

СЕВЕРО-ЗАПАДНЫЙ ОТКРЫТЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

Методические указания к выполнению практического занятия № 1

Санкт-Петербург
2015

ОСНОВНЫЕ ОПЕРАТОРЫ ЯЗЫКА VBA

1. Цель работы

Научиться составлять программы разветвленных и циклических алгоритмов

2. Основные теоретические положения

2.1. Программирование разветвленных алгоритмов

Для организации разветвлений в программе используется оператор перехода, который имеет структуру

If ... Then ... Else ... End If

После ключевого слова **If** записывается проверяемое условие. Если оно выполняется, осуществляется переход к операторам, расположенным после ключевого слова **Then**. Если условие не выполняется, осуществляется переход к операторам после ключевого слова **Else**.

Пример 1

Вычислить
$$y = \begin{cases} x^2, & \text{если } x < 0 \\ 5x, & \text{если } x \geq 0 \end{cases}$$

где x – значение ячейки A3 листа Excel. Процедура вычислений имеет вид

Sub Пример1()

Присвоение переменной X значения, взятого из ячейки A3 Excel

X = Range("A3").Value

If X < 0 Then

Y = X ^ 2

Else

Y = 5 * X

End If

End Sub

Если число условий больше двух, можно использовать конструкцию

If ... Then ...

ElseIf ... Then ...

Else

End if

После ключевого слова **If** записывается первое проверяемое условие. Если оно не выполняется, второе условие запишется после ключевого слова **ElseIf**.

Пример 2

Вычислить

$$y = \begin{cases} x^3, & \text{если } x \leq -2 \\ \text{Cos}x, & \text{если } -2 < x \leq 0 \\ x+2, & \text{если } x > 0 \end{cases}$$

x – значение ячейки B1 листа Excel.

Процедура примет вид

```
Sub Пример2()  
    X = Range("B1").Value  
    If X <= -2 Then  
        Y = X ^ 3  
    ElseIf X <= 0 Then  
        Y = cos(X)  
    Else  
        Y = X + 2  
    End If  
End Sub
```

В одном условном операторе можно использовать несколько проверок дополнительных условий (несколько ключевых слов ElseIf).

2.2. Программирование циклических алгоритмов. Обработка массивов

Массивы – это упорядоченные наборы данных, например A_1, A_2, \dots, A_{10} – одномерный массив из десяти элементов.

Матрица

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

представляет собой двумерный массив из двух строк и трех столбцов - $B(2,3)$.

При работе с массивами их обязательно следует описывать в начале программы.

Например, оператор

```
Dim C(20) As Integer
```

описывает массив C , в котором двадцать целых чисел.

Оператор

```
Dim D(4,6) As Single
```

описывает двумерный массив (матрицу), в которой четыре строки и шесть столбцов, то есть $4 * 6 = 24$ числа вещественного типа.

Для перебора всех элементов массивов, а также для организации других повторяющихся вычислений используются операторы цикла (повтора). Их в VBA несколько:

For ... Next – повторяет набор инструкций заданное число раз;

While ... Wend – повторяет набор инструкций, пока некоторое условие выполняется;

Do While ... Loop – повторяет набор инструкций, пока условие выполняется (или вторая модификация - **Do Until ...Loop** – пока условие не выполняется).

Рассмотрим структуру и применение этих операторов.

2.2.1. Оператор For ... Next

Оператор **For** (для) ... **Next** (следующий) имеет синтаксис

For [Счетчик шагов] = [Начало] **To** [Конец] **Step** [Шаг]

Операторы
(Тело цикла)

Next (конец цикла)

Работает оператор следующим образом: оператор **For** начинает цикл и присваивает **Счетчику шагов** начальное значение. После этого выполняются операторы в теле цикла и оператор **Next** возвращает управление оператору **For** для увеличения значения **Счетчика шагов**. Оператор **For** проверяет, не достиг ли счетчик шагов конечного значения. Если нет – цикл повторяется. Если достиг – происходит переход к оператору, стоящему после **Next**.

Пример 3

Создадим программу, повторяющую звуковой сигнал 15 раз.

Процедура имеет вид

```
Sub Сигнал( )  
  Dim Счетчик As Integer  
  For Счетчик = 1 To 15  
    Веер  
  Next  
End Sub
```

После запуска программы (F5) компьютер будет сигналить.

2.2.2. Оператор While ... Wend

Оператор **While** (пока) ... **Wend** (конец оператора) имеет структуру

```
While [Условие]  
  [Операторы]  
Wend
```

Работает оператор так. Если условие выполняется, работают все операторы до **Wend**. Затем управление возвращается оператору **While** и цикл повторяется.

Пример 4

Организуем суммирование элементов массива $B(40)$, пока не встретится отрицательный элемент.

```

Sub Оператор_WhileWend( )
Dim B(40) As Single
Dim S As Single
Dim Счетчик As Integer
S = 0
Счетчик = 1
    While B(Счетчик) >= 0
        S = S + B(Счетчик)
        Счетчик = Счетчик + 1
    Wend
End Sub

```

2.2.3. Оператор Do ... Loop

Существует два типа этих операторов.

Первый из них **Do While ... Loop** работает аналогично оператору **While ... Wend**. Второй **Do Until ... Loop** организует повтор тогда, когда условие **не** выполняется.

Составим программу для предыдущего примера с оператором **Do While ... Loop**

```

Sub ОператорDoWhile( )
Dim B(40), S As Variant
Dim I As Integer
S = 0
I = 1
    Do While B(I) >= 0
        S = S + B(I)
        I = I + 1
    Loop
End Sub

```

Эта же программа с использованием оператора **Do Until ... Loop** имеет вид

```

Sub ОператорDoUntil( )
Dim B(40), S As Variant
Dim I As Integer
S = 0
I = 1
    Do Until B(I) < 0
        S = S + B(I)
        I = I + 1
    Loop
End Sub

```

2.2.4. Обработка двумерных массивов

Двумерные массивы представляют собой двумерные матрицы. Например, массив $L(7,8)$ – матрица, в которой 7 строк и 8 столбцов.

Чтобы провести обработку двумерного массива, нужно организовать сложный (вложенный) цикл. Как правило во внешнем цикле будет организован перебор строк массива, во внутреннем – перебор столбцов.

Пример 5

Составим программу для вычисления произведения всех элементов массива $N(5,3)$.

Процедура имеет вид

```
Sub ДвумерныйМассив( )  
Dim N(5,3), P As Variant  
Dim I, J As Integer  
P = 1  
  ‘Перебор строк  
  For I = 1 To 5  
    ‘Перебор столбцов  
    For J = 1 To 3  
      ‘Вычисление произведения  
      P = P * N(I,J)  
    ‘Завершение цикла по столбцам  
  Next  
  ‘Завершение цикла по строкам  
Next  
End Sub
```

3. Порядок выполнения Практического задания

Задание состоит из четырех отдельных задач.

Задача 1. Создать и отладить процедуры с использованием условного оператора.

Задача 2. Создать и отладить процедуры циклических алгоритмов для обработки одномерных массивов.

Задача 3. Создать и отладить процедуры циклических алгоритмов для обработки двумерных массивов.

Задача 4. Самостоятельно написать и отладить программу с разветвлением вычислений.

3.1. Выполнение задачи 1

3.1.1. Написать процедуру для вычисления значений функции $Y = \ln X + 3X^2$. Значение X взять из ячейки A2. Результат вычислений записать в ячейку B2. Для этого:

1. Создать таблицу Excel табл.1.

Таблица 1

	A	B
1	X	Y
2		

Так как функция определена только при $X > 0$, следует при вводе $X \leq 0$ вывести сообщение, что вычисление функции невозможно.

2. Введите в ячейку A2 число -1.

3. Для создания программы перейти в Visual Basic for Applications: В MS Office 2003 выполнить команды **Сервис – Макрос – Редактор Visual Basic**. При работе в MS Office 2007 команды **Разработчик – Microsoft Visual Basic**. Откроется окно редактора Visual Basic. В левой части окна редактора Visual Basic вы видите окно проектов (рис.1). Если этого окна нет, нажмите комбинацию клавиш Ctrl +R.

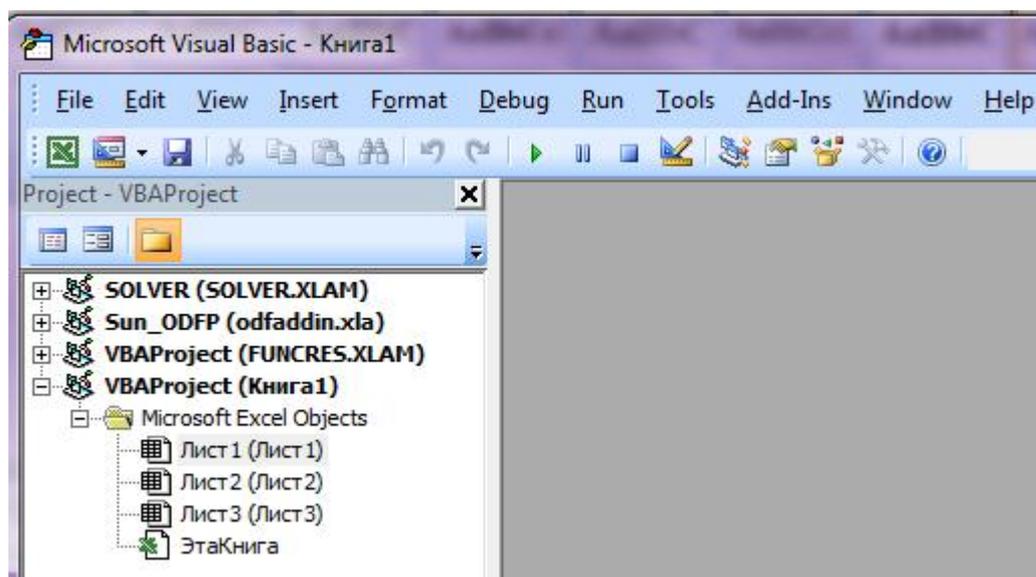


Рис. 1

4. Для создания своего модуля программы в главном меню редактора Visual Basic (строчка вверху окна) выбрать команды **Вставка – Модуль** (если версия не русифицирована, команды **Insert – Module**).

В ваше приложение будет добавлен **Модуль1** (рис.2).

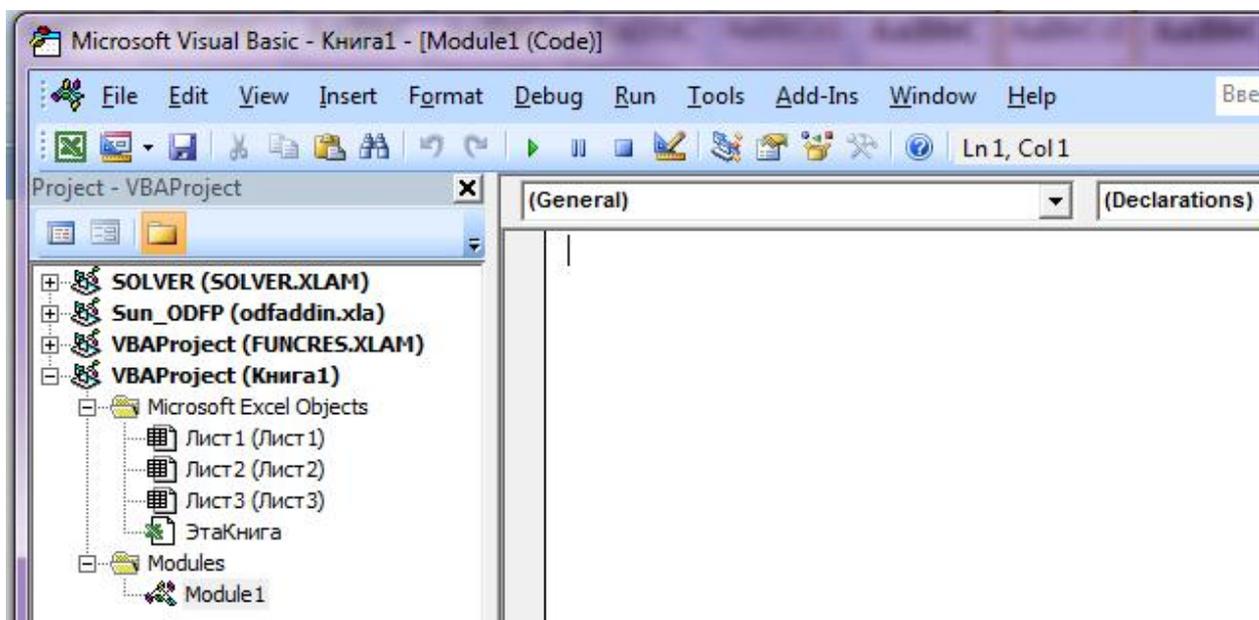


Рис. 2

5. Для создания программы–процедуры выполните команды: **Вставка - Процедура (Insert – Procedure)**. Откроется диалоговое окно **Добавить процедуру** (рис.3). В некоторых версиях Excel это окно называется **Вставка процедуры**.



Рис. 3

6. Введите **Имя (Name)** процедуры: Логарифм.

В секции **Типе (Тип)** установите переключатель **Sub (Подпрограмма-процедура)**. Щелкните по кнопке **Ок** - в модуль добавляется новая процедура.

7. В пустой строке (после заголовка **Public Sub Логарифм ()**) введите текст программы.

```
Public Sub Логарифм ()  
Dim X,Y As Single  
    X = Range("A2").Value  
    If X<= 0 Then  
        MsgBox("Функция не определена")  
    Else  
        Y = Log(X)  
        Range("B2").Value = Y  
    End if  
End Sub
```

8. Для запуска программы на выполнение нажмите клавишу F5. Появится диалоговое окно рис. 4.

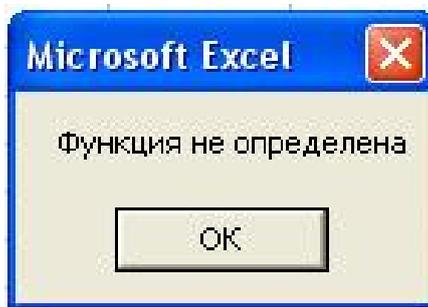


Рис.4

9. Для того, чтобы закрыть это окно, щелкнуть по Ок.
10. Перейти в Excel, щелкнув по кнопке Ms Excel на панели задач.
11. Ввести в ячейку A2 число 1.
12. Перейти в VBA, одновременно нажав клавиши Alt + F11.
13. Нажать клавишу F5 для повторного запуска программы на выполнение.
14. Для контроля вычислений перейти в Excel, щелкнув по кнопке Ms Excel на панели задач. Убедиться, что в ячейке B2 находится число 0.

3.1.2. Вычислить

$$y = \begin{cases} 7x, & \text{если } x < -2 \\ x^2, & \text{если } -2 \leq x \leq 5 \\ x + 4, & \text{если } x > 5 \end{cases}$$

Значение x разместить в ячейке C3 электронной таблицы, значение y разместить в ячейке D1.

Для этого:

1. Повторить пп. 3-5 предыдущего задания для создания новой процедуры.
2. Создать процедуру:

```

Public Sub ТриВетви ( )
Dim x, y As Single
    x = Range("C3").Value
    If x < -2 Then
        y = 7*x
    Elseif x <= 5 Then
        y = x^2
    Else
        y = x +4
    End If
    Range("D1").Value = y
End Sub

```

3. Изменяя значения x , проверить все ветви программы (аналогично пп. 8-14 предыдущего задания).

3.2. Выполнение задачи 2

3.2.1. Найти произведение элементов массива $M(8)$, размещенного в ячейках A4:H4 листа Excel. Результат поместить в ячейку B5.

Для этого:

1. Создать в Excel таблицу «Произведение элементов массива» (табл.1).

Таблица 1

	A	B	C	D	E	F	G	H
1	ПРОИЗВЕДЕНИЕ ЭЛЕМЕНТОВ МАССИВА							
2	Элементы массива							
3	M(1)	M(2)	M(3)	M(4)	M(5)	M(6)	M(7)	M(8)
4	2	-1	4	-3	1	-2	-1	2
5	Произведение элементов массива							

2. Создать процедуру.

```

Public Sub Произведение( )
Dim ПроизвМассива As Integer
Dim I As Integer
    ПроизвМассива = 1
    For I = 1 To 8
        ПроизвМассива = ПроизвМассива * Cells(4,I)
    Next
    Range("B5").Value = ПроизвМассива
End Sub

```

Обратите внимание! Если имена ячеек в Excel заранее не известны, для обращения к ним используется не ключевое слово *Range*, а ключевое слово *Cells*!

3. Запустить программу на выполнение, проверить результат в ячейке B5 (число 96).

3.2.2. Составить программу вычисления суммы отрицательных элементов массива R(7), размещенного в ячейках B5:H5 листа Excel. Результат поместить в ячейку E8.

Для этого:

1. Создать в Excel таблицу «Сумма отрицательных» (табл. 2).

Таблица 2

	A	B	C	D	E	F	G	H
1	СУММА ОТРИЦАТЕЛЬНЫХ							
2								
3	<i>Элементы массива</i>							
4		R(1)	R(2)	R(3)	R(4)	R(5)	R(6)	R(7)
5		12	-7	-2	18	24	-6	0
6								
7				Результат				
8								

2. Создать процедуру:

```
Public Sub СуммаОтрицательных()
```

```
    Dim СуммаМинус, I, R(7) As Integer
```

```
    ‘Цикл переписывает значения массива из таблицы в программу
```

```
    For I = 2 To 8
```

```
        R(I) = Cells(5,I)
```

```
    Next
```

```
    ‘Вычисление суммы отрицательных элементов
```

```
        СуммаМинус = 0
```

```
        For I = 1 To 8
```

```
            If R(I) < 0 Then
```

```
                СуммаМинус = СуммаМинус +R(I)
```

```
            End If
```

```
        Next
```

```
        Range(“E8”).Value = СуммаМинус
```

```
End Sub
```

3. Запустить программу на выполнение, проверить результат в E8.

3.3. Выполнение задачи 3

Создать процедуру для выполнения следующих действий:

1. Записать в массив P(2,5) данные, хранящиеся в ячейках A1:E2 листа Excel с именем Первый.

2. Найти сумму элементов каждой строки массива и записать в массив с именем S.

3. Вывести окно с сообщением «Сумма» I-й «строки равна» S(I).

Для этого:

1. Открыть новый файл;
2. Переименовать Лист1 в Первый (щелчок правой клавишей по Лист1 – Переименовать – ввести имя Первый).
3. Заполнить ячейки A1:E2 значениями

11	5	-1	4	8
4	0	-9	-8	2

4. Создать процедуру:

```
Public Sub ДвумерныйМассив( )  
Dim P(2, 5), S(2) As Integer
```

```
Dim I, J As Integer
```

```
Dim Первый As Object
```

```
Set Первый = Worksheets(“Первый”)
```

```
‘Переписываем массив из таблицы
```

```
For I = 1 To 2
```

```
For J = 1 To 5
```

```
P(I,J) = Задание.Cells(I,J)
```

```
Next
```

```
Next
```

```
‘Считаем суммы элементов по строкам
```

```
For I = 1 To 2
```

```
S(I) = 0
```

```
For J = 1 To 5
```

```
S(I) = S(I) + P(I,J)
```

```
Next
```

```
MsgBox(“Сумма” & I & “-й строки равна” & S(I))
```

```
Next
```

```
End Sub
```

5. Сохранить файл с именем «Задание3».

6. Запустить программу на выполнение и проверить результат счета (сумма первой строки равна 27, а второй -11).

3.4. Выполнение задачи 4

Написать и отладить программу с разветвлением вычислений согласно индивидуальному заданию. Номер варианта выбирается по последней цифре шифра студента.

Индивидуальные задания

Вариант 1. Составить программу вычисления $y = \begin{cases} x - 5, \text{ если } : x < 0 \\ x^2, \text{ если } : 0 \leq x < 7. \\ \sqrt{x}, \text{ если } : x \geq 7 \end{cases}$

Значение x прочитать с листа Excel из ячейки A1. Результат поместить в ячейку A2.

Вариант 2. Составить программу вычисления

$$y = \begin{cases} x + 10, \text{ если } : x < 0 \\ x^2, \text{ если } : 10 \leq x < 27. \\ \sqrt{x}, \text{ если } : x \geq 27 \end{cases}$$

Значение x прочитать с листа Excel из ячейки B2. Результат поместить в ячейку A4.

Вариант 3. Составить программу вычисления

$$y = \begin{cases} x^4, \text{ если } : x \leq -20 \\ x + 10, \text{ если } : -20 < x \leq -10 \\ \sin(x), \text{ если } : -10 < x \leq 0 \\ x^3 + x, \text{ если } : 0 < x \leq 5 \\ 9x, \text{ если } : x > 5 \end{cases}$$

Значение x прочитать с листа Excel из ячейки C3. Результат поместить в ячейку B2.

Вариант 4. Составить программу вычисления

$$y = \begin{cases} x^2, \text{ если } : x \leq 0 \\ x - 3, \text{ если } : 0 < x \leq 4 \\ \ln(x), \text{ если } : 4 < x \leq 8 \\ e^x + 5x, \text{ если } : 8 < x \leq 15 \\ \cos x, \text{ если } : x > 15 \end{cases}$$

Значение x прочитать с листа Excel из ячейки A5. Результат поместить в ячейку C2.

Вариант 5. Составить программу начисления премии сотрудникам, перевыполнившим норму: на 15% - премия 100р.; на 25% - премия 200р.; на 35% - премия 300р.; на 45% - премия 400р., если >45% - премия 0,5 оклада. Исходные данные (%) прочитать с листа Excel из ячейки B3, а результат (размер премии) вывести диалоговым окном.

Вариант 6. Составить программу вычисления скидок на товары:

стоимостью более 40 000р. – скидка 7%;

стоимостью от 30 000р. до 35 000р. – скидка 10%;

стоимостью от 20 000р. до 25 000р. – скидка 12%;

стоимостью от 10 000р. до 15 000р. – скидка 13%.

Исходные данные (стоимость товара) прочитать с листа Excel из ячейки B3, а результат (скидку) вывести диалоговым окном.

Вариант 7. Составить программу выпуска прохладительных напитков в зависимости от времени года:

Зима - 2,5 млн. литров; Весна – 3,5 млн. литров; Лето – 5 млн. литров; Осень – 3 млн. литров.

Исходные данные (время года) прочитать из ячейки A1 листа Excel, а количество напитков вывести диалоговым окном.

Вариант 8. Составить программу начисления стипендии студенту, в зависимости от среднего балла:

Если студент набрал $> 4,5$ балла, то получает 1750 руб. Если средний балл находится в пределах $[3,5; 4,5]$, то 1500 руб., в противном случае стипендии нет. Исходные данные (средний балл) прочитать из ячейки B3 листа Excel, а стипендию вывести диалоговым окном.

Вариант 9. Составить программу вычисления

$$y = \begin{cases} x^3, & \text{если } : x \leq 1 \\ x + 8, & \text{если } : 1 < x \leq 5 \\ \cos(x), & \text{если } : 5 < x \leq 10 \\ x^2 - 2x, & \text{если } : 10 < x \leq 12 \\ 7x, & \text{если } : x > 12 \end{cases}$$

Значение x прочитать с листа Excel из ячейки C8. Результат поместить в ячейку A2.

Вариант 0. Составить программу вычисления

$$y = \begin{cases} e^x, & \text{если } : x \leq -10 \\ |x|, & \text{если } : -10 < x \leq -5 \\ \sin(x), & \text{если } : -5 < x \leq 2 \\ \ln x, & \text{если } : 2 < x \leq 6 \\ x + 5, & \text{если } : x > 6 \end{cases}$$

Значение x прочитать с листа Excel из ячейки D3. Результат поместить в ячейку C2.

4. Отчет по работе

Распечатки процедур и таблиц работы

СЕВЕРО-ЗАПАДНЫЙ ОТКРЫТЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

Методические указания к выполнению практического занятия № 2

Санкт-Петербург
2015

СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ФОРМ

1. Цель работы

Научиться создавать диалоговые окна различного назначения, содержащие элементы управления, которые позволяют обмениваться информацией с приложениями Microsoft Office.

2. Основные теоретические положения

При создании большинства баз данных остро стоит вопрос о заполнении их информацией. Как правило, пользователи этих баз обладают недостаточным уровнем подготовки для прямого обращения к базам данных. Поэтому создают так называемые АРМ – автоматизированные рабочие места для банковских работников, медсестёр, кассиров и так далее. Ввод информации в АРМ осуществляется с помощью специализированных интерфейсов – экранных форм. Создание такого рода интерфейсов требует огромных трудозатрат программистов. Между тем Visual Basic for Applications предоставляет возможность относительно простой разработки такого рода интерфейсов.

3. Порядок выполнения работы

Задание состоит из четырех связанных между собой задач

Задача 1. Создать экранную форму “Прием заказов” для заполнения журнала заказов фирмы.

Задача 2. Провести инициализацию экранной формы, то есть создать процедуру, осуществляющую ввод в форму начальных данных (автоматический ввод даты оформления заявки и заполнение поля со списком **Наименование продукции**).

Задача 3. Создать процедуру для вызова формы.

Задача 4. Обеспечить сохранение вводимых данных на рабочем листе.

Задача 5. Создать процедуру проверки правильности вводимых данных.

3.1. Выполнение задачи 1

Создать экранную форму “Прием заказов” для заполнения журнала заказов фирмы. Для этого:

1. Открыть новую рабочую книгу в Excel.
2. Перейти в редактор Visual Basic (нажать комбинацию клавиш **Alt+F11**).
3. Для вставки формы выполните команды **Вставка – Форма** (для нерусифицированной версии **Insert – UserForm**). В рабочую книгу будет вставлена экранная форма, показанная на рис. 1.

Новая форма представляет собой пустое серое окно со строкой заголовка. Это та основа, на которой пользователь может создать собственные окна любых типов.

4. Установим значения свойств формы: то есть дадим ей имя и введём название в строку заголовка.

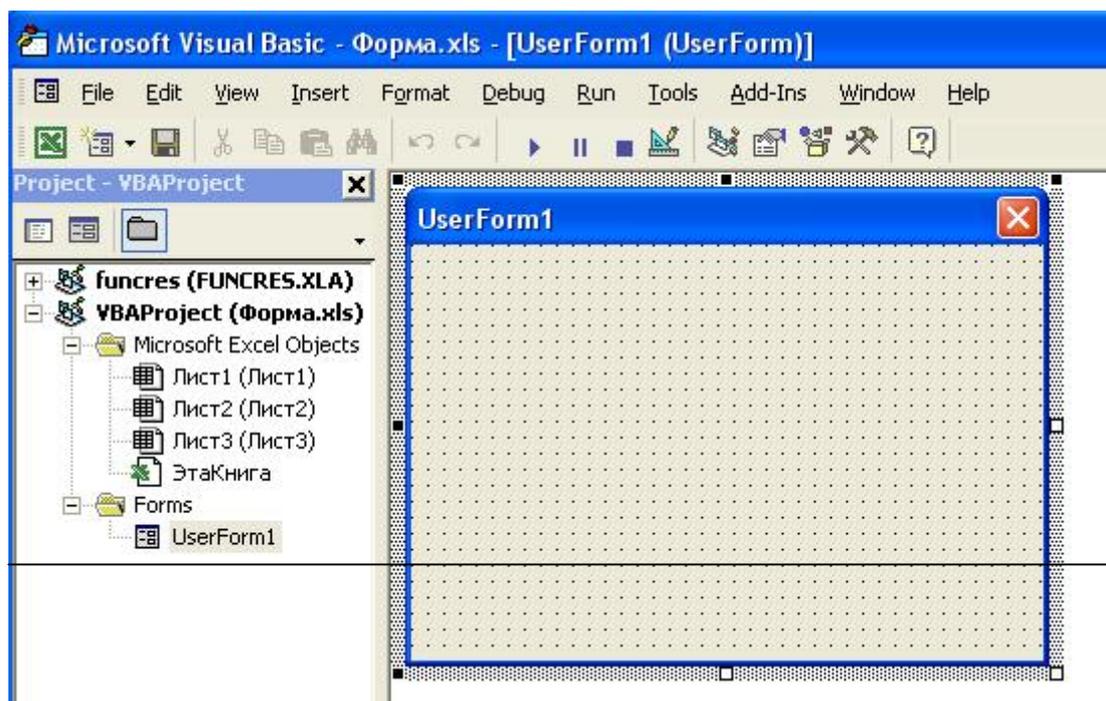


Рис. 1

Для этого выполним команды:

- 4.1. Если окно свойств не отображено на экране, выполнить команды **Вид – Окно** (или нажать клавишу **F4**). Появится **окно свойств** (рис.2).

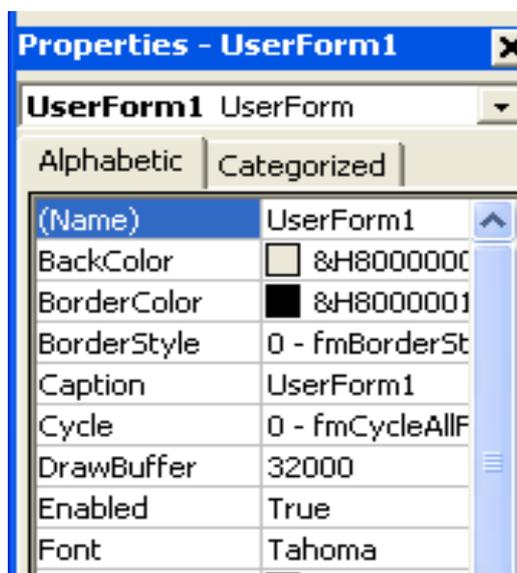


Рис. 2

4.2. Выберите свойство *Name* (Имя), которое находится в верхней части списка, упорядоченного по алфавиту. Щелкнув справа от *Name* в этой строке, введите вместо **UserForm1** новое имя **Заказ**. Мы задали имя, по которому будем обращаться к нашей форме в программе.

4.3. Для ввода заголовка формы выберите свойство *Caption*. (Как видим, Visual Basic при создании новых форм присваивает свойствам *Name* и *Caption* одинаковые названия - **UserForm1**). Щелкнув справа от **UserForm1** в данной

строке, введите **Журнал заказов**. Введённый текст появится в строке заголовка формы (рис. 3).

5. Вставим необходимые элементы управления в экранную форму. Мы хотим получить форму, показанную на рис.3.

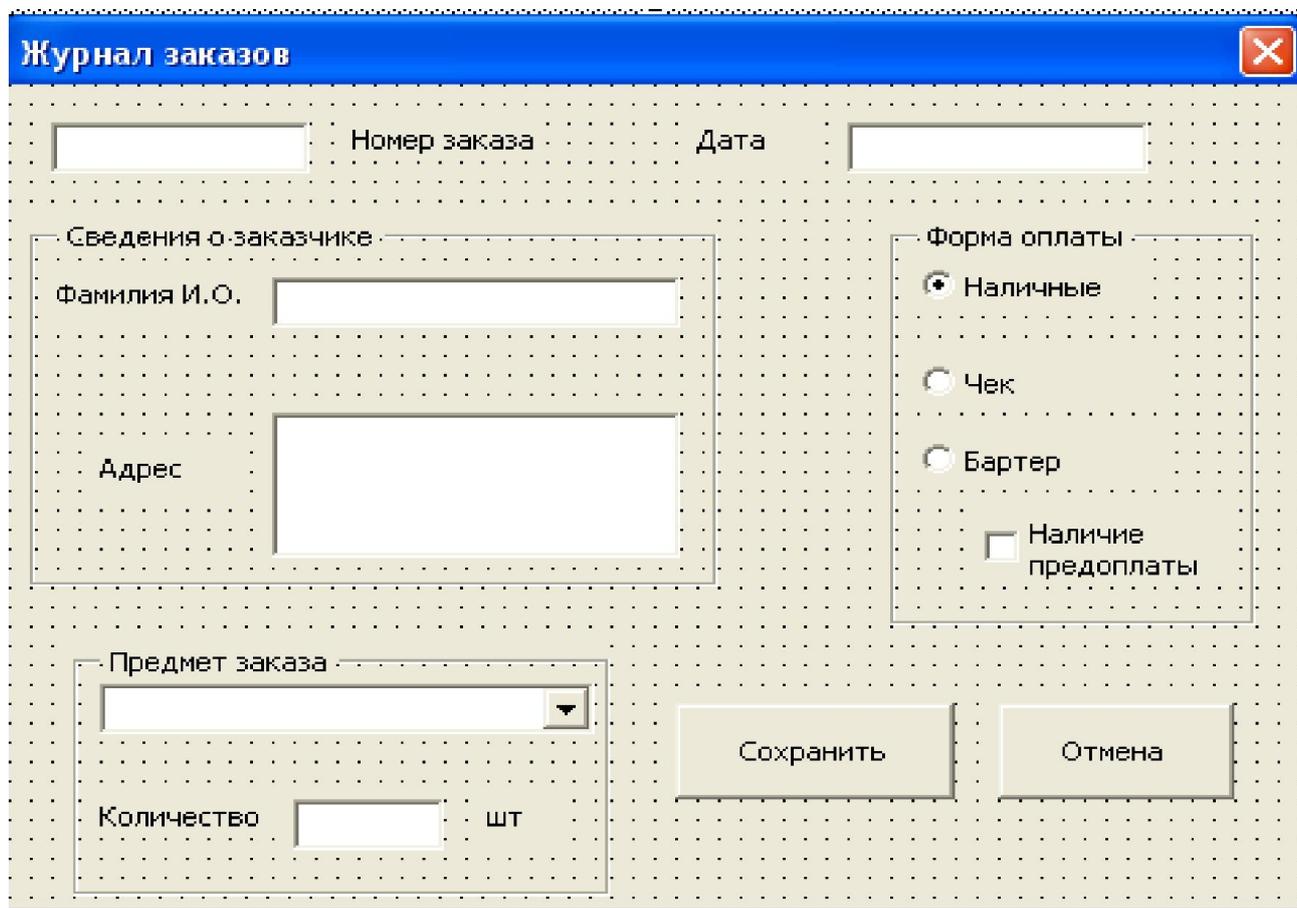


Рис. 3

Как видно, в форме использованы различные элементы управления. Чтобы вставить их в форму, выполните следующие действия:

- 5.1. Если панель элементов не отображена на экране, выполните команды **Вид – Панель элементов (View-Toolbox)**. Появится панель элементов рис 4.

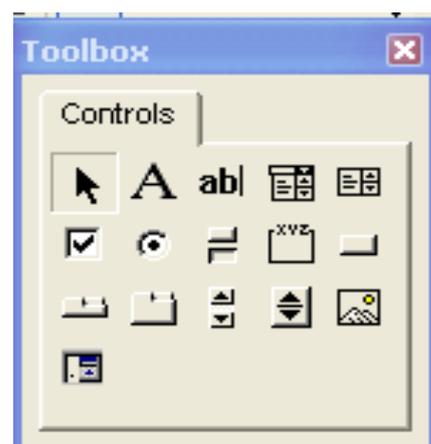


Рис. 4

5.2. На панели элементов щелкните по элементу **Поле (TextBox)**:

Поставив указатель мыши на этот элемент, нажмите левую



клавишу и переместите его в левый верхний угол формы (см. рис.5). Сохраним за полем предлагаемое Visual Basic имя **TextBox1**.

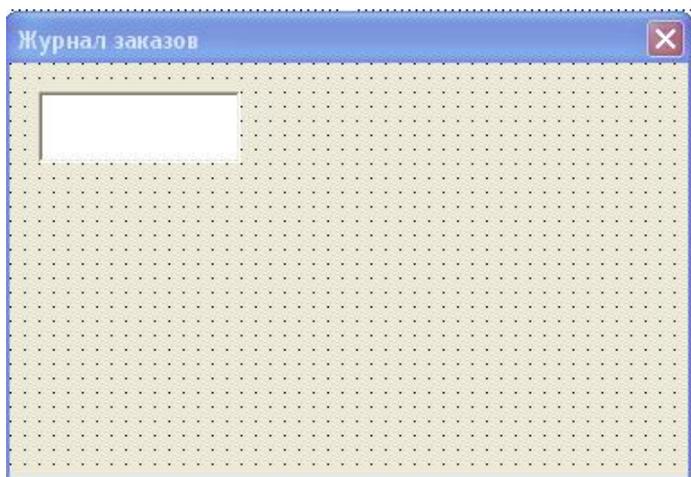


Рис. 5

5.3. На панели элементов выберите элемент **Надпись (Label)**:



Поставив указатель мыши на этот элемент, нажмите левую клавишу и переместите его справа от элемента **Поле** (рис. 6).

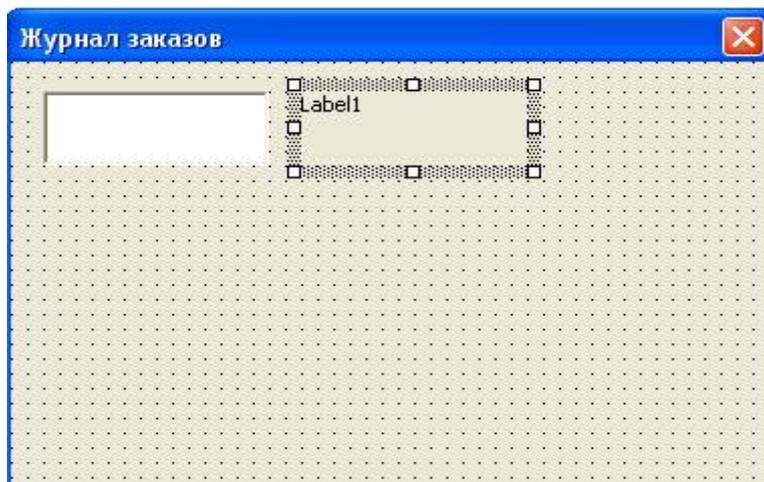


Рис. 6

5.4. Изменим заголовок элемента **Надпись** с *Label1* на **Номер заказа**. Если окно свойств элемента **Надпись** скрыто, нажмите клавишу **F4**. Задайте свойству **Caption** значение **Номер заказа** (аналогично п. 4.3). При этом на форме вместо слова **Label1** появится надпись «**Номер заказа**». Свойство **Name** оставляем без изменения, то есть в программе за этой надписью закреплено предложенное Visual Basic имя **Label1**.

5.5. Используя рис.3 как образец, продолжайте аналогично п.п. 5.2-5.4 создавать элементы управления. Описание значений свойств этих элементов управления приведены в табл.1.

Замечание 1. В форму необходимо трижды вставить элемент **Рамка**: для объединения сведений о заказчике, о предмете заказа и о форме оплаты.

Если Вы попытаетесь сначала разместить соответствующие элементы, а потом объединить их рамкой, то рамка закроет на форме элементы управления. Поэтому сначала следует разместить на форме рамку, изменить ее заголовок (свойство *Caption*) в соответствии с табл. 3, а затем поместить в рамке другие элементы управления.

Замечание 2. При выборе элементов управления лучше придерживаться того порядка, который указан в табл. 3. В противном случае *Visual Basic* присвоит им имена с другой нумерацией, и при автоматизации формы Вам придется самостоятельно исправлять эту нумерацию в создаваемых в данной работе программах.

Таблица 1

Тип элемента управления	Значение свойства Name	Значение свойства Caption	Другие свойства
Поле	TextBox1	-	-
Надпись	Label1	Номер заказа	-
Надпись	Label2	Дата	-
Поле	TextBox2	-	-
Рамка	Frame1	Сведения о заказчике	-
Надпись	Label3	Фамилия И.О.	-
Поле	TextBox3	-	-
Надпись	Label4	Адрес	
Поле	TextBox4	-	
Рамка	Frame2	Форма оплаты	
Переключатель	OptionButton1	Наличные	Value=True
Переключатель	OptionButton2	Чек	-
Переключатель	OptionButton3	Бартер	-
Флажок	CheckBox1	Наличие предоплаты	Value=False
Рамка	Frame3	Предмет заказа	
Поле со списком	ComboBox1	-	-
Надпись	Label5	Количество	-
Поле	TextBox5	-	
Надпись	Label6	шт	
Командная кнопка	ComandButton1	Сохранить	Default=True
Командная кнопка	ComandButton2	Отмена	Cancel=True

Для каждого элемента мы сохраняем предложенное *Visual Basic* свойство **Name**, для элементов **Надпись**, **Переключатель (Опция)**, **Рамка** и **Флажок** задаём свойство **Caption**. Для ряда элементов управления необходимо также задать дополнительные свойства.

Прокомментируем эти дополнительные свойства.

Свойства командных кнопок

Кнопка Сохранить. Чтобы командная кнопка работала как кнопка **Сохранить**, необходимо задать ей значение свойства **Default** (По умолчанию), равное **True** (Истина). В экранной форме только одна командная кнопка может иметь значение **True** свойства **Default**.

Кнопка Отмена. Щелчок по этой кнопке должен вызывать отмену введённых команд. Чтобы командная кнопка работала как кнопка **Отмена**, следует для неё задать свойству **Cancel** значение **True**. Так же, как и в случае свойства **Default**, в экранной форме только одна командная кнопка может иметь значение **True** свойства **Cancel**.

Свойства переключателей

Для переключателей в группе **Форма оплаты** прежде всего нужно указать, какой из них будет выбран по умолчанию (то есть будет находиться во включённом состоянии до выбора, сделанного пользователем).

Для этого нужно задать значение **True** свойства **Value** (Значение). Установим его для первого переключателя **Наличные**. Отметим, что в группе только один переключатель может иметь значение **True** свойства **Value**.

Свойства флажка

Чтобы в исходном состоянии флажок не был установлен, следует задать равным **False** значение его свойства **Value**.

5.6. Сохранить созданную форму с именем **Журнал заказов**. Для этого:

5.6.1. Перейти в Excel (щелкнув по кнопке Microsoft Excel на панели задач.

5.6.2. **Файл – Сохранить как...** - ввести **Журнал заказов**.

3.2. Выполнение задачи 2

В процедуре инициализации экранной формы должен быть предусмотрен блок задания исходных данных для поля со списком. *Для создания списка* выполните следующие действия:

1. Переименуйте рабочий **Лист2** на **Список**. Для этого:
 - 1.1. Щелкнуть **правой** клавишей мыши по имени **Лист 2**.
 - 1.2. В появившемся контекстном меню выбрать **Переименовать**.
 - 1.3. Ввести с клавиатуры имя **Список**.
2. На рабочем листе **Список** введите исходные данные согласно табл.2.

Таблица 2

	А
1	Предмет заказа
2	Оконная рама
3	Дверь
4	Стол
5	Стул
6	Ванна

3. Для присвоения диапазону ячеек имени **Предмет заказа** выполнить следующее:

- выделить диапазон ячеек **A2:A6**;
- выполнить команды **Вставка – Имя – Присвоить**;
- в открывшемся диалоговом окне ввести имя **ПредметЗаказа** (обратите внимание - имя списка, как и любое имя переменной или программы, пишется без пробелов) ;
- щёлкнуть по кнопке **ОК**.

4. Сохраните рабочую книгу с прежним именем (**Файл – Сохранить**).

5. Нажмите одновременно клавиши **Alt + F11** для перехода в Visual Basic.

6. Дважды щёлкните по экранной форме (не на элементах управления, а именно по форме!). Откроется окно с заголовком процедуры **UserForm_Click** (Щелчок по форме) – рис. 7.

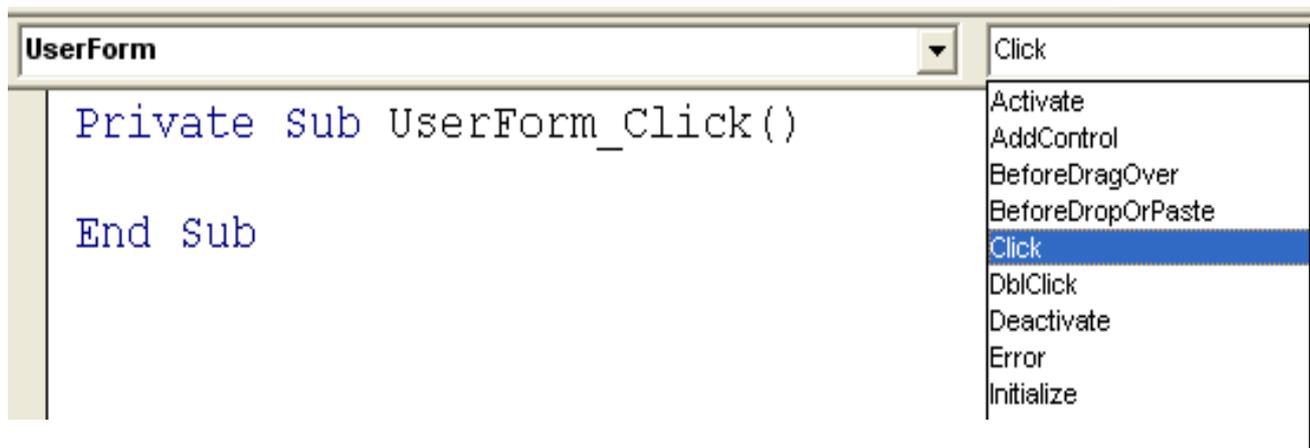


Рис. 7

7. Нам пока не нужна процедура **UserForm_Click**, которая вызывается после щелчка по форме. Нам нужна процедура, которая будет выполняться при открытии формы. На языке программистов она называется процедурой, обрабатывающей событие **Activate** (Активизация). Список процедур, возможных для формы, находится в правой верхней части окна рис. 11.

Щёлкнем по кнопке со стрелкой рядом со командой **Click** и в раскрывшемся списке выберем событие **Activate**. В окне программы появится заголовок процедуры **UserForm_Activate** (рис. 8).

8. Подготовим код процедуры для записи исходных данных с листа *Список* в поля со списками. Сначала приведём текст процедуры с подробными

```
Private Sub UserForm_Activate()  
  
End Sub
```

Рис. 8

комментариями – не нужно его набирать, постарайтесь понять смысл каждого оператора.

```
Private Sub UserForm_Activate()  
    ' присвоить полю со списком "Тема занятий" следующие свойства:  
    With ComboBox1  
        ' исходные данные брать из диапазона ячеек "Предмет заказа"  
        RowSource = "Предмет заказа"  
        ' нумерацию элементов списка из диапазона ячеек начинать с первого  
        ListIndex = 0  
        ' завершить операции с полем списка  
    End With  
,  
  
End Sub
```

Чтобы в поле *Дата* автоматически проставлялось значение текущей даты, перед оператором конца подпрограммы **End Sub** добавим в процедуру **UserForm_Activate** оператор

```
TextBox2 = Format(Now, "dd/mm/yy")
```

9. Теперь вводим в предоставленную нам процедуру *Private Sub UserForm_Activate()* следующий текст программы:

```
Private Sub UserForm_Activate()  
    With ComboBox1  
        .RowSource = "Предмет заказа"  
        .ListIndex = 0  
    End With  
    TextBox2 = Format(Now, "dd/mm/yy")  
End Sub
```

10. Нажать клавишу **F5** для выполнения процедуры. Отобразится экранная форма с заполненным списком *Предмет заказа* и проставленной текущей датой оформления заказа (рис. 9).

Рис. 9

11. Закрывать экранную форму щелчком по



3.3. Выполнение задачи 3

Для создания процедуры вызова экранной формы нужно воспользоваться командой Visual Basic «вывести на экран» – **Show**. Для этого следует выполнить следующие действия:

1. Создать в той же рабочей книге, где находится форма, новый модуль: **Insert - Module (Вставка – Модуль)**.
2. Создать новую процедуру: **Insert - Procedure (Вставка – Процедура)** и назвать её *ЗагрузкаФормы*.
3. Ввести код процедуры.

```
Public Sub ЗагрузкаФормы()  
Заказ.Show  
End Sub
```

4. Перейти в рабочую книгу на **Лист1** и вставить командную кнопку:

- поставить указатель мыши на любую из панелей инструментов и щёлкнуть правой клавишей (появится меню панелей инструментов);
- выбрать панель **Формы** (рис. 10);

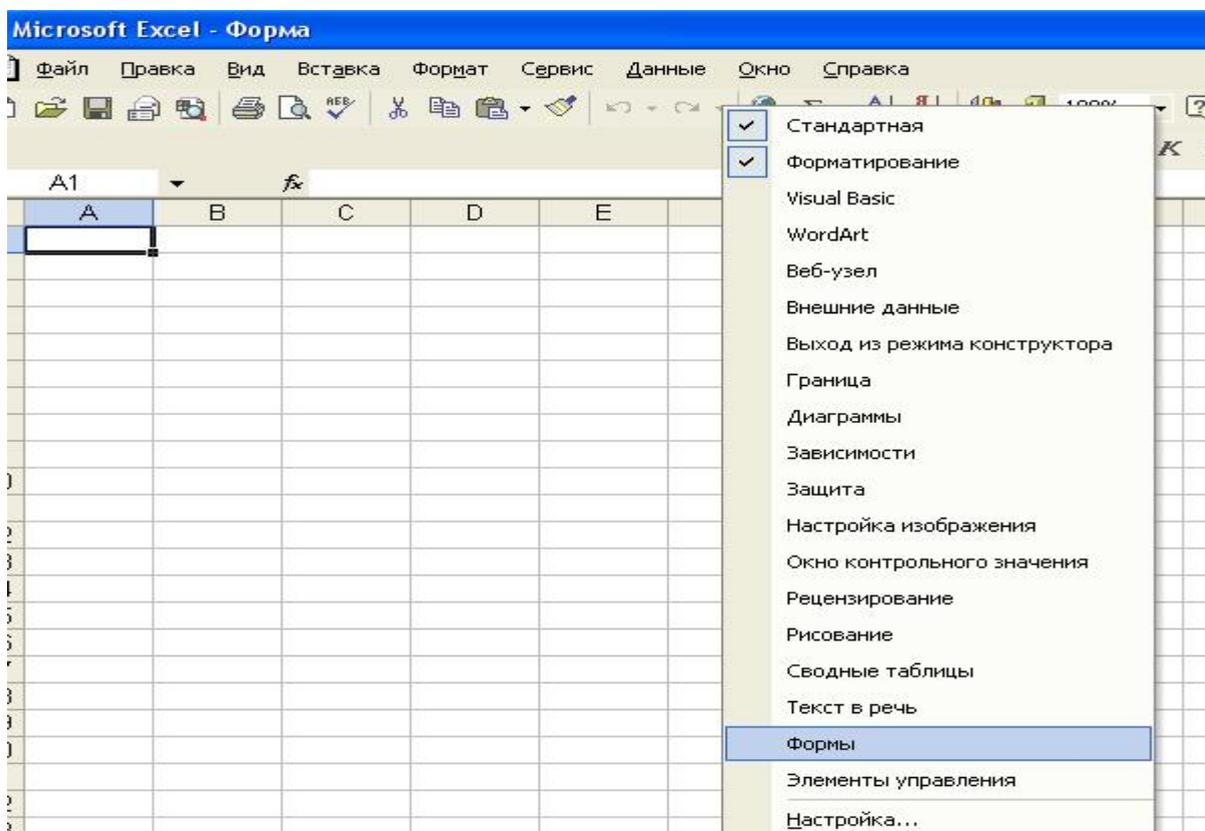


Рис. 10

- на появившейся панели Формы выбрать командную кнопку и перетащить ее с этой панели на лист (рис. 11).

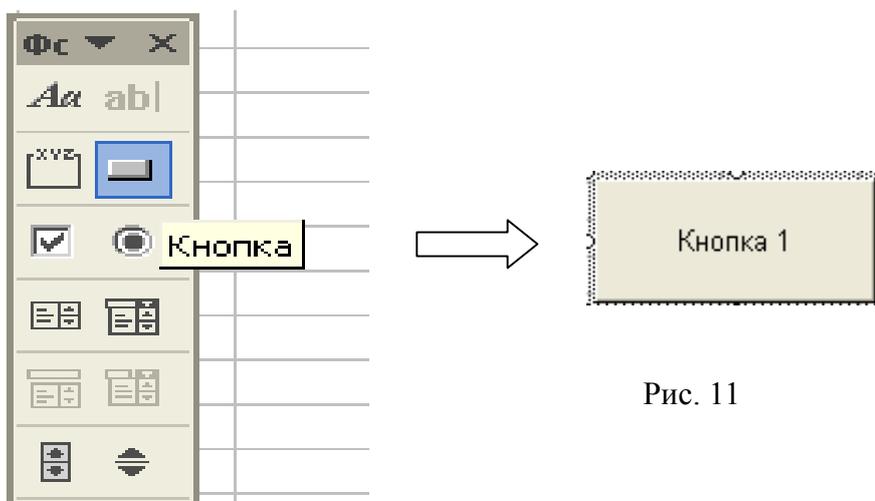


Рис. 11

5. После вставки кнопки откроется диалоговое окно **Назначить макрос объекту** (рис. 12). Выбрать в открывшемся списке **Имя макроса** пункт **ЗагрузкаФормы**.

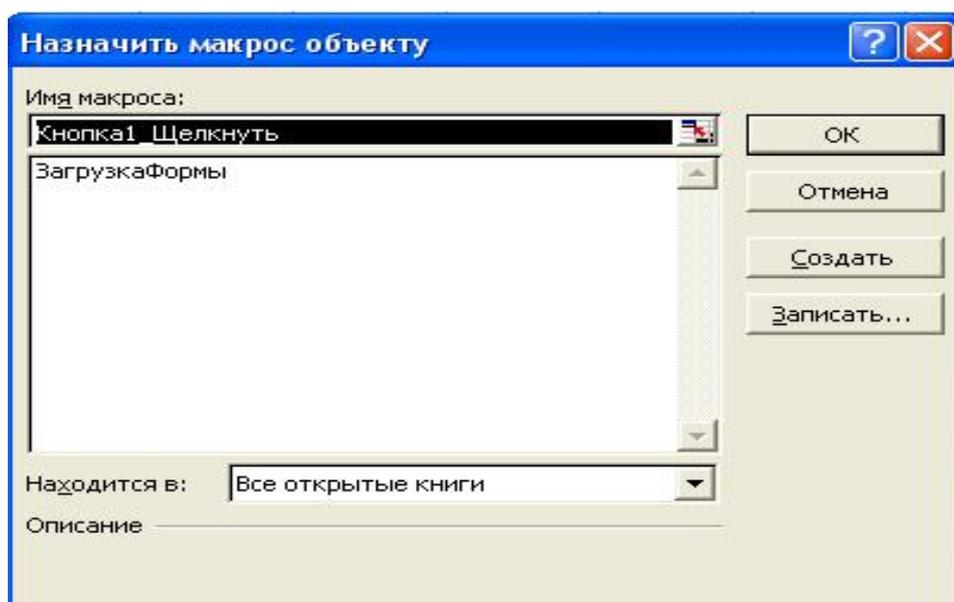


Рис. 12

6. Щёлкнуть по командной кнопке, стереть имя **Кнопка1** (в некоторых версиях Excel **Label1**) и дать кнопке имя **Журнал заказа**. После этого следует щёлкнуть по клеткам таблицы вне кнопки, чтобы снять с неё выделение.

7. Для запуска формы щёлкнуть по этой командной кнопке. Откроется наша форма. Обратите внимание: список **Предмет заказа** заполнен, как и поле **Дата** (рис. 9).

8. Закрыть форму.

3.4. Выполнение задачи 4

Данные, введённые в форму, следует сохранить в базе данных, организованной на рабочем листе книги Excel. Для этого:

1. Перейдите на рабочий Лист 3 и переименуйте его на **Форма** (если в рабочей книге нет **Листа 3**, поставьте указатель мыши на имя **Лист1** и выполните команды **Добавить – Лист**).

2. В первой строке этого листа Excel создайте строку заголовков Журнала заказов. За образец возьмите рабочий лист, показанный на рис.13.

	А	В	С	Д	Е	Ф	Г	Н
1	ЖУРНАЛ ЗАКАЗОВ							
2	Номер заказа	Дата	Фамилия И. О. заказчика	Адрес	Предмет заказа	Количество (шт)	Форма оплаты	Наличие предоплаты
3								

Рис. 13

3. Перейдите в редактор Visual Basic (Alt + F11). Если экранной формы в редакторе VBA нет, для ее отображения дважды щелкните по значку формы **Заказ** в окне проекта (рис. 14).

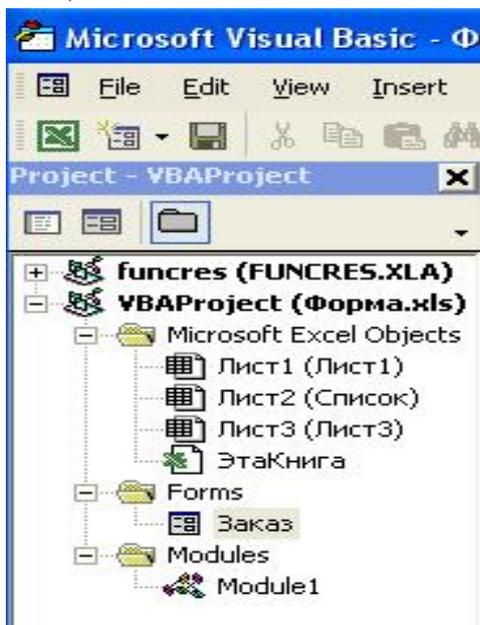


Рис. 14

4. Сохранение информации, введенной в форму, должно осуществляться при щелчке по кнопке **Сохранить**. Для создания такой процедуры дважды щелкните по кнопке **Сохранить**. VBA создает процедуру **CommandButton1_Click** (щелчок по кнопке **Сохранить**).

5. Создаем программу, обеспечивающую сохранение информации. Сначала разбираемся с каждым из операторов. Текст данной программы можно вводить без комментариев.

Private Sub CommandButton1_Click

‘ *Описание листа **Форма** как объекта*

Dim Форма As Object

‘ *Описание переменной **I** для перебора строк в таблице (как целого числа)*

Dim I As Integer

‘ *Описание объекта **Форма** как лист Excel*

Set Форма = Sheets("Форма")

‘ *Активизация листа **Форма***

Worksheets("Форма").Activate

‘ *Организуем цикл просмотра строк в таблице, чтобы найти пустую*

For I = 1 To 100

‘ *Проверка, является ли очередная строка **I** пустой*

If Форма.Cells(I, 1) = " " Then

‘ *Если строка пустая, записать в первый столбец (столбец **A**) этой*

‘ *строки **I** информацию из поля ввода **Номер заказа***

Форма.Cells(I, 1) = TextBox1.Value

‘ *во второй столбец (столбец **B**) записать информацию из поля **Дата***

```

Форма.Cells(I, 2) = TextBox2.Value
‘ в столбец C строки I записать информацию из поля Фамилия И.О.
Форма.Cells(I, 3) = TextBox3.Value
‘ в столбец D строки I записать информацию из поля Адрес
Форма.Cells(I, 4) = TextBox4.Value

‘ Запись в пятый столбец таблицы (столбец E) информации
‘ поля со списком Предмет заказа
Форма.Cells(I, 5) = ComboBox1.Value

‘ Запись в шестой столбец таблицы (столбец F) информации
‘ о количестве заказанной продукции
Форма.Cells(I, 6) = TextBox5.Value

‘ Проверка установки переключателей Форма оплаты. Если установлен
‘ переключатель Наличные, в столбец G (седьмой столбец таблицы)
записать слово «Наличные»
If OptionButton1 = True Then
Форма.Cells(I, 7) = "Наличные"
‘ Проверка установки переключателя Чек
ElseIf OptionButton2 = True Then
Форма.Cells(I, 7) = "Чек"
‘ Проверка установки переключателя Бартер
ElseIf OptionButton3 = True Then
Форма.Cells(I, 7) = "Бартер"
End If

‘ Запись в восьмой столбец таблицы (столбец H) информации
‘ о наличии предоплаты.
If CheckBox1.Value = True Then
Форма.Cells(I, 7) = "Оплата наличными"
End If

‘ Завершение цикла, если найдена пустая строка для записи
Exit For
End If

Next
‘ Команда закрыть форму
Unload Me
End Sub

```

7. Для того, чтобы работала кнопка **Отмена**, дважды щелкните по ней на форме и создайте процедуру:

```

Private Sub CommandBatton2_Click
Unload Me
End Sub

```

8. Перейдите в рабочую книгу на **Лист1**.

10. Сохраните рабочую книгу.

11. Щёлкните по кнопке **Журнал заказа**, вызывающей экранную форму.

Введите (см. рис. 15):

- в поле **Номер заказа** - число **2222**;
- в поле **Фамилия И.О.** - фамилию **Иванов И.И.**;
- в поле **Адрес** - **С.Петербург, ул. Миллионная, д. 5**.
- установите переключатель **Форма оплаты** в положение “**Чек**”;
- установите флажок **Наличие предоплаты** в положение «**Включено**»;
- в поле со списком **Предмет заказа** выберите **Стол**;
- в поле **Количество** введите **10**.

Журнал заказов

2222 Номер заказа Дата 04.04.08

Сведения о заказчике

Фамилия И.О. Иванов И.И.

Адрес С.Петербург, Миллионная, 5

Форма оплаты

Наличные

Чек

Бартер

Наличие предоплаты

Предмет заказа

Стол

Количество 10 шт

Сохранить Отмена

Рис. 15

Щёлкните по кнопке **Сохранить**.

12. Форма закроется, а введённые в неё данные будут записаны в ячейки рабочего листа **Формы**.

13. Перейдите на рабочий лист **Форма** и убедитесь, что информация, введенная в форму, правильно записана в базу данных (рис. 16).

14. Повторите ввод данных в форму (для другого заказа) и убедитесь, что данные о нём будут записаны в следующей строке таблицы (рис. 17).

	A	B	C	D	E	F	G	H
1	ЖУРНАЛ ЗАКАЗОВ							
2	Номер заказа	Дата	Фамилия И. О. заказчика	Адрес	Предмет заказа	Количество (шт)	Форма оплаты	Наличие предоплаты
3	2222	04.04.08	Иванов И.И.	С.Петербург,	Стол	10	Чек	Оплата наличными
4								

Рис. 16

	A	B	C	D	E	F	G	H
1	ЖУРНАЛ ЗАКАЗОВ							
2	Номер заказа	Дата	Фамилия И. О. заказчика	Адрес	Предмет заказа	Количество (шт)	Форма оплаты	Наличие предоплаты
3	2222	04.04.08	Иванов И.И.	С.Петербург, Миллионная, 5	Стол	10	Чек	Оплата наличными
4	3333	04.04.08	Петров П.П.	С.Петербург, Миллионная, 6	Оконная рама	5	Бартер	

Рис. 17

3.5. Выполнение задачи 5

Рассмотрим вопросы правильности ввода информации в форму.

Необходимо проследить, чтобы пользователь не забыл ввести сведения в соответствующие поля ввода: **Номер заказа**, **Фамилия И.О.**, **Адрес**, **Количество**. Для создания такой процедуры необходимо выполнить следующие действия:

1. Дважды щёлкните по кнопке **Сохранить**. В окне кода появится процедура **CommandButton1_Click**.

2. В начале программы, после операторов описания переменных (**Dim** и **Set**) поместим следующий фрагмент программы. Как всегда, сначала разбираемся с операторами, не вводя текст программы.

*Проверяем, не осталось ли пустым поле ввода **Номер заказа***

If TextBox1 = "" Then

‘ *Вывод окна сообщения с просьбой ввести данные о студенте*

MsgBox ("Пожалуйста, введите Номер заказа")

‘ *Установка курсора в поле ввода **Номер заказа***

TextBox1.SetFocus

‘ *Прерывание процедуры для повторного вывода на экран формы*

Exit Sub

‘ *Завершение проверки*

End If

‘ *Проверяем, не осталось ли пустым поле ввода **Фамилия И.О.***

If TextBox3 = "" Then

‘ *Вывод окна сообщения с просьбой ввести данные о заказчике*

MsgBox ("Пожалуйста, введите *Фамилию И.О.* заказчика")

‘ *Установка курсора в поле ввода **Фамилия И.О.***

TextBox3.SetFocus

‘ *Прерывание процедуры для повторного вывода на экран формы*

Exit Sub

‘ *Завершение проверки*

End If

‘ *Проверяем, не осталось ли пустым поле ввода **Адрес***

If TextBox4 = "" Then

‘ *Вывод окна сообщения с просьбой ввести адрес заказчика*

MsgBox ("Пожалуйста, введите адрес заказчика")

‘ *Установка курсора в поле ввода **Адрес***

TextBox4.SetFocus

‘ *Прерывание процедуры для повторного вывода на экран формы*

Exit Sub

‘ *Завершение проверки*

End If

‘ *Проверяем, не осталось ли пустым поле ввода **Количество***

If TextBox5 = "" Then

‘ *Вывод окна сообщения с просьбой ввести заказанное количество изделий*

MsgBox ("Пожалуйста, введите количество заказанной

продукции")

‘ *Установка курсора в поле ввода **Количество***

TextBox5.SetFocus

‘ *Прерывание процедуры для повторного вывода на экран формы*

Exit Sub

‘ *Завершение проверки*

End If

3. Теперь вводим рассмотренный фрагмент программы в процедуру, созданную при выполнении *Задачи 4*. Программа работы с формой приобрела вид (жирным шрифтом выделены вновь введенные фрагменты):

```
Private Sub CommandButton1_Click
```

```
Dim Форма As Object
```

```
Dim I As Integer
```

```
Set Форма = Sheets("Форма")
```

```
If TextBox1 = "" Then
```

```
MsgBox ("Пожалуйста, введите Номер заказа")
```

```
TextBox1.SetFocus
```

```

Exit Sub
End If
If TextBox3 = "" Then
    MsgBox ("Пожалуйста, введите Фамилию И.О. заказчика")
    TextBox3.SetFocus
Exit Sub
End If
If TextBox4 = "" Then
    MsgBox ("Пожалуйста, введите адрес заказчика")
    TextBox4.SetFocus
Exit Sub
End If
If TextBox5 = "" Then
MsgBox ("Пожалуйста, введите количество заказанной продукции")
    TextBox5.SetFocus
Exit Sub
End If
Worksheets("Форма").Activate
For I = 1 To 100
    If Форма.Cells(I, 1) = " " Then
        Форма.Cells(I, 1) = TextBox1.Value
        Форма.Cells(I, 2) = TextBox2.Value
        Форма.Cells(I, 3) = TextBox3.Value
        Форма.Cells(I, 4) = TextBox4.Value
        Форма.Cells(I, 5) = ComboBox1.Value
        Форма.Cells(I, 6) = TextBox5.Value
        If OptionButton1 = True Then
            Форма.Cells(I, 7) = "Наличные"
        ElseIf OptionButton2 = True Then
            Форма.Cells(I, 7) = "Чек"
        ElseIf OptionButton3 = True Then
            Форма.Cells(I, 7) = "Бартер"
        End If
        If CheckBox1.Value = True Then
            Форма.Cells(I, 8) = "Оплата наличными"
        End If
    End If
Exit For
End If
Next
Unload Me
End Sub

```

4. Перейдите в рабочую книгу, **Лист 1** и щёлкните по командной кнопке **Журнал заказа**, открывающей форму.

5. Не вводя никаких данных, щёлкните по кнопке **Сохранить**. Появится окно с сообщением «Пожалуйста, введите Номер заказа».

6. Щёлкните по кнопке **ОК**, чтобы закрыть окно сообщения. На экране наша форма. Курсор находится в поле ввода **Номер заказа**. Введите в это поле следующие данные: **4444** и щёлкните по кнопке **Сохранить**.

7. Появится окно с сообщением «Пожалуйста, введите фамилию заказчика»

8. Щёлкните по кнопке **ОК**, чтобы закрыть окно сообщения. На экране наша форма. Курсор находится в поле ввода **Фамилия И.О.**. Введите в это поле **Сидоров С.С.** и щёлкните по кнопке **Сохранить**.

Аналогичным образом проверьте работу фрагментов программы, отслеживающих ввод информации в поля **Адрес** и **Количество**.

4. Отчет по работе Диалоговая форма и текст программы