин	16 698,60р.	
9	Украина	Тихонова	74 715,00р.	
10	Беларусь	Крылова	15 534,00р.	
11	Россия	Воронова	33 597,00р.	
12	Россия	Белова	48 446,40р.	
13	Россия	Воронова	3 024,00р.	
14	Россия	Воронова	45 139,50р.	
15	Россия	Воронова	13 440,00р.	

3. Сохраните рабочую книгу с именем **Сводная_таблица**.

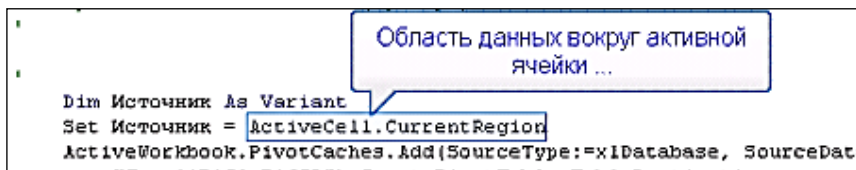
4. Выделите несколько первых строк диапазона с данными.
5. Запишите в текущей книге макрос создания сводной таблицы на текущем листе с именем **СводнаяТаблица**:
 - выберите на вкладке **Вставка** команду **Сводная таблица...**;
 - перетащите поле **Продавец** в область строк;
 - перетащите поле **Страна назначения** в область столбцов;
 - перетащите поле **Сумма заказа** в область данных.
6. Запишите макрос удаления сводной таблицы с именем **Удаление**:
 - выделите занятые сводной таблицей столбцы;
 - выберите на вкладке **Главная** команду **Удалить** в группе **Ячейки**.
7. Проанализируйте записанные коды.
8. Создайте новый модуль и в нем процедуру-подпрограмму с именем **Диапазон** для выделения диапазона с занятыми ячейками:



9. Выполните созданный модуль:
 - выделите произвольную ячейку с данными;
 - выполните процедуру-подпрограмму **Диапазон**, в результате ее работы должны выделяться все данные.
10. Внесите изменения в процедуру-подпрограмму **СводнаяТаблица**:
 - добавьте описание новой переменной **Источник**:

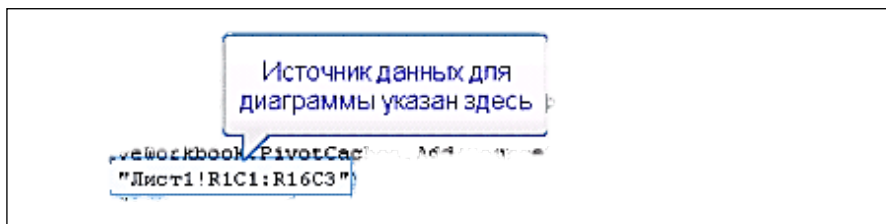


- область данных вокруг активной ячейки используйте в качестве источника:



- замените источник данных для диаграммы на переменную

Источник:



```
ActiveWorkbook.PivotCaches.Add(SourceType:=xlDatabase, SourceData:="Лист1!R1C1:R16C3").CreatePivotTable TableDestination:="Сводная таблица.xls!Лист1!R1C7", TableName:="СводнаяТабл
```

11. Перейдите к рабочей книге.
12. Выберите любую ячейку данных.
13. Выполните запуск измененной процедуры-подпрограммы **Сводная**

Таблица.

Сводная таблица будет построена по всему набору данных.

14. Выполните запуск процедуры-подпрограммы **Удаление**.

Сводная таблица будет удалена.

15. Добавьте кнопки для запуска процедур-подпрограмм:

- добавьте вкладку **Разработчик**, если ее нет;
 - на вкладке **Разработчик** включите кнопку **Режим конструктора** и элемент управления **Кнопка**;
 - щелчком на рабочем листе создайте кнопку;
 - в окне **Назначить макрос объекту** выберите соответствующий макрос;
 - измените надпись на кнопке в соответствии с назначенным макросом и настройте размеры кнопки;
 - на вкладке **Разработчик** выключите кнопку **Режим конструктора**.
16. Выберите любую ячейку данных.
 17. Проверьте работу кнопок.

Задача 4: Копирование данных из MS Excel в MS Word:

- подключить библиотеку MS Word;
- использовать коллекцию объектов MS Word, свойства объектов и методы.

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Excel.
2. Создайте три таблицы:

ПеребросДанных.xls:1			ПеребросДанных.xls:2			ПеребросДанных.xls:3		
	А	В		А	В		А	В
1	Фамилия	Сумма	1	Фамилия	Сумма	1	Фамилия	Сумма
2	Батурич	747 482,40р.	2	Батурич	596 368,50р.	2	Батурич	3 375 186,00р.
3	Белова	2 501 552,40р.	3	Белова	1 381 138,20р.	3	Белова	656 001,00р.
4	Воронова	3 375 401,10р.	4	Воронова	1 533 991,20р.	4	Воронова	2 220 122,00р.
5	Иванова	1 220 227,50р.	5	Иванова	1 371 210,60р.	5	Иванова	3 407 281,00р.
6	Королев	1 041 795,30р.	6	Королев	905 707,80р.	6	Королев	2 534 546,00р.
7	Крылова	1 927 301,70р.	7	Крылова	2 812 905,00р.	7	Крылова	3 080 245,00р.
8	Рощин	676 334,70р.	8	Рощин	850 349,70р.	8	Рощин	2 504 713,00р.
9	Сидоров	2 070 925,50р.	9	Сидоров	1 146 927,60р.	9	Сидоров	839 089,00р.
10	Тихонова	737 994,00р.	10	Тихонова	861 612,00р.	10	Тихонова	959 466,00р.
11			11			11		

3. Сохраните рабочую книгу с именем **ПеребросДанных**.
4. Перейдите в редактор кода и создайте новый модуль.
5. Добавьте ссылку на библиотеку MS Word, чтобы книга Excel могла управлять документом Word:

➤ **Tools / References...**;

➤ найдите и выберите **Microsoft Word 12.0 Object Library / ОК**.

6. Введите текст процедуры:

```
Sub ЗаполнениеДокумента()
Dim ПрограммаWord, ДокументWord As Variant

Set ПрограммаWord = New Word.Application
ПрограммаWord.Visible = True

End Sub
```

Запуск Word

Отображение окна Word

7. Выполните пробный запуск. Появится окно макроса с выделенным именем **ЗаполнениеДокумента**.

8. Щелкните по кнопке **Run**. Открывается окно **Microsoft Word**.

9. Закройте окно Microsoft Word без сохранения.

10. Измените код:

```
ПрограммаWord.Visible = True

Set ДокументWord = ПрограммаWord.Documents.Add

End Sub
```

Создание нового документа **Word**

11. Выполните пробный запуск. Открывается окно **Документ1 – Microsoft Word**.

12. Закройте окно Microsoft Word без сохранения.

13. Измените код:

```
Set ДокументWord = ПрограммаWord.Documents.Add  
ДокументWord.Paragraphs(1).Range.Text = "Исходные данные"  
End Sub
```

Добавим текст в первый абзац документа Word

14. Выполните пробный запуск. Открывается окно **Документ1 – Microsoft Word** с текстом "Исходные данные".

15. Закройте окно Microsoft Word без сохранения.

16. Измените код:

```
ДокументWord.Paragraphs(1).Range.Text = "Исходные данные"  
ДокументWord.Range.InsertParagraphAfter  
ДокументWord.Range.InsertParagraphAfter  
ДокументWord.Paragraphs(3).Range.Select  
End Sub
```

Вставляем два пустых абзаца и выделяем последний

17. Выполните пробный запуск. Открывается окно **Документ1 – Microsoft Word** с текстом "Исходные данные" и двумя пустыми абзацами.

18. Закройте окно Microsoft Word без сохранения.

19. Измените код:

```
ДокументWord.Paragraphs(3).Range.Select  
Worksheets(1).Range("A1").CurrentRegion.Copy  
ПрограммаWord.Selection.Paste  
End Sub
```

На первом рабочем листе копируем диапазон с данными

```
Worksheets(1).Range("A1").CurrentRegion.Copy  
ПрограммаWord.Selection.Paste
```

End Sub

Данные вставляем в документ Word

20. Выполните пробный запуск. В документе Word появится таблица.

21. Закройте окно Microsoft Word без сохранения.

22. Перейдите в Microsoft Excel.

23. На трех листах создайте диаграммы разного типа: гистограмму, линейчатую, круговую.

24. Перейдите в окно кода и измените его:

```

ДокументWord.
ДокументWord.
Worksheets(1).
ПрограммаWord.Selection.Paste
Worksheets(1).ChartObjects(1).Copy
ПрограммаWord.Selection.PasteSpecial Placement:=wdInline, _
    DataType:=wdPasteMetafilePicture
End Sub

```

На первом рабочем листе скопируем диаграмму

```

ПрограммаWord.Selection.Paste
Worksheets(1).ChartObjects(1).Copy
ПрограммаWord.Selection.PasteSpecial Placement:=wdInline, _
    DataType:=wdPasteMetafilePicture
End Sub

```

Вставим ее в документ Word, отбечение: в тексте (**wdInline**), тип: изображение формата метафайл (**wdPasteMetafilePicture**)

25. Выполните пробный запуск. Документ Word содержит таблицу и диаграмму.

26. Закройте окно Microsoft Word без сохранения.

27. Измените код:

```

Set ДокументWord = ПрограммаWord.Documents.Add
Докум
Докум
Докум
Докум
Dim Лист As Worksheet
For Each Лист In ActiveWorkbook.Sheets
    Лист.Range("A1").CurrentRegion.Copy
    ПрограммаWord.Selection.Paste
    Лист.ChartObjects(1).Copy
    ПрограммаWord.Selection.PasteSpecial Placement:=wdInline, _
        DataType:=wdPasteMetafilePicture
Next
End Sub

```

С помощью цикла **For Each ... Next** "Исходные данные" проделаем те же операции со всеми листами документа

28. Выполните пробный запуск. Документ Word содержит 3 таблицы и 3 диаграммы.

29. Закройте окно Microsoft Word без сохранения.

30. Измените код, чтобы отформатировать заголовок:

```

Лист.ChartObjects(1).Copy
ПрограммаWord.Selection.Paste
ПрограммаWord.Selection.PasteSpecial Placement:=wdInline, _
    DataType:=wdPasteMetafilePicture
Next
ДокументWord.Paragraphs(1).Range.Font.Size = 14
ДокументWord.Paragraphs(1).Range.ParagraphFormat.Alignment = _
    wdAlignParagraphCenter
End Sub

```

Размер шрифта текста первого абзаца -- 14 pt

```

Лист.ChartObjects(1).Copy
ПрограммаWord.Selection.Paste
ПрограммаWord.Selection.PasteSpecial Placement:=wdInline, _
    DataType:=wdPasteMetafilePicture
Next
ДокументWord.Paragraphs(1).Range.Font.Size = 14
ДокументWord.Paragraphs(1).Range.ParagraphFormat.Alignment = _
    wdAlignParagraphCenter
End Sub

```

Выравнивание -- по центру

31. Выполните пробный запуск. Документ Word содержит 3 таблицы и 3 диаграммы и отформатированный заголовок.

32. Сохраните документ с именем **ПеребросДанных**.

Раздел 4. УСЛОВНЫЕ ОПЕРАТОРЫ. ВСТРОЕННАЯ ПРОЦЕДУРА MSGBOX

Процедуры-подпрограммы используют переменные. *Переменная* – это именованная область памяти, где могут храниться различные данные, которые можно изменять во время выполнения программы.

У *переменной* есть несколько важнейших характеристик. Первая – имя *переменной*. Используя имя, мы можем обращаться к *переменной* в программе. Вторая характеристика – это *тип данных*, которые могут храниться в *переменной*. Тип определяет характер данных, которые мы можем хранить в *переменной*. Например, это могут быть числовые данные (возраст пользователя) и строковые данные (имя пользователя).

Задача 1: Создание процедуры-подпрограммы в MS Word:

- **создать процедуру-подпрограмму для подсчета количества слов в документе;**
- **изменить программу для нахождения количества появлений определенного слова в документе;**
- **обеспечить возможность ввода образца для поиска.**

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Word.
2. Введите следующий текст:

Огромным достоинством офисных приложений является то, что действия, выполняемые через пользовательский интерфейс, можно перевести в программный код на Visual Basic путем записи макроса.

Чтобы записать макрос, надо находиться в окне приложения, а не в окне редактора Visual Basic.

Чтобы просмотреть Код, выберите из меню **Tools** (Сервис) команду **Macros** (Макрос), а затем – строку **Macros** (Макросы). В диалоговом окне **Macros** (Макрос) выберите имя нужного макроса и щелкните кнопку **Edit** (Изменить).

Найти записанный макрос вы можете также и в окне проекта в папке Модули (Modules). Обычно в Excel вы найдете его в модуле Module 1 в папке Modules данной рабочей книги. А при работе с MS Word – в модуле NewMacros в папке Modules проекта Normal.

3. Сохраните документ в своей рабочей папке с именем **Текст_для_поиска**.
4. Создайте процедуру для поиска и подсчета количества слов:
 - перейдите в редактор кода;
 - создайте новый модуль;

- измените имя модуля на **Поиск**;
- введите текст процедуры:

```
Sub Поиск()
Dim слово As Range
Dim i As Integer
i = 0
For Each слово In ActiveDocument.Words
    i = i + 1
Next
MsgBox ("Всего слов: " & i)
End Sub
```

5. Выполните пробный запуск, появится окно диалога с результатом.
6. Щелкните **ОК** в окне диалога.
7. Измените процедуру, добавив образец для поиска:

```
Sub Поиск()
Dim слово As Range
Dim i As Integer
Dim образец As String

образец = "код"
i = 0
For Each слово In ActiveDocument.Words
    If слово = образец Then
        i = i + 1
    End If
Next
MsgBox ("Всего слов: " & i)
End Sub
```

Пробный запуск: F5

8. Выполните пробный запуск – **F5**. Слова не найдены. Причиной является то, что VBA относит к слову и последующий пробел.
9. Щелкните **ОК** в окне диалога.
10. Для удаления пробела можно использовать функцию **Trim**:

```
Sub Поиск()
Dim слово As Range
Dim i As Integer
Dim образец As String

образец = "код"
i = 0
For Each слово In ActiveDocument.Words
    If Trim(слово) = образец Then
        i = i + 1
    End If
Next
MsgBox ("Всего слов: " & i)
End Sub
```

11. Выполнить пробный пуск. Найдены не все слова. Причина – VBA различает строчные и прописные буквы.

12. Щелкните в окне диалога по кнопке **ОК**.

13. Преобразуйте найденное слово и образец к верхнему регистру функцией **UCase**:

```
Sub Поиск()  
Dim слово As Range  
Dim i As Integer  
Dim образец As String  
  
образец = "код"  
i = 0  
For Each слово In ActiveDocument.Words  
    If UCase(Trim(слово)) = UCase(образец) Then  
        i = i + 1  
    End If  
Next  
MsgBox ("Всего слов: " & i)  
End Sub
```

14. Выполните пробный пуск. Все работает правильно.

15. Щелкните по кнопке **ОК** в окне диалога.

16. Добавьте возможность ввода слова для поиска:

```
Sub Поиск()  
Dim слово As Range  
Dim i As Integer  
Dim образец As String  
  
образец = InputBox("Введите образец для поиска")  
i = 0  
For Each слово In ActiveDocument.Words  
    If UCase(Trim(слово)) = UCase(образец) Then  
        i = i + 1  
    End If  
Next  
MsgBox ("Всего слов: " & i)  
End Sub
```

Задавать образец будем
с помощью **InputBox**

17. Выполните пробный запуск. Для поиска введите слово по своему усмотрению.

Раздел 5. ОПЕРАТОРЫ ЦИКЛОВ

Циклы позволяют выполнять одни и те же команды много раз.

Основные виды циклов:

➤ **For – Next** с фиксированным количеством повторов. Выполняется заданное количество раз;

➤ **While – Wend** с предусловием. Если не верно условие, заданное на входе в цикл, может не выполниться ни разу;

➤ **Do – Loop** с постусловием. Выполняется по меньшей мере один раз.

Пример цикла For – Next:

```
For i = 1 To 10
```

```
MsgBox(i)
```

```
Next i
```

Ключевое слово **For** задает начало цикла. После него следует переменная **i**, которая увеличивается при каждом проходе цикла. В ходе работы конструкции For – Next значение **i** изменяется от **1** до **10**.

Начальное значение счетчика устанавливается при входе в цикл. Конечное значение переменной задается после ключевого слова **To**. Переменная **i** доступна внутри цикла.

Ключевое слово **Next** с указанием переменной, к которой оно относится, закрывает цикл.

Задача 1: Создание процедуры-подпрограммы в MS Word:

- создать программу для выделения указанного слова.

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Word.

2. Откройте документ с именем **Текст_для_поиска.doc**.

3. Запишите макрос для анализа команды выделения:

➤ выделите любое слово в документе;

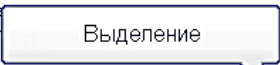
➤ создайте макрос в текущем документе;

➤ выберите на вкладке **Главная** из списка **Цвет выделения текста** желтый цвет, а затем из этого же списка значение **Нет цвета**.

4. Перейдите в окно **Microsoft Visual Basic – Текст_для_поиска**.

5. Перейдите в модуль макроса:

```
Sub Макрос1()  
'  
' Макрос1 М  
' Макрос за  
'  
Options.DefaultHighlightColorIndex = wdYellow  
Selection.Range.HighlightColorIndex = wdYellow  
Options.DefaultHighlightColorIndex = wdNoHighlight  
Selection.Range.HighlightColorIndex = wdNoHighlight  
End Sub
```



```
Options.DefaultHighlightColorIndex = wdYellow
Selection.Range.HighlightColorIndex = wdYellow
Options.DefaultHighlightColorIndex = wdNoHighlight
Selection.Range.HighlightColorIndex = wdNoHighlight
End Sub
```

Снятие выделения

6. Создайте новую процедуру, которая будет выделять указанное слово желтым цветом:

- перейдите в модуль **Поиск**:

```
Sub Поиск()
Dim слово As Range
Dim i As Integer
Dim образец As String

образец = InputBox("Введите образец для поиска")
i = 0
For Each слово In ActiveDocument.Words
If UCase(Trim(слово)) = UCase(образец) Then
i = i + 1
End If
Next
MsgBox ("Всего слов: " & i)
End Sub
```

Для ускорения работы сделаем копию программного кода

- скопируйте коды модуля **Поиск** и вставьте ниже:

```
Next
MsgBox
End Sub

Sub Поиск()
Dim слово As Range
Dim i As Integer
Dim образец As String

образец = InputBox("Введите образец для поиска")
i = 0
For Each слово In ActiveDocument.Words
If UCase(Trim(слово)) = UCase(образец) Then
i = i + 1
End If
Next
MsgBox ("Всего слов: " & i)
End Sub
```

Имя процедуры: **Выделение**

- в копии измените имя процедуры на **Выделение**;
- измените код, чтобы при обнаружении слово выделялось;
- удалите команды **i=i+1**;

```
Dim i As Integer
Dim образец As String

образец = InputBox("Введите образец для поиска")
i = 0
For Each слово In ActiveDocument.Words
If UCase(Trim(слово)) = UCase(образец) Then
i = i + 1
End If
Next
MsgBox ("Всего слов: " & i)
End Sub
```

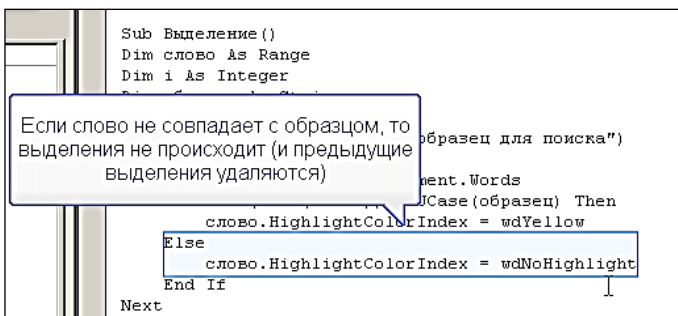
При обнаружении слова его необходимо выделить

- введите:

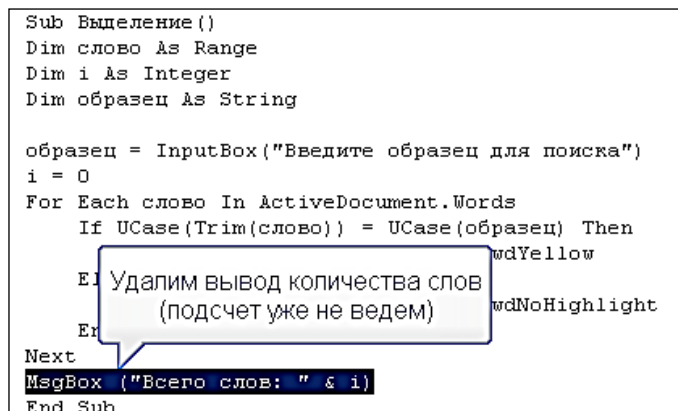
слово.HighlightColorIndex=wdYellow

Else

слово.HighlightColorIndex=wdNoHighlight



- удалите вывод количества слов:



7. Выполните пробный запуск:

- введите образец для поиска слова: **Код**;
 - перейдите в документ, нажав **Alt+F11** и просмотрите результат.
8. Выполните еще раз пробный запуск:
- оставьте поле ввода для поиска пустым, все выделения удалятся.

Раздел 6. МАССИВЫ

Массив – это именованный набор индексированных ячеек. Ячейки так же называют элементами или индексированными переменными.

У каждого массива 5 основных характеристик: имя, размерность, число элементов, номер первого элемента и тип элементов.

По размерности различают одномерные и многомерные массивы.

Одномерные массивы напоминают одну строку таблицы, каждая ячейка которой содержит какие-то данные. Многомерные массивы имеют больше измерений. Их можно сравнивать с таблицами, имеющими множество строк и столбцов и с наборами таблиц.

Не всегда количество элементов и размерность массива известны до начала работы программы. VBA умеет работать с динамическими массивами, параметры которых можно менять в ходе выполнения программы.

Задача 1: Создание процедуры-подпрограммы в MS Excel для переупорядочения рабочей книги, расставляющей рабочие листы в алфавитном порядке их имен:

- создать при помощи макроса пробный код для перемещения рабочего листа;
- определить, сколько листов в рабочей книге и имена листов, пользуясь окном **Immediate**;
- создать пробную процедуру-подпрограмму, определяющую все элементы коллекции листов;
- создать процедуру-подпрограмму сортировки листов рабочей книги.

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Excel.
2. Сохраните рабочую книгу с именем **Сортировка_листов**.
3. Создайте в рабочей книге 5 рабочих листов с именами: **Лист1**, **Лист2**, **Лист3**, **Лист4**, **Лист5**.
4. Создайте макрос, переставляющий рабочий лист **Лист3** на первое место в рабочей книге (перед листом **Лист1**):

```
Sub Макрос1()  
,  
' Макрос1 Макрос  
' Макрос записан 14.12.2011 (comp1)  
,  
,  
  
    Sheets("Лист3").Select  
    Sheets("Лист3").Move Before:=Sheets(1)  
End Sub
```

5. Расположите листы рабочей книги в следующем порядке: **Лист3, Лист1, Лист5, Лист2, Лист4.**

6. Определите, сколько листов в рабочей книге:

➤ в окне **Immediate** введите: **?ActiveWorkbook.Sheets.Count**

7. Определите имя первого листа:

➤ в окне **Immediate** введите: **?ActiveWorkbook.Sheets(1).Name**

8. Создайте пробную процедуру-подпрограмму для определения всех элементов коллекции листов:

```
Sub Test()  
    For Each Элемент In ActiveWorkbook.Sheets  
        MsgBox Элемент.Name  
    Next Элемент  
End Sub
```

9. Создайте пустую процедуру-подпрограмму с именем **Сортировка_листов** и назначьте комбинацию клавиш для ее запуска.

10. Введите код процедуры, который присваивает имена листов массиву и выводит их на экран:

```
Sub Сортировка_листов()  
    Dim Имена()  
    Кол = ActiveWorkbook.Sheets.Count  
    ReDim Имена(1 To Кол)  
    For i = 1 To Кол  
        Имена(i) = ActiveWorkbook.Sheets(i).Name  
        MsgBox Имена(i)  
    Next i  
End Sub
```

11. Удалите из текста процедуры оператор, который выводит имена листов на экран.

12. Создайте новую процедуру и поместите в нее код сортировки элементов массива:

```
Sub Сорт(Список())  
    Dim Первый As Integer, Последний As Integer  
    Dim i As Integer, j As Integer  
    Dim Временный  
    Первый = LBound(Список)  
    Последний = UBound(Список)  
    For i = Первый To Последний - 1  
        For j = i + 1 To Последний  
            If Список(i) > Список(j) Then  
                Временный = Список(j)  
                Список(j) = Список(i)  
                Список(i) = Временный  
            End If  
        Next j  
    Next i  
End Sub
```

13. Измените процедуру сортировки листов, добавив в нее вызов процедуры сортировки элементов массива:

```
Sub Сортировка_листов()  
    Dim Имена()  
    Кол = ActiveWorkbook.Sheets.Count  
    ReDim Имена(1 To Кол)  
    For i = 1 To Кол  
        Имена(i) = ActiveWorkbook.Sheets(i).Name  
    Next i  
    Call Сорт(Имена)  
End Sub
```

14. Добавьте в процедуру сортировки листов код для переупорядочения листов в соответствии с отсортированным массивом листов:

```
Sub Сортировка_листов()  
    Dim Имена()  
    Кол = ActiveWorkbook.Sheets.Count  
    ReDim Имена(1 To Кол)  
    For i = 1 To Кол  
        Имена(i) = ActiveWorkbook.Sheets(i).Name  
    Next i  
    Call Сорт(Имена)  
    For i = 1 To Кол  
        Sheets(Имена(i)).Move Sheets(i)  
    Next i  
End Sub
```

15. Объявите все используемые переменные и вставьте комментарии для облегчения чтения кода:

```
Sub Сортировка_листов()  
    ' Эта процедура переставляет листы активной рабочей книги  
    ' в алфавитном порядке их имен  
    Dim Имена()  
    Dim Кол As Integer  
    Dim i As Integer  
    Кол = ActiveWorkbook.Sheets.Count  
    ReDim Имена(1 To Кол)  
    ' Заполнение массива именами листов  
    For i = 1 To Кол  
        Имена(i) = ActiveWorkbook.Sheets(i).Name  
    Next i  
    ' Сортировка массива в возрастающем порядке  
    Call Сорт(Имена)  
    ' Перемещение листов в рабочей книге  
    For i = 1 To Кол  
        ActiveWorkbook.Sheets(Имена(i)).Move ActiveWorkbook.Sheets(i)  
    Next i  
End Sub
```

16. Проверьте работу процедуры.

Задача 2: Создание процедуры-подпрограммы в MS Excel для построения таблицы на рабочем листе:

- **создать процедуру для формирования двумерного массива, содержащего ФИО и телефон каждого студента группы, записать элементы двумерного массива на лист MS Excel в виде таблицы и выполнить ее оформление;**

- **создать процедуру-подпрограмму, выполняющую транспонирование таблицы;**

- **создать процедуру-подпрограмму для удаления созданной таблицы и результатов ее транспонирования.**

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Excel.

2. Сохраните рабочую книгу с именем **Список_группы**.

3. Создайте модуль VBA с именем **Список**.

4. Создайте процедуру-подпрограмму с именем **Массив**, формирующую двумерный массив из двух строк и двух столбцов. Данные массива вводите при помощи функции InputBox(). Вывод данных массива на экран выполните при помощи функции MsgBox().

5. Измените процедуру-подпрограмму **Массив**, убрав вывод на экран элементов массива и добавив код для формирования списка группы из элементов массива на листе с именем **Лист1**, начиная с ячейки **A1**.

6. Скопируйте модуль **Список** с новым именем **Оформленный_список**.

7. Измените процедуру-подпрограмму **Массив** в модуле **Оформленный_список**, добавив код для оформления таблицы:

- первая строка таблицы должна содержать имена полей **ФИО** и **Телефон**, шрифт – **Times New Roman**, начертание – **полужирный**, размер – **16 пунктов**, выравнивание – **По центру**;

- все остальные строки: шрифт – **Times New Roman**, начертание – **обычный**, размер – **12 пунктов**, выравнивание – **По левому краю**;

- все ячейки таблицы должны быть обрамлены.

8. Объясните все используемые переменные и вставьте комментарии для облегчения чтения кода.

9. Создайте на рабочем листе **Лист1** кнопку с надписью **Создание списка** для запуска процедуры-подпрограммы **Массив** из модуля **Оформленный_список**.

10. В модуле **Оформленный список** создайте процедуру-подпрограмму с именем **Транспонирование** для транспонирования таблицы, расположенной на листе с именем **Лист1**. Результат транспонирования разместите на листе с именем **Лист2**.

11. Создайте на рабочем листе **Лист1** кнопку с надписью **Транспонирование** для запуска процедуры-подпрограммы **Транспонирование** из модуля **Оформленный_список**.

12. В модуле **Оформленный список** создайте процедуру-подпрограмму с именем **Вызов**, последовательно вызывающую процедуры **Оформленный_список** и **Транспонирование**.

13. В модуле **Оформленный список** создайте процедуру-подпрограмму с именем **Очистка**, удаляющую всю информацию и оформление с листов **Лист1** и **Лист2**.

14. Замените кнопки на рабочем листе **Лист1** на кнопки, вызывающие процедуры **Вызов** и **Очистка**.

15. Измените модуль **Оформленный список** так, чтобы он мог работать с массивом задаваемой размерности.

Раздел 7. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ ПОЛЬЗОВАТЕЛЯ

Синтаксические ошибки в приложениях выявляются сразу после набора текста в редакторе. Для поиска логических ошибок в приложении пользователя полезно протестировать программу.

Сначала надо протестировать программу в обычном режиме работы. Если обычные проверки закончились успешно, можно считать, что программа корректно работает, не имеет синтаксических и логических ошибок.

После этого можно усложнить режим тестирования или целенаправленно вызвать ошибку.

Можно выполнить программу в пошаговом режиме, используя встроенные в редактор VBA средства для отладки программ.

Основной метод отладки – это пошаговое исполнение программы с использованием точек останова.

Чтобы создать в программе точку останова, достаточно щелкнуть мышью в редакторе на серой панели напротив команды, на которой нужно остановить выполнение программы. Там появится большая красная точка – программа будет остановлена в процессе выполнения. Строка будет подсвечена красным цветом.

Строка, подсвеченная при остановке программы, еще не выполнялась – редактор указывает на то, что эта строка будет выполнена следующей.

Таких точек останова можно установить столько, сколько нужно – на тех строках программы, где могут возникать ошибки. Чтобы убрать точку останова, надо щелкнуть по ней мышью.

Вместо точек останова, расставленных мышью или из меню, можно использовать оператор Stop. Он останавливает работу программы и переводит ее в режим отладки.

Так же режим отладки можно включить, нажав во время работы программы комбинацию клавиш CTRL + Pause/Break и нажав в появившемся окне кнопку Debug. Текущая строка будет выделена желтым цветом, напротив нее будет установлена желтая стрелочка. Однако такой способ обычно не позволяет точно попасть в то место программы, где находится предполагаемая ошибка.

Задача 1: Выполнение тестирования приложения и устранение обнаруженных недостатков:

- протестировать приложение Сортировка_листов, изменив количество листов в книге, их имена (прописные буквы, строчные буквы) и положение листов;
- обратить внимание на недостатки работы приложения:

- непрерывное обновление экрана при выполнении процедуры;
- прописные буквы идут раньше, чем строчные при упорядочении имен;
- если нет открытых книг, появляется сообщение об ошибке;
- если структура рабочей книги защищена, то метод Move не работает;
- после упорядочения активным становится последний лист;
- если прервать выполнение комбинацией CTRL+Break, VBA выдает сообщение об ошибке;

■ **изменить программу, чтобы исключить недостатки.**

Для решения задачи надо выполнить изменение процедуры-подпрограммы **Сортировка_листов** следующим образом:

```
Sub Сортировка_листов()
' Эта процедура переставляет листы активной рабочей книги
' в алфавитном порядке их имен
Dim Имена()
Dim Кол As Integer
Dim i As Integer
Dim Актив As Object
On Error Resume Next
Кол = ActiveWorkbook.Sheets.Count
If Err <> 0 Then Exit Sub ' Активной книги нет
' Проверка защиты структуры рабочего листа
If ActiveWorkbook.ProtectStructure Then
    MsgBox ActiveWorkbook.Name & " защищена", _
        vbCritical, "Переупорядочение книги невозможно"
    Exit Sub
End If
' Невосприятие клавиш <CTRL+Break>
Application.EnableCancelKey = xlDisabled
' Переопределение массива
ReDim Имена(1 To Кол)
' Сохранение ссылки на активный лист
Set Актив = ActiveSheet
' Заполнение массива именами листов
For i = 1 To Кол
    Имена(i) = ActiveWorkbook.Sheets(i).Name
Next i
' Сортировка массива в возрастающем порядке
Call Сорт(Имена)
' Отключение обновления экрана
Application.ScreenUpdating = False
' Перемещение листов в рабочей книге
For i = 1 To Кол
    ActiveWorkbook.Sheets(Имена(i)).Move ActiveWorkbook.Sheets(i)
Next i
' Восстановление исходного активного листа
Актив.Activate
End Sub
```


Раздел 8. СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОЙ ФОРМЫ

Пользовательская форма предоставляет возможность создавать диалоговые окна для приложений.

По своему поведению формы различаются на модальные, блокирующие при своем появлении документ, и немодальные. Тип формы определяется выбором параметра ShowModal (True – для модальных, False – для немодальных) на панели Properties.

Форма может содержать следующие элементы управления:

- **Надпись** (*Label*). Позволяет отобразить текст на форме.
- **Текстовое поле** (*TextBox*). Позволяет пользователю вводить текст.
- **Кнопка** (*CommandButton*). Используется для выполнения пользователем каких-либо действий.
- **Флажок** (*CheckBox*). Позволяет пользователю делать выбор из двух возможностей.
- **Переключатель** (*OptionButton*). Позволяет пользователю выбирать одну из нескольких возможностей.
- **Выпадающий список** (*ComboBox*). Позволяет пользователю выбрать одну из нескольких возможностей или ввести свой вариант.
- **Рамка** (*Frame*). Позволяет объединить несколько элементов в группу.

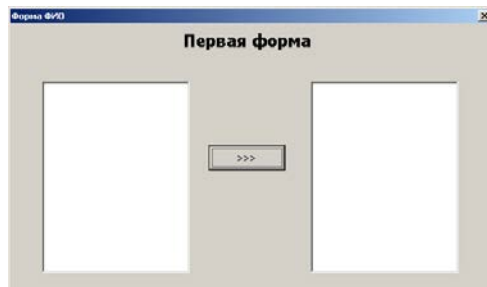
Задача 1: Создание пользовательской формы в MS Word для переноса текста из одного поля в другое:

- создать форму, содержащую элементы управления Надпись, Текстовое поле, Кнопка;
- настроить свойства формы и ее элементов;
- создать для кнопки процедуру обработки события – Нажатие кнопки;
- настроить запуск формы;
- ознакомиться с копированием программного кода;
- ознакомиться с возможностями импорта и экспорта;
- ознакомиться со способом защиты кода.

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Word. Сохраните пустой документ с именем **Документ1**.
2. Перейдите в окно редактора кода VBA. Вставьте в проект форму **Insert / UserForm**.
3. Добавьте на форму надпись **Первая форма**, размер шрифта **16** пт, начертание **жирный**:

- щелкните по кнопке **Label** на панели **Toolbox** и щелкните в области формы, появится рамка **label1**;
- в окне **Properties** в свойстве **Caption** введите **Первая форма**;
- выберите в окне **Properties** в свойстве **Font** размер **16** пт, начертание **жирный**.
- 4. Измените заголовок формы **UserForm1**:
- щелкните по заголовку формы **UserForm1**;
- в окне **Properties** в свойстве **Caption** введите **Форма_ФИО**.
- 5. Выполните пробный запуск формы щелчком мыши по кнопке **Run Sub/UserForm**.
- 6. Закройте окно появившейся формы.
- 7. Добавьте на форму текстовое поле с именем **ПолеЛевое**:
- на панели **Toolbox** щелкните по кнопке **TextBox**, растяните с помощью мыши область текстового поля на форме;
- в окне **Properties** в свойстве **(Name)** введите **ПолеЛевое**.
- 8. Добавьте аналогично правее на форме еще одно текстовое поле с именем **ПолеПравое**.
- 9. Добавьте на форму кнопку с именем **КнопкаЦентральная** и расположите ее между текстовыми полями:
- на панели **Toolbox** щелкните по кнопке **CommandButton**;
- щелкните в форме между текстовыми полями, появится изображение кнопки;
- в окне **Properties** в свойстве **(Name)** введите **КнопкаЦентральная**;
- в окне **Properties** в свойстве **Caption** введите **>>>**.



- 10. Выполните пробный запуск формы.
- 11. В текстовое поле **ПолеЛевое** введите ФИО, по умолчанию многострочный режим не поддерживается.
- 12. Закройте окно формы.
- 13. Определите многострочный режим для текстового поля **ПолеЛевое**:
- выберите текстовое поле **ПолеЛевое**;
- в окне **Properties** в свойстве **Multiline** выберите значение **True**.

14. Определите аналогично многострочный режим для текстового поля **ПолеПравое**.

15. Настройте всплывающую подсказку **Переброс текста** для кнопки:

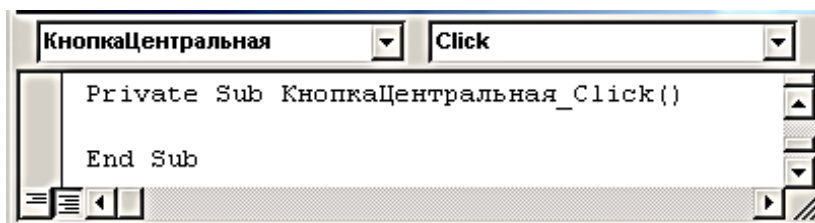
- щелчком мыши выберите кнопку с именем **КнопкаЦентральная**;
- в окне **Properties** в свойстве **ControlTipText** введите **Переброс текста**.

16. Выполните пробный запуск формы. Сохраните изменения в документе.

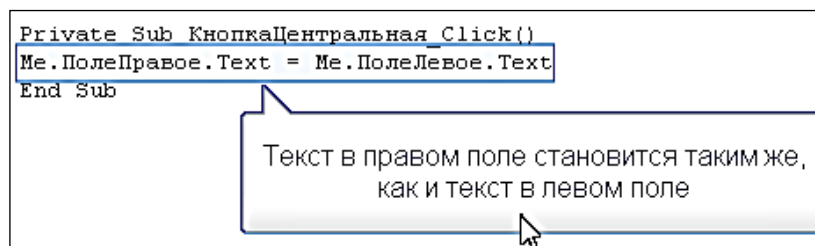
17. Перейдите в окно редактора кода VBA.

18. Перейдите в режим кода для создания процедуры обработки нажатия кнопки:

- дважды щелкните левой кнопкой мыши по кнопке. На экране появится заготовка для процедуры обработки при наступлении события щелчка **Click** по кнопке **КнопкаЦентральная**:



19. Введите оператор: **Me.ПолеПравое.Text = Me.ПолеЛевое.Text**

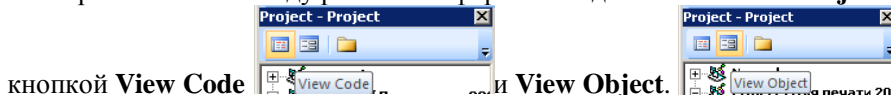


20. Выполните пробный запуск формы.

21. Проверьте работу формы: введите в левое поле ФИО и нажмите кнопку между текстовыми полями, в правом поле должны появиться символы из левого поля.

22. Закройте окно формы.

Переключаться между режимами формы и кода можно в окне **Project**



23. Перейдите в режим кода и добавьте оператор: **Me.ПолеЛевое = ""**
Левое поле после переброски данных будет очищаться.

```
Private Sub КнопкаЦентральная_Click()  
Me.ПолеПравое.Текст = Me.ПолеЛевое.Текст  
Me.ПолеЛевое = ""  
End Sub
```

После переброски данных левое поле будет очищаться

24. Выполните пробный запуск формы. Проверьте работу формы.

25. Перейдите в режим кода и внесите изменения:

Me.ПолеПравое.Текст = Me.ПолеПравое.Текст & Me.ПолеЛевое.Текст
Данные в правом поле будут не заменяться, а обновляться (к старому тексту будет добавляться новый):

```
Private Sub КнопкаЦентральная_Click()  
Me.ПолеПравое.Текст = Me.ПолеПравое.Текст & Me.ПолеЛевое.Текст  
Me.ПолеЛевое = ""  
End Sub
```

26. Выполните пробный запуск формы. Проверьте работу формы.

27. Перейдите в режим кода и внесите изменения:

Me.ПолеПравое.Текст = Me.ПолеПравое.Текст & vbCrLf & Me.ПолеЛевое.Текст
Добавления будут производиться с новой строки:

```
Private Sub КнопкаЦентральная_Click()  
Me.ПолеПравое.Текст = Me.ПолеПравое.Текст & vbCrLf & Me.ПолеЛевое.Текст  
Me.ПолеЛевое = ""  
End Sub
```

28. Выполните пробный запуск формы. Проверьте работу формы.

29. Создайте процедуру запуска формы:

- создайте модуль для процедуры: **Insert / Module**;
- введите: **Sub ЗапускФормы()
UserForm1.Show
End Sub**

(General) | ЗапускФормы

Option Explicit

```
Sub ЗапускФормы()  
UserForm1.Show  
End Sub
```

Команда показа формы UserForm1

30. Выполните пробный запуск формы.

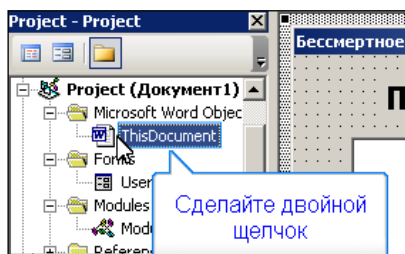
31. Назначьте для запуска формы сочетание клавиш **ALT + CTRL + Q**.

32. Проверьте работу комбинации клавиш **ALT + CTRL + Q**. Запустится процедура **ЗапускФормы**, которая запустит форму **UserForm1**.

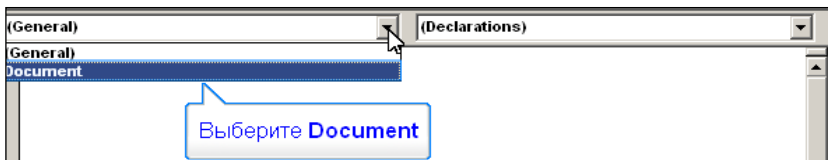
33. Вернитесь в редактор **VBA**.

34. Добавьте запуск формы при открытии документа:

- выполните двойной щелчок по файлу **ThisDocument**;



- выберите **Document** в списке **Object**:



- выберите **Open** в списке **Procedure**;
- введите: **Private Sub Document_Open()**

UserForm1.Show

End Sub

35. Сохраните внесенные изменения в документ и закройте его.

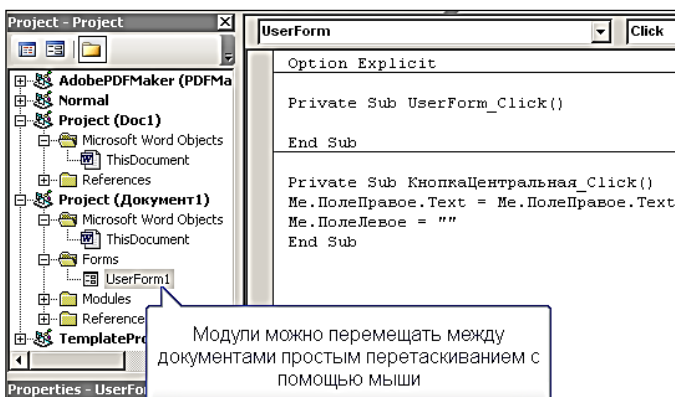
36. Проверьте работу, открыв сохраненный документ из приложения

Проводник.

37. Создайте новый текстовый документ и сохраните его с именем **Doc1.doc**.

38. Вернитесь в редактор кода.

39. Перетащите модуль **UserForm1** из **Project(Документ1)** в **Project(Doc1)**.



40. Закройте окно кода.

41. Сохраните документ.

42. Сохраните в отдельном файле программный код и форму:

- вернитесь в редактор кода и откройте форму **UserForm1**;
- вызовите контекстное меню для **UserForm1** и выберите команду

Export File...;

- выберите рабочую папку и щелкните кнопку **Сохранить**;
- закройте окно кода.

43. Перейдите в Microsoft Excel.

44. Создайте новую рабочую книгу.

45. Перейдите в окно кода.

46. Импортируйте в Excel форму, созданную в Word:

➤ **File / Import File...** / двойной щелчок по **UserForm1.frm** форма будет импортирована;

➤ откройте папку **Forms** и форму **UserForm1**.

47. Выполните защиту проекта:

➤ вызовите контекстное меню для **VBAProject (Книга1)**;

➤ выберите команду **VBAProject Properties...**; открывается окно диалога **VBAProject – Project Properties**;

➤ выберите вкладку **Protection**;

➤ включите флажок **Lock project viewing** для блокировки просмотра;

➤ введите пароль и его повтор, например **123**, пароль будет действовать после сохранения и закрытия файла;

➤ щелкните **ОК**;

➤ закройте окно кода;

➤ сохраните файл с именем **Книга1**;

➤ закройте окно **Microsoft Excel**.

48. Проверьте защиту проекта:

➤ откройте файл **Книга1** и перейдите в окно кода;

➤ откройте проект **VBAProject**. Появится окно для ввода пароля;

➤ не вводите пароль / **ОК**, появится информация о неверном пароле / **ОК**;

➤ введите пароль / **ОК** / откройте форму **UserForm1**.

Задача 2: Создание пользовательской формы в MS Word для управления запуском макроса:

- создать форму, содержащую элементы управления Текстовое поле, Полоса прокрутки, Кнопка;
- настроить параметры запуска формы;
- преобразовать форму в немодальную.

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Word.

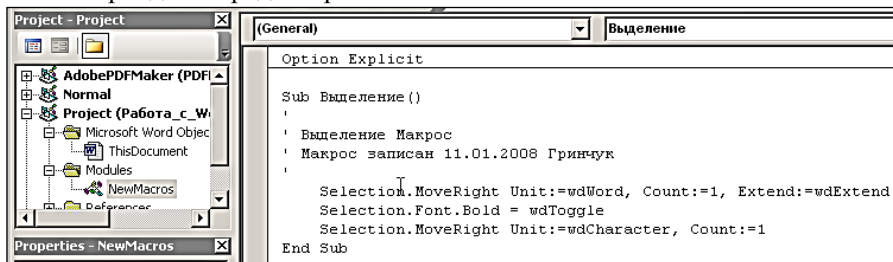
2. Создайте текстовый файл и сохраните его с именем **Данные.doc**:

Данные – величины, обрабатываемые программой. Имеется три основных вида данных: константы, переменные и массивы.

Константы – это данные, которые зафиксированы в тексте программы и не изменяются в процессе ее выполнения.

Массивы – последовательности логически связанных элементов одного типа, которым присвоено одно имя.

3. Перейдите в редактор VBA.



4. В документе создайте макрос выделения слова полужирным шрифтом.

5. Создайте форму для управления запуском макроса следующего вида:

➤ введите новое имя формы **ФормаМакроса**;

➤ в свойстве **Caption** введите заголовок формы **Запуск**;

➤ добавьте на форму текстовое поле, используя кнопку **TextBox** панели **Tool Box**;

➤ введите новое имя текстового поля **ПолеКоличество**;

➤ запретите вводить данные в текстовом поле: в свойстве **Enabled** значение **False**;

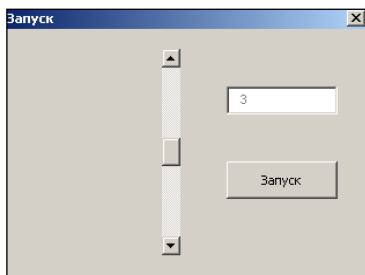
➤ добавьте полосу прокрутки при помощи кнопки **ScrolBar** на панели **ToolBox**;

➤ введите новое имя полосы прокрутки **ПолосаПрокрутки**;

➤ создайте кнопку, используя кнопку **CommandButton** на панели **ToolBox**;

➤ введите новое имя кнопки **КнопкаЗапуска**;

➤ введите надпись на кнопке **Запуск**.



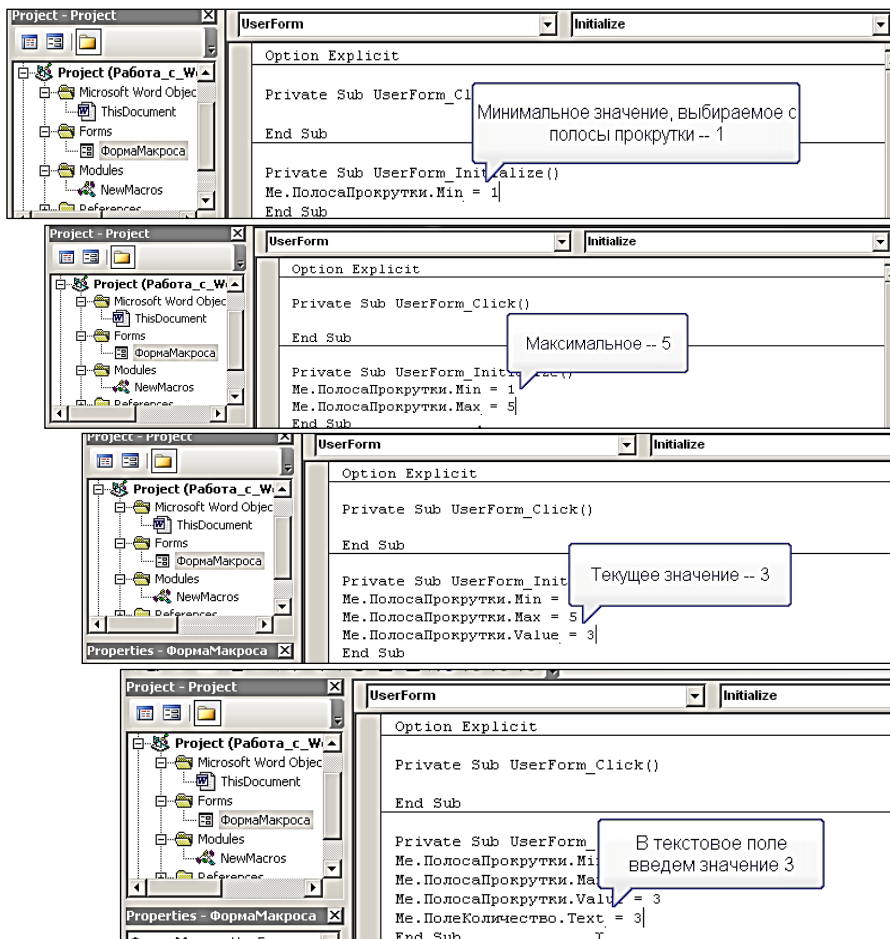
6. Выполните пробный запуск.

7. Закройте окно формы.

8. Настройте начальные состояния элементов формы:

➤ переключитесь в режим кода: щелкните по кнопке **View Code** в окне **Project**;

➤ выберите **UserForm** в списке **Object** выберите событие **Initialize** в списке **Procedure** и введите:



9. Выполните пробный запуск. Полоса прокрутки стартует со среднего значения (3), состояние изменяется от 1 (крайнее верхнее положение) до 5 (крайнее нижнее).

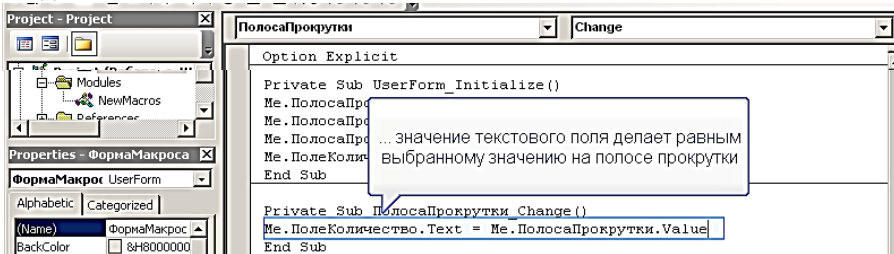
10. Закройте окно формы.

11. Переключитесь в режим кода: щелкните по кнопке **View Code** в окне **Project**.

12. Свяжите значения текстового поля с полосой прокрутки:

- установите курсор на новую строку после **End Sub**;
- выберите **ПолосаПрокрутки** в списке **Object**;

➤ выберите событие **Change** (при изменении) в списке **Procedure** и введите:



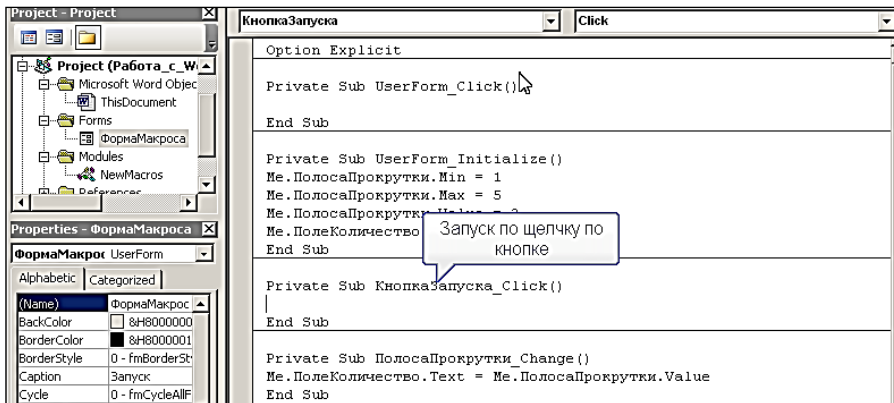
13. Выполните пробный запуск.

14. Закройте окно формы.

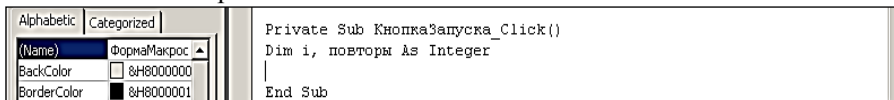
15. Переключитесь в режим кода: щелкните по кнопке **View Code** на панели **Project**.

16. Настройте количество запусков макроса в соответствии с выбранными значениями:

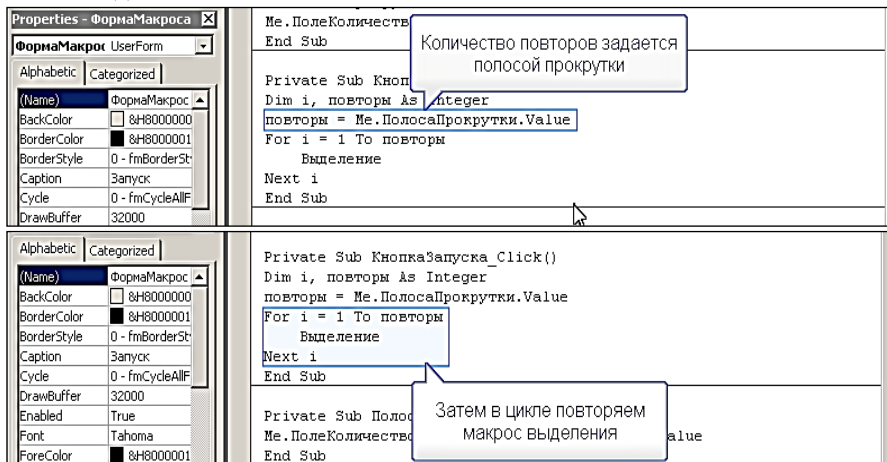
- установите курсор на новую строку после **End Sub**;
- выберите **КнопкаЗапуска** в списке **Object**;
- выберите событие **Click** в списке **Procedure**, появится:



➤ объявите переменные:



➤ **ВВЕДИТЕ:**



17. Выполните пробный запуск:

- щелкните по кнопке **Запуск**;
- прокрутите полосу прокрутки и щелкните вновь по кнопке **Запуск**;
- форма блокирует доступ к документу, т.е. является модальной.

18. Закройте окно формы.

19. Преобразуйте модальную форму в немодальную:

- выберите для свойства формы **ShowModal** значение **False**.

20. Выполните пробный запуск. Становится возможным перемещение курсора по тексту.

21. Сохраните внесенные изменения в документ.

Задача 3: Создание пользовательской формы в MS Word для заполнения типового договора:

- создать форму, содержащую элементы управления **Текстовое поле**, **Кнопка**;
- заполнить поля договора.

Для решения задачи надо выполнить следующие действия:

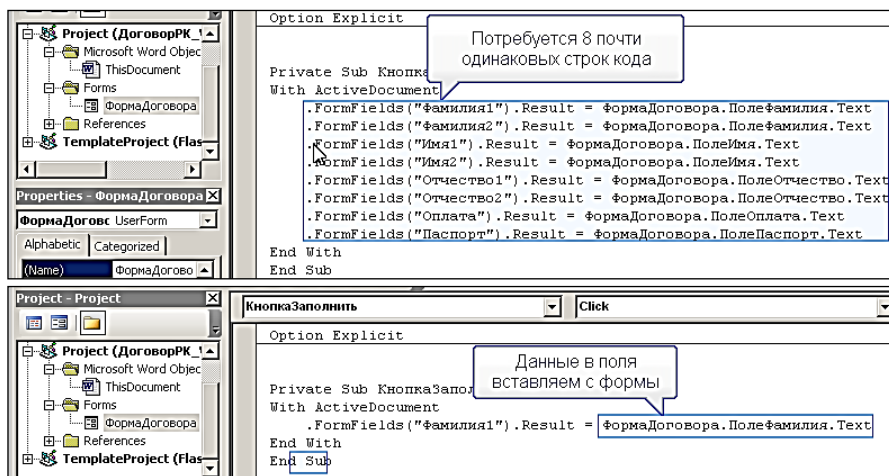
1. Запустите Microsoft Word.
2. Создайте документ по ниже приведенному образцу и сохраните с именем **Договор.doc**:

ДОГОВОР ОКАЗАНИЯ УСЛУГ	
г. Минск → → → → → → → → «09» января 2013 г.	
Закрытое акционерное общество «Аверест», именуемое в дальнейшем «Заказчик», в лице директора Иванова Ивана Ивановича, действующего на основании Устава, с одной стороны, и Фамилия Имя Отчество, именуемый в дальнейшем «Исполнитель», с другой стороны, заключили договор о нижеследующем.	
1. Предмет договора.	
1.1. → Заказчик поручает, а Исполнитель принимает на себя обязательства оказать следующие услуги.	
1.1.1.	
2. Стоимость услуг и порядок их оплаты.	
2.1. Стоимость услуг Исполнителя по пунктам 1.1.1, 1.1.3 за один час составляет Сумма оплаты: рублей.	
6. Реквизиты сторон:	
Заказчик:	Исполнитель:
Закрытое акционерное общество «Аверест»	Фамилия Имя Отчество
200030, г. Минск, ул. Октябрьская, 4-406,	паспорт: Номер паспорта
тел. 2-104-106	
р/с №3013013013013 в ф.ле МГД "Финк" г. Минска	
→ → Заказчик → → → → → → → →	Исполнитель
→ → → → → → → →	

- Поместите фамилию исполнителя в поле Word, т. к. она изменяется:
 - выделите фамилию исполнителя в договоре;
 - щелкните по кнопке **Поле** на панели **Разработчик** в группе **Элементы управления** в раскрывающемся списке **Формы предыдущих версий**;
 - щелкните по кнопке **Свойства** на панели **Разработчик**;
 - введите в поле **закладка** имя поля **Фамилия1** и щелкните кнопку **ОК**.
- Аналогично:
 - Имя и Отчество исполнителя поместите в поля Word с именами **Имя1**, **Отчество1**;
 - сумму оплаты поместите в поле Word с именем **Оплата**;
 - Фамилию, Имя и Отчество исполнителя из раздела **Реквизиты сторон** поместите в поля Word с именами **Фамилия2**, **Имя2**, **Отчество2**;
 - номер паспорта поместите в поле Word с именем **Паспорт**.
- Перейдите в редактор **VBA**.
- Создайте следующую форму для заполнения договора:
 - имя формы **ФормаДоговора**, заголовок формы **ВводДанных**;
 - имя первого поля **ПолеФамилия**;
 - имя второго поля **ПолеИмя**;
 - имя третьего поля **ПолеОтчество**;
 - имя четвертого поля **ПолеОплата**;

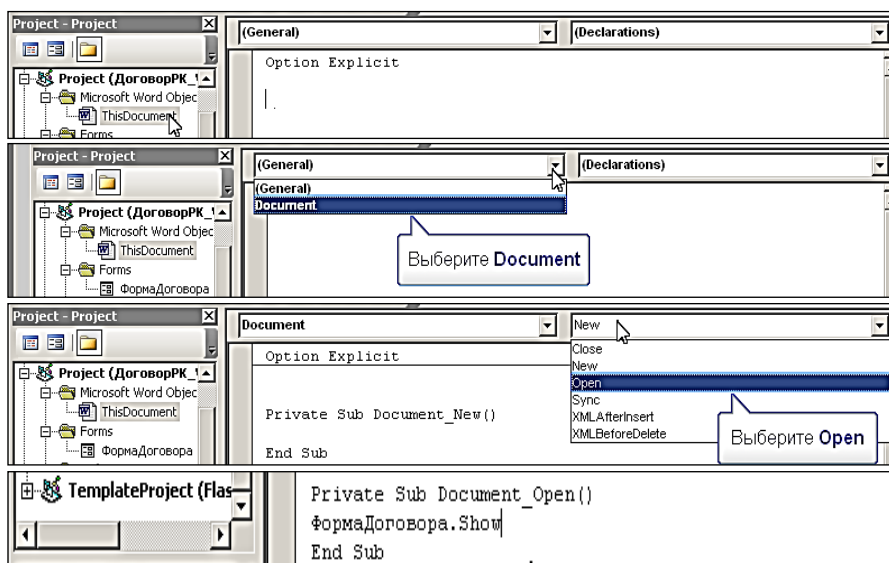
- имя пятого поля **ПолеПаспорт**;
- имя кнопки **КнопкаЗаполнить**.

7. Перейдите в режим кода двойным щелчком мыши по кнопке на форме и введите:



8. Организуйте запуск формы при открытии документа:

- выполните двойной щелчок по **ThisDocument**:



9. Перейдите в окно документа.
10. Отключите отображение непечатаемых символов.
11. Установите защиту формы для блокировки текста договора щелчком мыши по кнопке **Защитить документ**.
12. Сохраните внесенные изменения.
13. Закройте документ.
14. Проверьте работу формы, открыв документ **Договор**. Откроется диалоговое окно для ввода данных.
15. Заполните поля формы данными.
16. Щелкните по кнопке **Заполнить**.
17. Закройте окно **Ввод данных**.
18. Проверьте результат.

Задача 4: Создание пользовательской формы в MS Word для заполнения рекламного сообщения:

■ создать форму, содержащую элементы управления Рисунок, Набор вкладок, Текстовое поле, Поле со списком, Флажок, Переключатель, Кнопка;

■ **заполнить поля рекламного сообщения.**

Для решения задачи надо выполнить следующие действия:

1. Запустите Microsoft Word и сохраните документ с именем **Document1**.

2. Создайте пустую таблицу с невидимыми границами из одного столбца и трех строк:

- в верхней ячейке – выравнивание по центру;
- в средней – по ширине;
- в последней – по правому краю.

3. Перейдите в редактор кода Visual Basic и создайте форму.

4. Добавьте на форму рисунок:

- уберите границы рисунка: в поле **BorderStyle – 0**.

5. Укажите файл с рисунком:

- в поле **Picture** щелкните кнопку с тремя точками;
- в окне **Load Picture** выберите файл рисунка.

6. Разрешите изменение размеров рисунка:

- в поле **PictureSizeMode – 1**.

7. Создайте набор вкладок.

8. Добавьте третью вкладку:

- в контекстном меню команда **NewPage**.

9. Измените подпись первой вкладки на **Приветствие**:

- введите в поле **Caption – Приветствие**.

10. Введите текст всплывающей подсказки **Обращение к клиенту**:

- в поле **ControlTipText – Обращение к клиенту**.

11. Подпись второй вкладки **Предложение**, текст подсказки – **Предлагаемые варианты**.

12. Подпись третьей вкладки **Подпись**, текст подсказки – **Указание отправителя**.

13. Создайте кнопку с именем **КнопкаЗаполнить** и подписью **Заполнить**.

14. На вкладке **Приветствие** создайте:

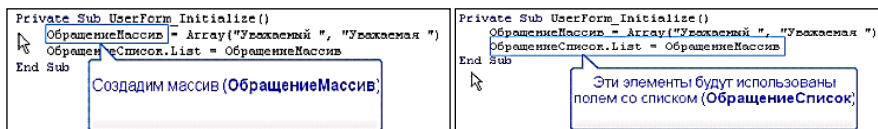
- поле со списком с именем **ОбращениеСписок** для выбора обращения к клиенту – **Уважаемый** или **Уважаемая**;
- поле с именем **ПолеИмя** для ввода фамилии и инициалов клиента.

15. Перейдите к созданию кода.

16. Из списка **Object** выберите **UserForm**. Из списка **Procedure** выберите **Initialize**:

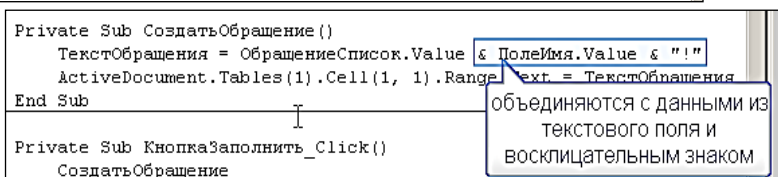
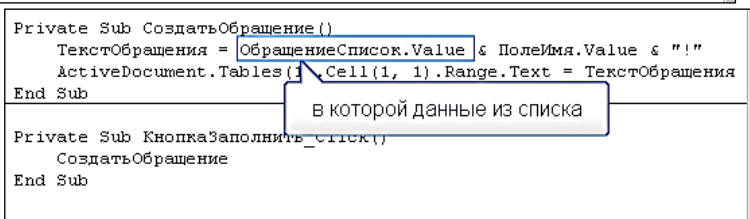
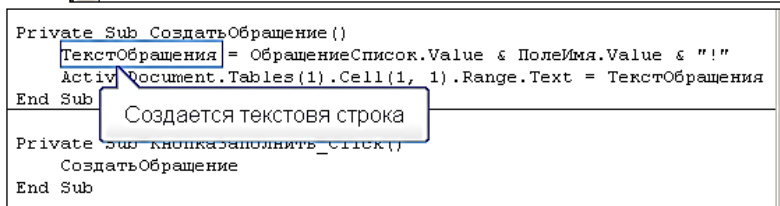
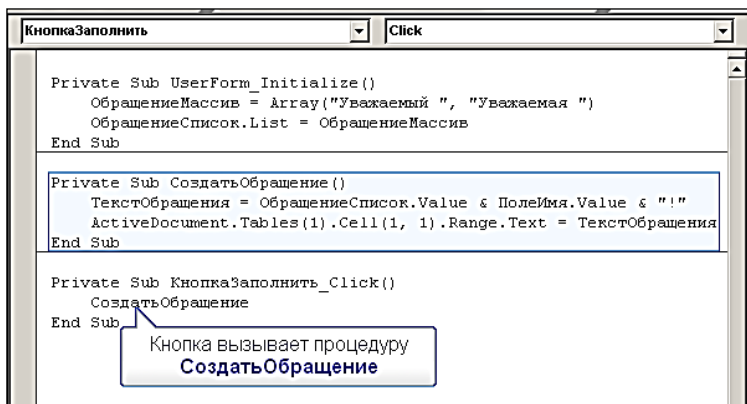


17. Создайте массив **ОбращениеМассив**:



18. Перейдите к форме и выполните пробный запуск.

19. Вернитесь к созданию кода и добавьте код для кнопки:



```
Private Sub СоздатьОбращение()
    ТекстОбращения = ОбращениеСписок.Value & ПолеИмя.Value & "!"
    ActiveDocument.Tables(1).Cell(1, 1).Range.Text = ТекстОбращения
End Sub
```

Полученный текст
внедряется в активный
документ

```
Private Sub СоздатьОбращение()
    ТекстОбращения = ОбращениеСписок.Value & ПолеИмя.Value & "!"
    ActiveDocument.Tables(1).Cell(1, 1).Range.Text = ТекстОбращения
End Sub
```

в первую таблицу

```
Private Sub КнопкаЗаполнить_Click()
    СоздатьОбращение
End Sub
```

```
Private Sub СоздатьОбращение()
    ТекстОбращения = ОбращениеСписок.Value & ПолеИмя.Value & "!"
    ActiveDocument.Tables(1).Cell(1, 1).Range.Text = ТекстОбращения
End Sub
```

в первую ячейку

```
Private Sub КнопкаЗаполнить_Click()
    СоздатьОбращение
End Sub
```

20. Перейдите к форме и выберите для свойства **ShowModal** значение **False**, так как для правильной работы эта форма должна быть немодальной.

21. Выполните пробный запуск:

- выберите обращение из списка;
- в поле введите фамилию и инициалы;
- щелкните кнопку **Заполнить**, первая строка таблицы должна заполниться.

22. Вернитесь в окно кода.

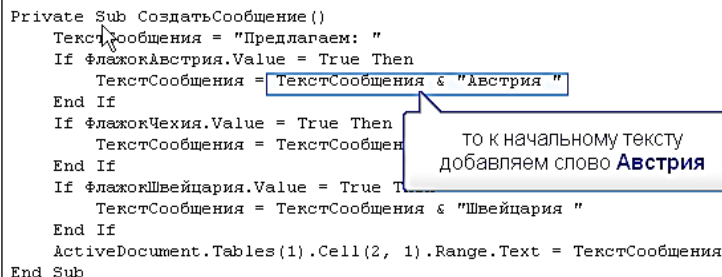
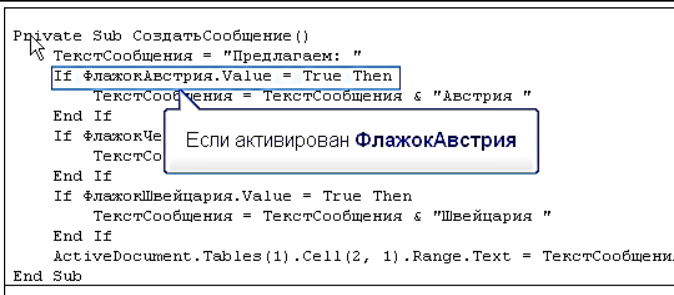
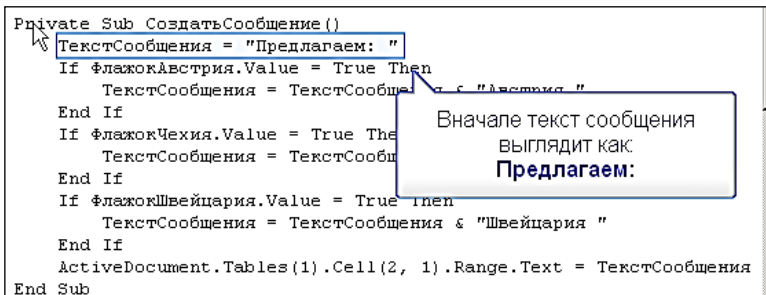
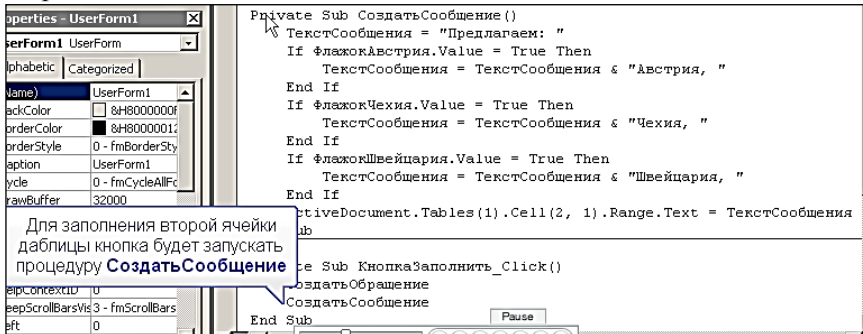
23. Создайте на второй вкладке набор флажков для имеющихся в наличии вариантов:

- Name – **ФлажокАвстрия**, Caption – **Австрия**;
- Name – **ФлажокЧехия**, Caption – **Чехия**;
- Name – **ФлажокШвейцария**, Caption – **Швейцария**.

24. Выполните пробный запуск. Флажки устанавливаются независимо.

25. Вернитесь в окно кода.

26. Создайте процедуру с именем **СоздатьСообщение** для заполнения второй ячейки:



27. Выполните пробный запуск. Недостаток формы – перечисления идут в одну строку без знаков препинания.

28. Вернитесь в окно кода и выполните изменение:

```
Private Sub СоздатьСообщение()  
    ТекстСообщения = "Предлагаем: "  
    If флажокАвстрия.Value = True Then  
        ТекстСообщения = ТекстСообщения & vbCrLf & "Австрия"  
    End If  
    If флажокЧехия.Value = True Then  
        ТекстСообщения = ТекстСообщения & vb  
    End If  
    If флажокШвейцария.Value = True Then  
        ТекстСообщения = ТекстСообщения & vb
```

Перед названием страны добавим переход на новую строку (vbCrLf)

```
Private Sub СоздатьСообщение()  
    ТекстСообщения = "Наша фирма предлагает путевки на " & _  
        "отдых в лучших горных курортах Европы: "  
    If флажокАвстрия.Value = True Then  
        ТекстСообщения = ТекстСообщ  
    End If  
    If флажокЧехия.Value = True Then  
        ТекстСообщения = ТекстСообщ
```

Создадим более развернутый начальный текст

29. Выполните пробный запуск.

30. Вернитесь в окно кода и выполните изменение:

```
Private Sub СоздатьСообщение()  
    ТекстСообщения = "Предлагаем: "  
    If флажокАвстрия.Value = True Then  
        ТекстСообщения = ТекстСообщения & "Австрия "  
    End If  
    If флажокЧехия.Value = True Then  
        ТекстСообщения = ТекстСообщения & "Чехия "  
    End If  
    If флажокШвейцария.Value = True Then  
        ТекстСообщения = ТекстСообщения & "Швейцария "  
    End If  
    ActiveDocument.Tables(1).Cell(2, 1).Range.Text = ТекстСообщения  
End Sub  
  
Private Sub КнопкаЗаполнить_Click()  
    СоздатьОбращение  
    СоздатьСообщение  
End Sub
```

Итоговый текст помещаем во вторую строку таблицы

31. Выполните пробный запуск.

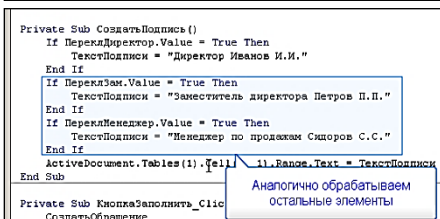
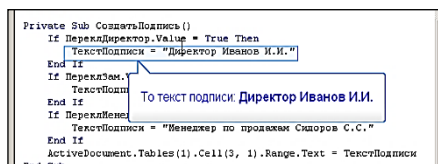
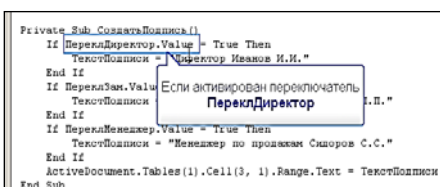
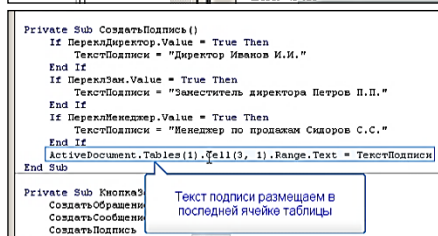
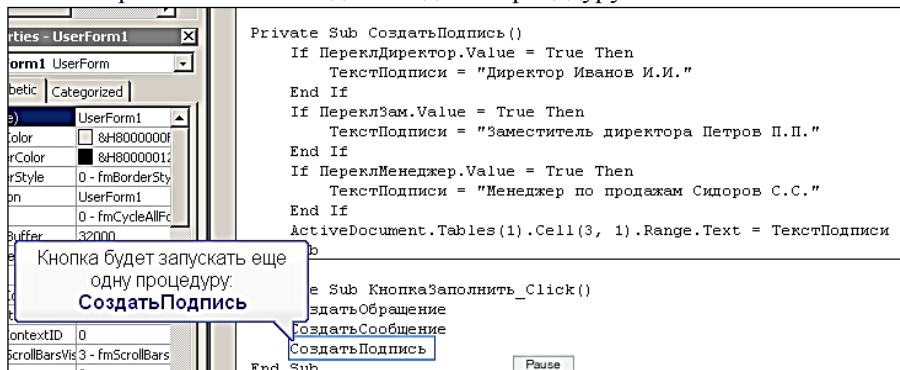
32. Вернитесь в окно кода.

33. Создайте на третьей вкладке набор переключателей для выбора вариантов подписи:

- Name – ПереклДиректор, Caption – Директор;
- Name – ПереклЗам, Caption – Заместитель директор;
- Name – ПереклМенеджер, Caption – Менеджер по продажам.

34. Выполните пробный запуск. Может быть активирован только один переключатель.

35. Вернитесь в окно кода и создайте процедуру:



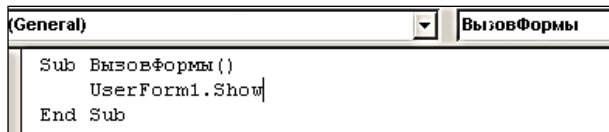
36. Выполните пробный запуск. Будут заполнены все три строки таблицы.

37. Вернитесь в окно кода и измените заголовок формы:

➤ **Caption – Заполнение письма.**

38. Создайте новый модуль с процедурой запуска формы:

➤ **Insert / Module.**



39. Перейдите в **MS Word** и назначьте сочетание клавиш **Alt + Ctrl + Q** для запуска формы.

ЛИТЕРАТУРА

Безручко, В. Т. Информатика. Курс лекций : учебное пособие / В. Т. Безручко. – М.: ИНФРА-М, 2013. – 432 с.

Гуда, А. Н. Информатика и программирование : компьютерный практикум / А. Н. Гуда, В. И. Колесников. – М.: Дашков и К, 2010. – 240 с.

Исаев, Г. Н. Моделирование информационных ресурсов : теория и решение задач: учебное пособие / Г. Н. Исаев. – М.: ИНФРА-М, 2012. – 224 с.

Немцова, Т. И. Практикум по информатике : учебное пособие / Т. И. Немцова, Ю. В. Назарова. – М.: ИНФРА-М, 2011. – 320 с.

Симонович, С. В. Информатика. Базовый курс / С. В. Симонович. – СПб.: Питер, 2012. – 640 с.

Слепцова, Л. Д. Программирование на VBA в Microsoft Office 2010. Самоучитель / Л. Д. Слепцова. – М.: Вильямс, 2010 – 432 с.

Трофимов, В. В. Информационные технологии / В. В. Трофимов. – М.: Высшее образование, 2011. – 632 с.