

Практическая работа № 6.

Тема: *Решение задач в Excel с помощью VBA. Использование встроенных функций.*

Цель: Научиться выполнять вычисления с использованием функций и операторов VBA, организовывать ввод и вывод данных, научиться выполнять отладку программ.

Время: 40 мин.

Задание: Изучите основные функции VBA для работы с различными типами данных, составьте программу для расчёта краткосрочной ссуды.

- Литература:**
1. Ривкинд Й.Я., Лысенко Т.И., Черникова Л.А., Шакотько В.В. Информатика. 11 класс. Учебник для общеобразовательных учебных заведений. Академический уровень. - К.: Генеза, 2010
 2. Берков Н.А. программирования на VISUAL BASIC: Учебное пособие. - М: МГИУ, 2001.
 3. Воробьева, Ф.И. Приемы программирования в среде VISUAL BASIC for APPLICATION: учебное пособие / Ф.И. Воробьева, Е.С. Воробьев. - Казань: Изд-во Казан. гос. технол. ун-та; 2010.

Интернет-ссылки:

1. http://msk.edu.ua/ivk/Informatika/informatika_2k_2s.html

Последовательность выполнения работы:

1. Включите компьютер, войдите в систему и запустите табличный процессор Microsoft Excel.
2. Переименуйте Лист1 в «Ссуда», сохраните рабочую книгу в папке «Мои документы» в формате «Книга Excel с поддержкой макросов»:

Имя файла:	<input type="text"/>
Тип файла:	Книга Excel с поддержкой макросов (*.xlsm)

В имени файла укажите свою фамилию и номер практической работы, например: «Иванов, пр13».

3. На первом листе («Ссуда») подготовьте таблицу и кнопку-рисунок по образцу:

Ссуда (потребительский кредит)	
Размер ссуды	
Дата выдачи ссуды	
Дата возврата ссуды	
Годовая процентная ставка	
Возвращаемая сумма	
Потери (прибыль банка)	



Рассчитать

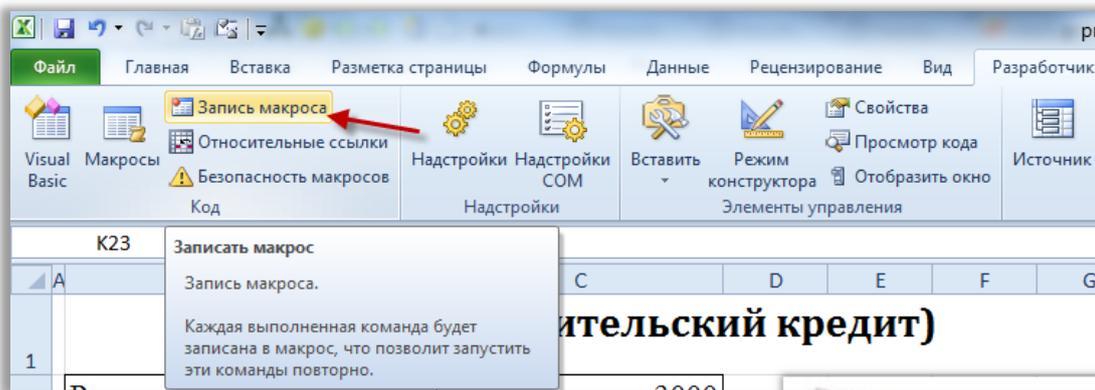
Другие элементы управления (кнопки) добавятся чуть позже.

- Для таблицы используйте шрифт *Times New Roman*, размер – 14;
- Рисунок можно использовать свой, или как в образце, его можно [скачать с сайта](#).

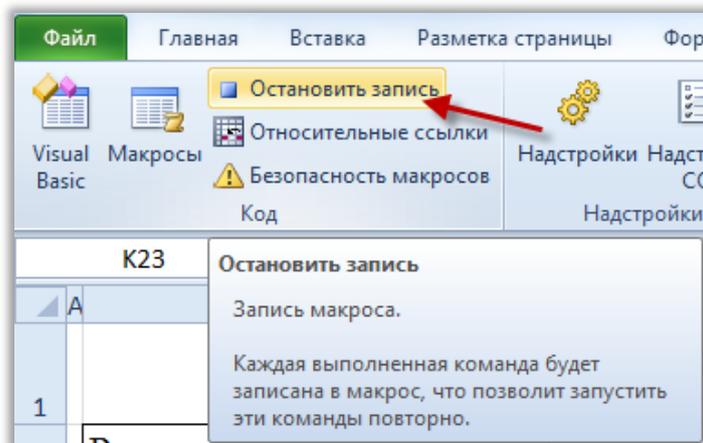
4. Ячейки C2:C7 (см. рисунок ниже) заполните произвольными данными и с помощью макрорекодера создайте макрос, очищающий этот диапазон ячеек. Добавьте кнопку «Очистить» и назначьте этой кнопке созданный макрос.

	A	B	C	D	E	F	G
1	Ссуда (потребительский кредит)						
2		Размер ссуды	3000	 Рассчитать			
3		Дата выдачи ссуды	22.01.2013				
4		Дата возврата ссуды	22.07.2013				
5		Годовая процентная ставка	36				
6		Возвращаемая сумма	3494,15				
7		Потери (прибыль банка)	494,15				
8							
9							
10	Ввод исходных данных			Очистить			
11							
12							

- Для создания макроса воспользуйтесь вкладкой «Разработчик», командой «Запись макроса»:



- Просто нажмите на кнопку и выполняйте нужные действия. Макрорекодер автоматически запишет все действия в виде программного кода.
 – Выделите нужные ячейки и нажмите клавишу «Delete». Переместите курсорную рамку куда-нибудь в сторону и остановите запись.



- После небольшой коррекции у вас должно получиться следующее:

```
Sub Очистить ()
'
' Очистить Макрос
'
    Range ("C2:C7") .ClearContents
End Sub
```

5. Следующая программа берёт данные из таблицы, обрабатывает и записывает результат в ту же таблицу:

```

Sub Ссуда ()
Dim P, R, i As Double
Dim D_beginning, D_end As Date
P = CDBl(Cells(2, 2))
D_beginning = CDate(Cells(3, 2))
D_end = CDate(Cells(4, 2))
i = CDBl(Cells(5, 2)) / 100
If D_end < D_beginning Then
MsgBox "Ошибка в датах", _
vbExclamation, "Расчет ссуды"
Exit Sub
End If
R = P * (1 + i) ^ ((D_end - D_beginning) / 365)
'R = Format(R, "Fixed")
Cells(6, 2) = CStr(R)
End Sub

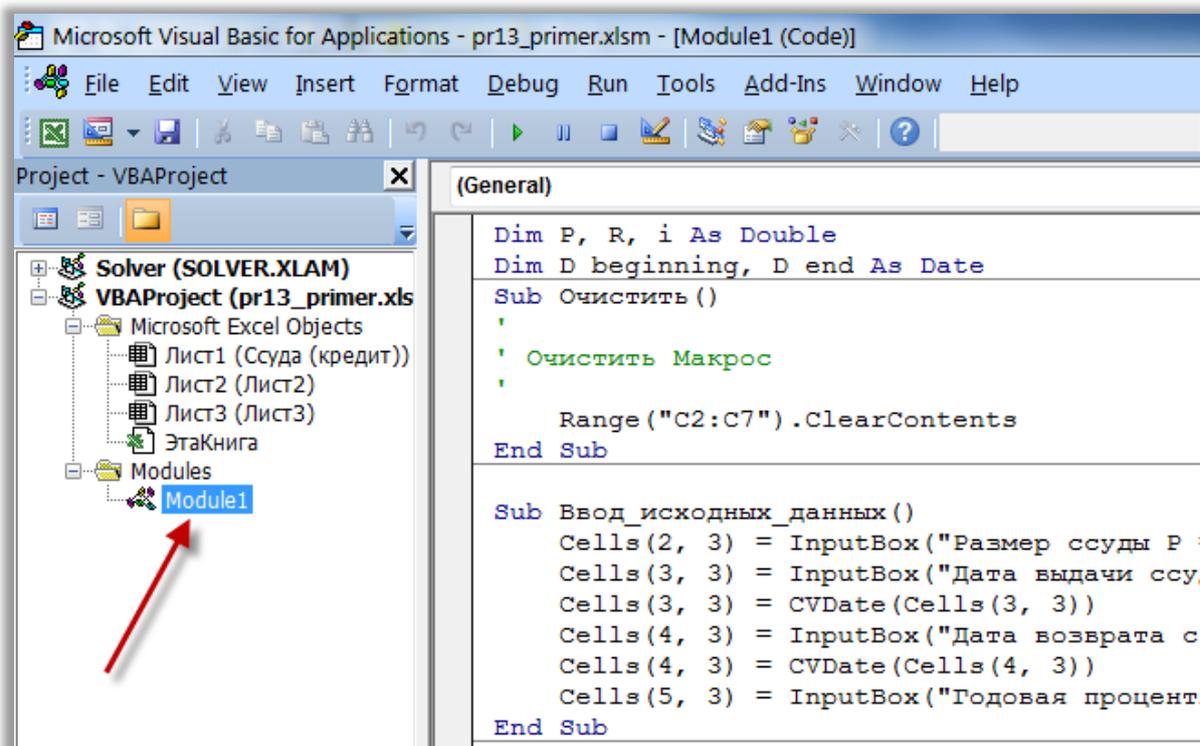
```

Часто для заполнения таблицы используют специальные формы или программный код. Так меньше вероятность занесения ошибочных данных. Для ввода текстовых данных используется функция **InputBox**. Создайте процедуру *Ввод_исходных_данных()* с использованием этой функции.

- Синтаксис функции **InputBox** выглядит так:

InputBox (prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])

- Параметры в квадратных скобках – необязательные.



- Процедура создаётся вручную в окне модуля. Можно создать новый модуль (Insert → Module), а можно использовать и тот, который был создан автоматически макрорекодером (Module1);

- Текст программы (процедуры) записывается внутри блока

Sub <имя_процедуры> () ... End Sub. Сравните вашу процедуру с образцом:

```

Sub Ввод_исходных_данных()
Cells(2, 3) = InputBox("Размер ссуды P = ", "Расчет ссуды", 3000)
Cells(3, 3) = InputBox("Дата выдачи ссуды: ", "Расчет ссуды", "22.01.2013")
Cells(3, 3) = CDate(Cells(3, 3))

```

```
Cells(4, 3) = InputBox("Дата возврата ссуды: ", "Расчет ссуды", "22.07.2013")
```

```
Cells(4, 3) = CVDate(Cells(4, 3))
```

```
Cells(5, 3) = InputBox("Годовая процентная ставка, % ", "Расчет ссуды", 36)
```

```
End Sub
```

– Функция CVDate преобразует текст, содержащий дату, в дату (в тип variant)

6. Добавьте на лист «Ссуда» ещё одну кнопку и назначьте ей макрос «Ввод исходных данных».
7. Составьте программу для расчёта краткосрочной ссуды, полученный макрос свяжите с кнопкой-рисунком на листе «Ссуда» рабочей книги Excel.

Образец:

```
Dim P, R, i As Double
Dim D_beginning, D_end As Date
Sub Очистить()
'
' Очистить, Макрос
'
    Range("C2:C7").Select
    Selection.ClearContents
    Range("C2").Select
End Sub

Sub Ссуда()
    If Cells(2, 3) = Empty Then          'Проверка, заполнена ли ячейка
        MsgBox "Не введена сумма ссуды", vbExclamation, "Расчет ссуды"
        Exit Sub
    End If

    P = CDbI(Cells(2, 3))
    D_beginning = CDate(Cells(3, 3))
    D_end = CDate(Cells(4, 3))
    i = CDbI(Cells(5, 3)) / 100

    If D_end < D_beginning Then
        MsgBox "Ошибка в датах", _
            vbExclamation, "Расчет ссуды"
        'Если оператор слишком длинный,
        'можно разместить его в двух или более строках,
        'добавив в конце каждой из строк
        '(кроме последней) символ подчеркивания (_)

        Exit Sub
    End If
    R = P * (1 + i) ^ ((D_end - D_beginning) / 365)
    R = CDbI(Format(R, "Fixed"))
    Cells(6, 3) = R
    Cells(7, 3) = R - P: Cells(7, 3).Font.Bold = True
    'Можно сделать и наоборот —
    'разместить несколько операторов в одной строке программного кода.
    'Для этого применяется знак :
    Debug.Print "D_beginning = "; D_beginning
    Debug.Print "D_end = "; D_end
    Debug.Print "D_end - D_beginning = "; (D_end - D_beginning)
    Debug.Print "i = "; i
    Debug.Print "R = "; R

End Sub
```

- Этот текст (из образца) можно просто скопировать в модуль VBA Excel. Транслятор VBA проверит код и преобразует его к виду:

```

Sub Ссуда()
  If Cells(2, 3) = Empty Then 'Проверка, заполнена ли ячейка
    MsgBox "Не введена сумма ссуды", vbExclamation, "Расчет ссуды"
    Exit Sub
  End If

  P = CDb1(Cells(2, 3))
  D_beginning = CDate(Cells(3, 3))
  D_end = CDate(Cells(4, 3))
  i = CDb1(Cells(5, 3)) / 100

  If D_end < D_beginning Then
    MsgBox "Ошибка в датах", _
      vbExclamation, "Расчет ссуды" 'Если оператор слишком длинный,
                                     'можно разместить его в двух или более строках,
                                     'добавив в конце каждой из строк
                                     '(кроме последней) символ подчеркивания (_)

    Exit Sub
  End If

  R = P * (1 + i) ^ ((D_end - D_beginning) / 365)
  R = CDb1(Format(R, "Fixed"))
  Cells(6, 3) = R
  Cells(7, 3) = R - P: Cells(7, 3).Font.Bold = True 'Можно сделать и наоборот –
  'разместить несколько операторов в одной строке программного кода.
  'Для этого применяется знак :
  Debug.Print "D_beginning = "; D_beginning
  Debug.Print "D_end = "; D_end
  Debug.Print "D_end - D_beginning = "; (D_end - D_beginning)
  Debug.Print "i = "; i
  Debug.Print "R = "; R

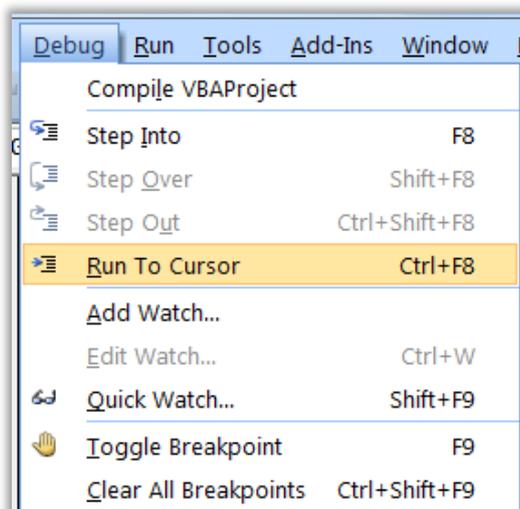
End Sub

```

- Синим цветом выделяются служебные слова VBA, встроенные функции, стандартные операторы и инструкции языка.

Debug.Print служит для отображения значения выражения в специальном окне **Immediate** в процессе выполнения программы.

Выводить в это окно можно любые допустимые выражения, включая свойства объектов. Для того чтобы вывести значение переменной, нужно ввести знак "?" и имя переменной и нажать клавишу <Enter>. Значение переменной отобразится на следующей строке. При этом программа должна выполняться в режиме отладки:



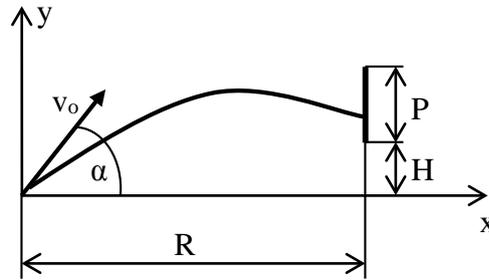
8. Решите задачи по вариантам.

Варианты заданий:

1. На плоскости расположена окружность радиуса R с центром в начале координат. Ввести заданные координаты точки и определить, лежит ли она на окружности. Результат присвоить символьной переменной. Решить задачу при $R=2$ для точек с координатами $(0; 2)$, $(-1.5; 0.7)$, $(1; 1)$, $(3; 0)$.
Считать, что точка с координатами x, y лежит на окружности радиуса R , если $\left| \sqrt{x^2 + y^2} - R \right| \leq e$, где e – точность, с которой осуществляется проверка на равенство (можно принять $e = 10^{-5}$).
2. Заданы координаты двух точек. Определить, лежат ли они на одной окружности с центром в начале координат. Результат присвоить символьной переменной. Задачу решить для точек с координатами $(0; 2)$, $(2; 0)$; $(1; 3)$, $(2, 2)$. Проверку на равенство осуществлять с точностью $e = 10^{-5}$.
3. Заданы площади круга S_1 и квадрата S_2 . Определить, поместится ли квадрат в круге. Задачу решить при 1) $S_1 = 70$, $S_2 = 36.74$; 2) $S_1 = 0.86$, $S_2 = 0.64$
4. Заданы площади круга S_1 и квадрата S_2 . Определить, поместится ли круг в квадрате. Задачу решить при 1) $S_1 = 3.2$, $S_2 = 3.5$; 2) $S_1 = 3.2$, $S_2 = 4$; 3) $S_1 = 6$, $S_2 = 9$.
5. В киоске продаётся газета стоимостью 40 копеек и журнал стоимостью 1 грн. 50 коп. Составить программу, которая спрашивает о желании покупателя (журнал или газета?), принимает деньги (сумма денег вводится с клавиатуры), печатает причитающуюся сдачу или выдаёт сообщение о недостаточности полученной суммы. Исходные данные задать самостоятельно.
6. Составить программу, контролирующую знание закона Ома. Обучаемый вводит формулу закона Ома в строковую переменную, которая далее сравнивается с правильным ответом, хранящимся в строковой константе.
(Совет: перед вводом формулы на экране должно появиться точное указание о том, в каком виде и с использованием каких обозначений нужно осуществлять ввод формулы. Например, такое: «Формулу вводить без пробелов. Использовать обозначения: I – сила тока, U – напряжение, R – сопротивление. Выразить I через U и R .»)
7. Даны числа a_1, b_1, c_1 и a_2, b_2, c_2 . Определить точки пересечения прямых, описываемых уравнениями $a_1x + b_1y = c_1$ и $a_2x + b_2y = c_2$, либо сообщить, что прямые совпадают, не пересекаются или вовсе не существуют. Исходные данные задать самостоятельно.
8. Даны произвольные числа a, b и c . Определить, можно ли построить треугольник с такими длинами сторон. Если треугольник прямоугольный, равнобедренный или равносторонний, то сообщить об этом. Исходные данные ввести с клавиатуры.
9. Начав тренировки, спортсмен в первый день пробежал 10 км, каждый последующий день он увеличивал дневную норму на 10% от нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?
10. Траектория снаряда, вылетающего из орудия под углом α с начальной скоростью v_0 , описывается уравнениями

$$x = v_0 \cdot \cos \alpha \cdot t,$$

$$y = v_0 \cdot \sin \alpha \cdot t - gt^2/2,$$
 где $g = 9.8 \text{ м/с}^2$ – ускорение свободного падения, t – время. Вводя n заданных пар v_0 и α , определить, сколько снарядов поразит цель высотой P , расположенную в вертикальной плоскости ствола орудия на расстоянии R на высоте H (см. рис.). Задачу решить при $R = 5400 \text{ м}$; $H = 100 \text{ м}$; $P = 30 \text{ м}$ для пяти пар v_0 и α : 180 м/с и 45° ; 252 м/с и 30° ; 280 м/с и 40° ; 233 м/с и 48° ; 260 м/с и 25° .
11. Составить программу, печатающую таблицу значений градусов температуры по Цельсию и Фаренгейту. Значения градусов температуры по Цельсию изменяются от 0° до 20° с шагом 1° . ($^\circ\text{F} = ^\circ\text{C} \cdot 1.8 + 32$)



12. Написать программу выбора наименьшего из трёх чисел. Исходные данные задать самостоятельно. Числа должны вводиться с клавиатуры. Результат вывести в виде: «Наименьшее из чисел ..., ..., ... - число ...».
13. Написать программу выбора наибольшего из трёх чисел. Исходные данные задать самостоятельно. Числа должны вводиться с клавиатуры. Результат вывести в виде: «Наибольшее из чисел ..., ..., ... - число ...».
14. Написать программу вычисления значения функции

$$y = \begin{cases} x^2 + 1, & \text{если } x < 0; \\ x - 2.1, & \text{если } x > \pi/2; \\ \sin x, & \text{если } 0 \leq x \leq \pi/2; \end{cases}$$

15. Боковое ребро правильной треугольной пирамиды равно a и составляет с плоскостью основания угол α . В эту пирамиду вписан цилиндр с квадратным осевым сечением. (Основание цилиндра лежит в плоскости основания пирамиды). Объём этого цилиндра можно найти по формуле:

$$V = \frac{\pi \cdot a^3 \sqrt{2} \sin^3 2\alpha}{128 \sin^3 \left(\frac{\pi}{4} + \alpha\right)}$$

Определить боковое ребро a пирамиды, если угол $\alpha = 1,2$ радиана, $V = 40 \text{ см}^3$.

Высокий уровень (дополнительное задание):

9. Составьте программу для расчёта потребительского кредита, предлагаемого Приват-банком:
- 1) Первые 55 дней – беспроцентный кредит;
 - 2) Ставка – 30% годовых;
 - 3) Ежемесячное обязательное погашение (после 55 дней) – не менее 7% от оставшейся суммы (2,5% - банку, остальное – погашение долга).
 - 4) 30-го числа каждого месяца – день расчёта, когда начисленные проценты добавляются к сумме долга. А начисляются они каждый день на ту сумму, которую должен клиент. Их можно рассчитать по формуле

$$R = P * (1 + i)^{((D_end - D_begin) / 365)},$$
 где $i = 30$ (% ставка),
 P - долг, взятая сумма,
 D_end - 30-е число,
 D_begin - 30 число предыдущего месяца или дата взятия кредита.

Если происходит платёж, этот интервал делится на два, в формуле для второго интервала вместо P будет фигурировать $P - P_1$, где P_1 – сумма платежа.

Рассчитайте, сколько вы заплатите сверх взятой на один год суммы, если взяли 8000 гривен в начале года и погашаете кредит ежемесячно, в начале каждого месяца, начиная со второго (скажем, 10-го числа), путём перечисления фиксированной суммы 700 гривен. Когда вы полностью выплатите кредит? Каким будет последний платёж?

Краткая справка:

Применение операторов ветвления в VBA.

Ветвление реализуется с помощью оператора **IF** и инструкции **Select Case**.

Оператор ветвления **IF**.

Блочные структуры **If** – наиболее эффективные из структур логического ветвления. Их можно применять для создания логических структур практически любой сложности.

В общем виде эта структура выглядит следующим образом:

```
IF логическое_выражение THEN
    блок операторов
END IF
```

Когда *логическое выражение* принимает значение *истина* (true), то выполняется блок операторов, заключенный между операторами **IF** и **END IF**. Если *логическое выражение* принимает значение *ложь* (false), то этот блок не выполняется и управление переходит к оператору, следующему за оператором **END IF**.

Структура **IF** может содержать оператор **ELSE**:

```
IF логическое_выражение THEN
    блок операторов
ELSE
    блок операторов
END IF
```

Управление переходит к блоку операторов, следующему за **ELSE** в том случае, если *логическое выражение* принимает значение *ложь* (false).

Это был пример одноблочной структуры оператора **IF**. Одноблочная структура оператора **IF** преобразуется в многоблочную путем добавления оператора **ElseIf**. В этом случае структура выглядит следующим образом:

```
IF логическое_выражение_1 THEN
    блок операторов 1
ElseIf логическое_выражение_2 THEN
    блок операторов 2
ElseIf логическое_выражение_3 THEN
    блок операторов 3
ELSE
    блок операторов4
END IF
```

Когда встречается многоблочная структура **IF, VB** определяет значение логического выражения 1. Если оно равно **true**, то выполняется блок операторов 1, если **false**, то блок операторов 1 полностью пропускается и проверяется значение *логического выражения* 2 и т.д. Если ни одно из поставленных условий не выполняется, то управление передается блоку операторов 4.

В приведенном ниже примере показано использование как блочной, так и однострочной форм инструкции **If...Then...Else**:

```
Dim Number, Digits, MyString
Number = 53 ' Инициализирует переменную.
If Number < 10 Then
    Digits = 1
ElseIf Number < 100 Then
    ' Условие является истинным, поэтому выполняется следующая инструкция.
    Digits = 2
Else
    Digits = 3
End If
```

'Использует однострочную форму для присвоения значения.
 If Digits = 1 Then MyString = "Один" Else MyString = "Больше 1"

Использование форм

Объект UserForm

Объект - комбинация кода и данных, которая может рассматриваться как единое целое, например, элемент управления, форма или компонент приложения.

Объект UserForm представляет собой окно или диалоговое окно, являющееся составной частью интерфейса пользователя в приложении.

Семейство UserForms является семейством, компоненты которого представляют все загруженные формы UserForm в приложении. У семейства UserForms имеются свойства Count и Item, а также метод Add. Свойство Count указывает число компонентов в семействе; свойство Item (компонент, используемый по умолчанию) указывает определенный компонент семейства; метод Add помещает в семейство новый компонент UserForm.

Синтаксис:

UserForm

UserForms[.Item](индекс)

Прототип индекс представляет собой значение типа Integer в диапазоне от 0 до UserForms.Count – 1. Item -- это используемый по умолчанию компонент семейства UserForms, поэтому его можно не указывать.

Формы пользователя имеют свойства, определяющие внешний вид, как например, положение, размер, цвет и реакция на события.

Формы пользователя также могут реагировать на события, инициированные пользователем или запущенные системой. Например, пользователь может написать подпрограмму обработки события Initialize (Инициализация) для UserForm, чтобы инициализировать переменные уровня модуля до того, как UserForm будет выведена на экран.

Помимо свойств и событий для программной обработки форм пользователя могут использоваться методы. Например, можно использовать метод Move для изменения положения и размера формы UserForm.

При конструировании форм пользователя следует задать свойство BorderStyle для определения границ и свойство Caption для занесения текста в область заголовка. В программе можно использовать методы Show и Hide, чтобы делать форму соответственно видимой и невидимой во время выполнения.

Доступ к семейству элементов управления формы UserForm осуществляется с помощью семейства Controls. Например, чтобы скрыть все элементы управления формы UserForm, можно использовать код, подобный следующему:

```
For Each Control in UserForm1.Controls
    Control.Visible = False
Next Control
```

Метод Show

Метод Show выводит на экран объект UserForm. Синтаксис: **[объект.]Show**

Необязательный прототип объект представляет собой объектное выражение, значением которого является один из объектов, перечисленных в списке "Применение". Если этот элемент синтаксиса опущен, предполагается, что объектом является форма UserForm, связанная с активным модулем UserForm.

Когда UserForm выведена на экран, другие формы данного приложения недоступны, однако это не относится к другим приложениям.