

Материалы для подготовки по теме объектно-ориентированное программирование по учебнику Н. Угриновича «Информатика и ИКТ 9 класс»

Материалы подготовлены учителем информатики: Белкиным В.В.

Контрольные вопросы

1. Какие объекты могут быть использованы при конструировании графического интерфейса проекта?
2. На какие события реагирует кнопка?

Задания для самостоятельного выполнения

Windows-CD 

- 2.4. *Практическое задание.* Запустить систему объектно-ориентированного визуального программирования Visual Basic 2005, создать проект «Первый проект», описанный в параграфе.

2.3. Переменные: тип, имя, значение

В объектно-ориентированном языке программирования Visual Basic переменные используются для хранения и обработки данных в программах.

Переменные задаются именами, которые определяют области оперативной памяти компьютера, в которых хранятся значения переменных. Значениями переменных могут быть данные различных типов (целые или вещественные числа, последовательности символов, логические значения и т. д.).



Переменная в программе представлена именем и служит для обращения к данным определенного типа, конкретные значения, которых хранятся в ячейках оперативной памяти.

Тип переменной. Тип переменных определяется диапазоном значений, которые могут принимать переменные, и допустимыми операциями над этими значениями. Значениями переменных числовых типов *Byte*, *Short*, *Integer*, *Long*, *Single*, *Double* являются числа, логического типа *Boolean* — значения *True* («истина») или *False* («ложь»), строкового типа *String* — последовательности символов.

Различные типы данных требуют для своего хранения в оперативной памяти компьютера различное количество ячеек (байтов) (табл. 2.2).

Таблица 2.2. Некоторые типы переменных в языке Visual Basic 2005

Тип переменной	Возможные значения	Объем занимаемой памяти
<i>Byte</i>	Целые неотрицательные числа от 0 до 255	1 байт
<i>Short</i>	Целые числа от -32 768 до 32 767	2 байта
<i>Integer</i>	Целые числа от -2 147 483 648 до 2 147 483 647	4 байта
<i>Long</i>	Целые числа от -9 223 372 036 854 до 9 223 372 036 853	8 байтов
<i>Single</i>	Десятичные числа одинарной точности (7–8 значащих цифр) от $-1,4 \cdot 10^{-45}$ до $3,4 \cdot 10^{38}$	4 байта
<i>Double</i>	Десятичные числа двойной точности (15–16 значащих цифр) от $-5,0 \cdot 10^{-324}$ до $1,7 \cdot 10^{308}$	8 байтов
<i>Boolean</i>	Логическое значение <i>True</i> или <i>False</i>	2 байта
<i>String</i>	Строка символов в кодировке Unicode	2 байта на символ
<i>Date</i>	Даты от 1 января 0001 года до 31 декабря 9999 года и время от 0:00:00 до 23:59:59	8 байтов

Имя переменной. Имена переменных определяют области оперативной памяти компьютера, в которых хранятся значения переменных. Имя каждой переменной (идентификатор) уникально и не может меняться в процессе выполнения программы. Имя переменной может состоять из различных символов (латинские и русские буквы, цифры и т. д.), но должно обязательно начинаться с буквы и не должно включать знак точки «.». Количество символов в имени не может быть более 1023, однако для удобства обычно ограничиваются несколькими символами.

Объявление переменных. Необходимо объявлять переменные, для того чтобы исполнитель программы (компьютер) «понимал», переменные какого типа используются в программе.

Для объявления переменной используется оператор *Dim*. С помощью одного оператора можно объявить сразу несколько переменных, например:

```
Dim A As Byte, B As Short, C As Single,
    D As String, G As Boolean
```

Присваивание переменным значений. Переменная может получить или изменить значение с помощью оператора присваивания. При выполнении оператора присваивания

переменная, имя которой указано слева от знака равенства, получает значение, которое находится справа от знака равенства. Например:

```
A = 255
B = -32768
C = 3.14
D = "информатика"
G = True
```

Значение переменной может быть задано числом, строкой или логическим значением, а также может быть представлено с помощью арифметического, строкового или логического выражения.

2.4. Арифметические, строковые и логические выражения

Проект «Переменные». Создать проект, в котором объявить переменные различных типов, присвоить им значения и вывести значения в поле списка, размещенное на форме.

Создадим графический интерфейс (рис. 2.8).

1. Поместить на форму:

- поле списка `ListBox1` для вывода значений переменных;
- кнопку `Button1` для запуска событийной процедуры.

Создадим событийную процедуру, реализующую присваивание значений переменным различных типов. Вывод значений переменных в поле списка произведем с использованием метода `Items.Add()`, аргументами которого будут переменные.

2. Dim A As Byte, B As Short, C As Single,

D As String, G As Boolean

```
Private Sub Button1_Click()
```

```
A = 255
```

```
B = -32768
```

```
C = 3.14
```

```
D = "информатика"
```

```
G = True
```

```
ListBox1.Items.Add(A)
```

```
ListBox1.Items.Add(B)
```

```
ListBox1.Items.Add(C)
```

```
ListBox1.Items.Add(D)
```

```
ListBox1.Items.Add(G)
```

```
End Sub
```

3. Запустить проект на выполнение. После щелчка по кнопке начнет выполняться событийная процедура, в которой будут выполнены операции присваивания (в отведенные переменным области оперативной памяти будут записаны их значения).

Затем с помощью метода `Items.Add()` будет произведен вывод значений переменных в поле списка. В этом процессе значения переменных считываются из оперативной памяти и печатаются в столбик в поле списка (см. рис. 2.8).

Проанализируем процесс выполнения программы компьютером. После запуска проекта оператор объявления переменных `Dim` ответит в оперативной памяти для их хранения необходимое количество ячеек (табл. 2.3):

- для целой неотрицательной переменной `A` — одну ячейку;
- для целочисленной переменной `B` — две ячейки;
- для переменной одинарной точности `C` — четыре ячейки;
- для строковой переменной `D` — по две ячейки на символ;
- для логической переменной `G` — две ячейки.

Таблица 2.3. Значения переменных в оперативной памяти

Имя переменной	Оперативная память	
	Номера ячеек	Значение переменной
A	1	255
B	2-3	-32768
C	4-7	3,14
D	8-29	информатика
G	30-31	True

Таким образом, в памяти для хранения значений переменных будет отведена 31 ячейка, например, ячейки с 1-й по 31-ю.

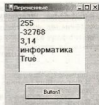


Рис. 2.8. Проект «Переменные»

Проект «Переменные» хранится
в папке ..\informatika9\Переменные!

Windows-CD 

Контрольные вопросы

1. В чем состоит разница между типом, именем и значением переменной?
2. Какие основные типы переменных используются в языке программирования Visual Basic 2005?
3. Почему рекомендуется объявлять переменные перед их использованием в программе?

Задания для самостоятельного выполнения

- 2.5. *Задание с кратким ответом.* Определить количество ячеек оперативной памяти, необходимое для хранения значений переменных, перечисленных в табл. 2.2 (по одной переменной первых семи типов).
- 2.6. *Практическое задание.* В системе программирования Visual Basic 2005 создать проект «Переменные», описанный в параграфе.

2.4. Арифметические, строковые и логические выражения

2.4.1. Арифметические выражения

В состав арифметических выражений могут входить переменные числового типа, числа, знаки арифметических операций, а также математические функции.

2.5.1. Математические функции

Порядок вычисления арифметических выражений производится в соответствии с общезвестным порядком выполнения арифметических операций (возведение в степень, умножение или деление, сложение или вычитание), который может изменяться с помощью скобок.

Проект «Калькулятор». Разработать проект, который позволит производить четыре арифметических действия над числами (сложение, вычитание, умножение и деление).

Работа над проектом начинается с создания графического интерфейса, для этого на форму помещаются элементы управления (рис. 2.10).

1. Разместить на форме:

- два текстовых поля TextBox1 и TextBox2 для ввода числовых данных;
- надпись Label1 для вывода результата;
- четыре кнопки Button1, Button2, Button3 и Button4 для создания событийных процедур: сложения, вычитания, умножения, деления.

Следующим шагом является создание программного кода событийных процедур. Двойной щелчок мышью по кнопке вызывает окно *Программный код* с пустой заготовкой событийной процедуры. Событийная процедура сложения чисел Button1_Click должна присвоить значению свойства Text надписи Label1 сумму числовых значений, введенных в текстовые поля TextBox1 и TextBox2.

Для преобразования строковых значений свойства Text текстовых полей в десятичные числа воспользуемся функцией Val(), аргументом которой является строка, а значением — число.

```
2. Private Sub Button1_Click()  
Label1.Text = Val(TextBox1.Text) +  
Val(TextBox2.Text)  
End Sub
```

Событийные процедуры вычитания, умножения и деления создаются аналогично.

Графический интерфейс проекта можно сделать более понятным и привлекательным.

3. Для этого необходимо в режиме конструирования проекта последовательно выделить объекты графического интерфейса и с помощью диалогового окна *Свойства (Properties)* установить новые значения некоторых свойств для каждого объекта:

- для объекта «форма» Form1 изменить значение свойства Text и цвет (значение свойства BackColor);
- для объектов «кнопка» Button1, Button2, Button3 и Button4 изменить значение свойства Text;
- для объектов «текстовое поле» TextBox1 и TextBox2 установить выравнивание текста по правому краю

(значение свойства `TextAlign`) и шрифт (значение свойства `Font`);

- для объекта «надпись» `Label1` удалить значение свойства `Text`, установить выравнивание текста по правому краю (значение свойства `TextAlign`), шрифт (значение свойства `Font`) и установить запрет на автоматическое изменение размера (значение свойства `AutoSize`) (табл. 2.4).

Таблица 2.4. Объекты графического интерфейса и значения их некоторых свойств

Объект	Свойство	Значение по умолчанию	Новое значение
Form1	Text	Form1	Обычный калькулятор
	BackColor	Серый	Синий
Button1	Text	Button1	+
Button2	Text	Button2	=
Button3	Text	Button3	*
Button4	Text	Button4	/
TextBox1	TextAlign	Left	Right
	Font	MS Sans Serif, обычный, 8	MS Sans Serif, обычный, 18
TextBox2	TextAlign	Left	Right
	Font	MS Sans Serif, обычный, 8	MS Sans Serif, обычный, 18
Label1	Text	Label1	""
	Font	MS Sans Serif, обычный, 8	MS Sans Serif, обычный, 18
	AutoSize	True	False
	BackColor	Серый	Белый

Изменения значений свойств объектов с помощью диалогового окна *Свойства (Properties)* могут производиться различными способами. В большинстве случаев нужно просто стереть старое значение свойства и ввести новое. Однако для ввода значений некоторых свойств используются раскрывающиеся списки или диалоговые окна. Так, автоматическое изменение размера надписи (значение свойства `AutoSize`) устанавливается с использованием списка, а цвет фона (значение свойства `BackColor`) и шрифт (значение свойства `Font`) устанавливаются с использованием диалоговых окон (рис. 2.9).



Рис. 2.9. Установка значений свойств объектов с использованием раскрывающихся списков и диалоговых окон

4. Запустить проект на выполнение. Ввести числа в два текстовых поля и щелкнуть по кнопке арифметической операции. На надпись будет выведен результат (см. рис. 2.10).



Рис. 2.10. Проект «Калькулятор»

Проект «Калькулятор» хранится в папке `..informatika9\Калькулятор1` Windows-CD

Контрольные вопросы

1. Какие существуют способы установки значений свойств элементов управления с использованием окна *Свойства*?

Задания для самостоятельного выполнения

Windows-CD 

2.7. *Практическое задание.* В системе программирования Visual Basic 2005 создать проект «Калькулятор». Дополнить проект операцией возведения числа в степень (знак операции возведения в степень «^»).

2.4.2. Строковые выражения

В состав строковых выражений могут входить переменные строкового типа, строки (последовательности символов) и строковые функции.

2.5.2. Строковые функции

Над переменными и строками может производиться операция конкатенации. Эта операция объединяет строки или значения строковых переменных в единую строку. Операция конкатенации обозначается знаком «+», который не следует путать со знаком сложения чисел в арифметических выражениях.

Проект «Сложение строк». Разработать проект, в котором для сложения двух строковых переменных A и B и строки "форма" используется операция конкатенации.

Создадим графический интерфейс проекта (рис. 2.11).

1. Поместить на форму кнопку Button1 и надпись Label1.

Щелчком по кнопке и введем в заготовку событийной процедуры программный код.

2. Dim A, B As String

```
Private Sub Button1_Click(...)
```

```
A = "ин"
```

```
B = "тика"
```

```
Label1.Text = A + "форма" + B
```

```
End Sub
```

3. После запуска проекта на выполнение и щелчка по кнопке *Сложение строк* на надписи появится слово «информатика» (см. рис. 2.11).

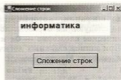


Рис. 2.11. Проект «Сложение строк»

Проект «Сложение строк»
хранится в папке
..informatika9\Сложение строк\

Windows-CD 

Контрольные вопросы

1. Какие элементы могут входить в состав строковых выражений?

Задания для самостоятельного выполнения

Windows-CD 

2.8. *Практическое задание.* В системе программирования Visual Basic 2005 создать проект «Сложение строк». Изменить программный код событийной процедуры так, чтобы выполнялась конкатенация строк «Кило» и «байт».

2.4.3. *Логические выражения

В состав логических выражений могут входить логические переменные, логические значения, результаты операций сравнения чисел и строк, а также логические операции. Логические выражения могут принимать лишь два значения: **True** (Истина) и **False** (Ложь).

Операции сравнения =, >, <, <>, >= и <= сравнивают выражение в левой части оператора с выражением в правой части оператора и представляют результат в виде логического значения **True** или **False**. Например:

```
5 > 3 = True; "А" = "В" = False
```

Над элементами логических выражений могут производиться логические операции, которые на языке Visual Basic обозначаются следующим образом: логическое умножение — **And**, логическое сложение — **Or** и логическое отрицание — **Not**. При записи сложных логических выражений используются скобки. Например:

```
(5 > 3) And ("A" = "B") = False
```

```
(5 > 3) Or ("A" = "B") = True
```

```
Not (5 > 3) = False
```

Проект «Логические выражения». Разработать проект, в котором определяется истинность или ложность логических выражений, рассмотренных выше.

Создадим графический интерфейс проекта (рис. 2.12).

1. Поместить на форму:

- кнопку Button1 для запуска событийной процедуры;
- надписи Label1, Label2, Label3, Label4 и Label5 для вывода значений пяти логических выражений;
- пять надписей для вывода самих логических выражений.

Создадим событийную процедуру, в которой значения логических выражений выводятся на пять надписей.

2. Private Sub Button1_Click(...)

```
Label1.Text = 5 > 3
```

```
Label2.Text = "A" = "B"
```

```
Label3.Text = (5 > 3) And ("A" = "B")
```

```
Label4.Text = (5 > 3) Or ("A" = "B")
```

```
Label5.Text = Not (5 > 3)
```

```
End Sub
```

3. После запуска проекта и щелчка по кнопке *Логические выражения* на надписи будут выведены значения пяти логических выражений (см. рис. 2.12).

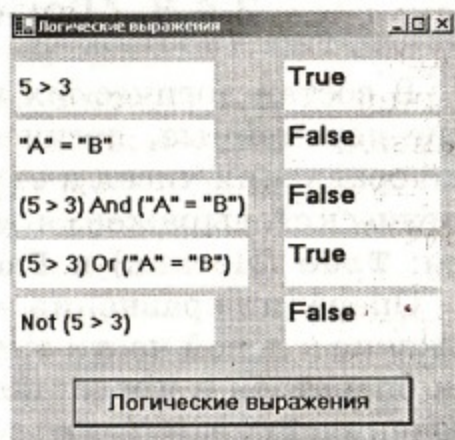


Рис. 2.12. Проект «Логические выражения»