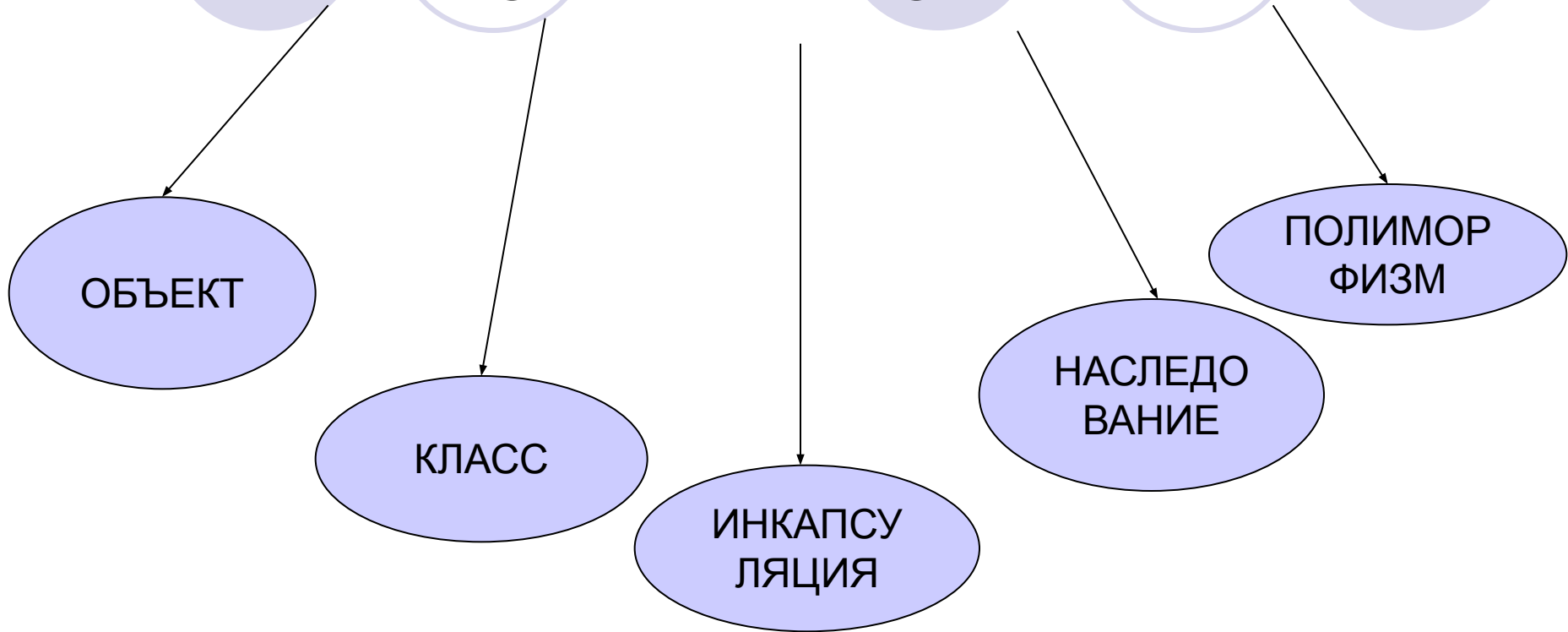




Основы программирования в среде Visual Basic for Application (VBA)

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ



Структура программного кода VBA

- Процедуры макросов

Sub ИмяМакроса()

<Команды макроса>

End Sub

- Процедуры обработки событий, связанных с каким-либо объектом

Private Sub ИмяОбъекта_Событие()

<Код обработки события>

End Sub

- Процедуры пользователя

[Private | Public] Sub ИмяПроцедуры(Список параметров)

<Команды подпрограммы>

End Sub

Язык программирования VBA

- Форматы операторов VBA:

Объект.Свойство = Значение

Объект.Метод [Параметр1] [,Параметр2]...] или

Объект.Метод [Параметр1:=”значение1”] [,Параметр2 :=”значение1”] ...] , где:

Объект – имя объекта, для которого устанавливается свойство или метод.

Свойство – характеристика объекта, подлежащая изменению.

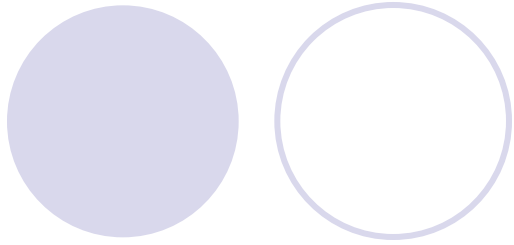
Значение – новое состояние объекта.

Метод – команда или процедура.

Параметр – аргумент, используемый методом. В качестве параметров методам передается дополнительная информация, ее необходимо указывать через пробел. Если параметров несколько, они разделяются запятой.

Пример

ActiveWorkbook.Close SaveChange:=True,FileName:=“Книга1.xlsx” 'оператор закрытия рабочей книги с сохранением. Метод Close использует именованные параметры.



- Объявление переменных:

Dim ИмяПеременной [As Тип], где:

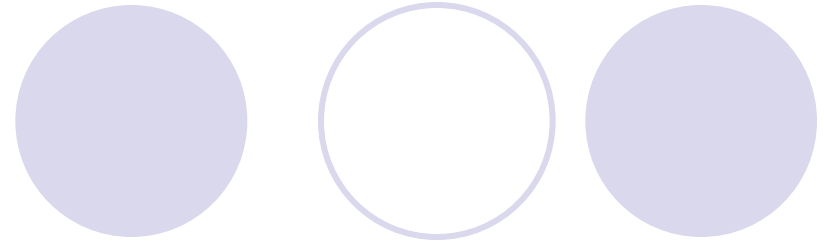
Dim – оператор, сообщающий программе имя переменной и резервирующий область памяти для хранения ее значения.

ИмяПеременной – идентификатор, определяющий допустимое имя.

As Тип – определяет тип переменной.

Пример

- Dim a As String
- Dim b As Integer



- Адресация ячеек:

Range – свойство, определяющее диапазон ячеек;

Cells – свойство, позволяющее выбирать ячейки рабочего листа;

ActiveCell – свойство, возвращающее одну активную ячейку;

Columns – свойство, возвращающее столбец;

Rows – свойство, возвращающее строку;

Selection – метод, возвращающий выделенный объект.

Для ссылки на одиночную ячейку используются команды:

- **[Объект.] Range (“адрес ячейки”)**

Для свойства Range в качестве аргумента используется любая ссылка в формате A1.

Например, `Worksheets(“Лист1”).Range(“A2”)=45` или для текущего листа `Range(“A2”)=45` – запись в ячейку A2 числа 45.

- **[Объект.] Cells(<номер строки>, <номер столбца>)**

Свойство используется для доступа к отдельной ячейке. В качестве аргументов указываются номер строки и столбца.

Например, для ввода числа 45 в ячейку Лист1!A2, необходима команда:

```
Worksheets(1).Cells(2, 1).Value=45
```

Операторы условия:

- Линейный оператор

IF <Условие> THEN <Оператор1> [ELSE <Оператор2>]

Пример

If Range("D2").Value<15000 Then Range("E2").Value=0.2

- Блочный оператор

IF <Условное выражение> THEN

<Группа операторов1>

[ELSE

<Группа операторов2>]

END IF

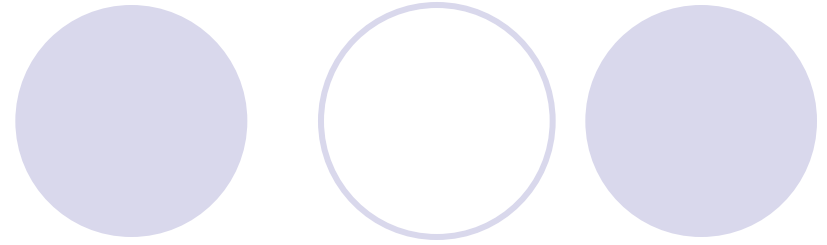
Пример



Если заработная плата сотрудника, расположенная в ячейке D2 текущего листа, составляет менее 15000 рублей, то необходимо назначить ему премию в 20% (ячейка E2). Если заработная плата сотрудника составляет от 15000 до 30000 рублей, то необходимо назначить премию в 10%, во всех остальных случаях – 5%.

```
If Range("D2").Value < 15000 Then
Range("E2").Value = 0.2
Else
If (Range("D2").Value >= 15000) And (Range("D2").Value < 30000) Then
Range("E2").Value = 0.1
Else
Range("E2").Value = 0.05
End If
End If
```


Цикл со счетчиком



FOR СчетчикЦиклов = НачальноеЗначение TO КонечноеЗначение [STEP Шаг]

<Группа операторов>

[EXIT FOR]

NEXT [СчетчикЦиклов]

где:

FOR , TO, NEXT – обозначение начала цикла, разделение начального и конечного значения счетчика и конец цикла;

СчетчикЦиклов – целочисленная переменная;

STEP – задание шага цикла;

Шаг – значение шага, по умолчанию 1;

EXIT FOR – необязательный оператор принудительного выхода из цикла.

Пример



Найти произведение четных чисел от 1 до 10 включительно и поместить результат в ячейку A2 текущего рабочего листа.

```
Sub proizved()
```

```
Dim p As Integer, i As Integer 'объявление целочисленных переменных.
```

```
p=1
```

```
For i = 2 To 10 Step 2 'переменной-счетчику i присваиваем начальное и конечное значение с шагом 2.
```

```
p = p*i 'переменная-счетчик увеличивается на величину шага и проверяется на достижение конечного значения.
```

```
Next 'передача управления в начало цикла.
```

```
Range("A2").Value = p 'вычисленное произведение помещается в ячейку A2 текущего листа.
```

```
End Sub
```

Цикл с условием

Формат цикла с
предусловием

Формат цикла с
постусловием

DO WHILE | UNTIL <Условие >
<Группа операторов>
[EXIT DO]
LOOP

DO
<Группа операторов>
[EXIT DO]
LOOP WHILE | UNTIL <Условие >

где DO , LOOP – обозначение начала и окончания цикла;
WHILE и UNTIL – обозначение типа цикла. Цикл DO WHILE
выполняется до тех пор, пока условие имеет значение True. Цикл
DO UNTIL выполняется до тех пор, пока условие имеет значение
False.

EXIT DO – принудительный выход из цикла.

Пример. Реализация предыдущего примера на базе цикла с условием будет иметь следующий вид.

Цикл с предусловием	Цикл с постусловием
<pre>i = 2 DO WHILE i<=10 p = p*i i=i+2 LOOP</pre>	<pre>i = 2 DO p = p*i i=i+2 LOOP UNTIL i>10</pre>

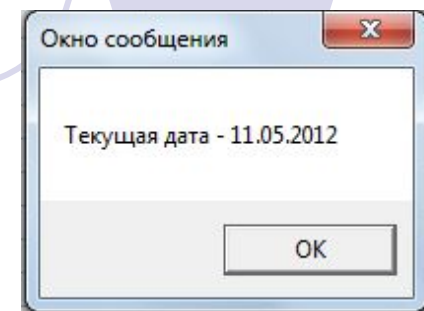
Основы создания интерфейса пользователя

- Окна сообщения

`MsgBox < выражение> [, <Заголовок окна>]`

Пример.

`MsgBox "Текущая дата - " & Date, , "Окно сообщения"`



- Окна ввода

`text = InputBox(<"Сообщение"> [,<"Заголовок">]) ,`

где, `text` – переменная, куда помещается введенная информация;

`Сообщение` – текст сообщения;

`Заголовок` – заголовок окна.

Пример.

`имя=InputBox("Введите Ваше Имя", "Окно ввода")`

