

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

Материалы подготовлены учителем информатики: Белкиным В.В.

2.2. Основы объектно-ориентированного визуального программирования на языке Visual Basic

2.2.1. Разработка проектов в системе программирования Visual Basic 2005

Проект (Project). С одной стороны, система объектно-ориентированного визуального программирования Visual Basic является **системой программирования**, так как позволяет кодировать алгоритмы на этом языке. С другой стороны, система объектно-ориентированного визуального программирования Visual Basic является **средой проектирования**, так как позволяет осуществлять визуальное конструирование графического интерфейса.

Результатом процессов программирования и проектирования является **проект**, который объединяет в себе программный код и графический интерфейс. Система объектно-ориентированного программирования Visual Basic содержит программу-транслятор, поэтому проекты могут выполняться в самой системе, а также могут быть преобразованы в **приложения**, которые выполняются в операционной системе Windows.

Программы-трансляторы языков программирования бывают двух типов: интерпретаторы и компиляторы. **Интерпретатор** — это программа, которая обеспечивает последовательный «перевод» команд программы на машинный язык с одновременным их выполнением. Поэтому при каждом запуске программы на выполнение эта процедура повторяется. Достоинством интерпретаторов является удобство отладки программы (поиска в ней ошибок), так как возможно «пошаговое» ее выполнение, а недостатком — сравнительно малая скорость выполнения.

Компилятор действует иначе, он переводит весь текст программы на машинный язык и сохраняет его в исполнимом файле (обычно с расширением exe). Затем этот уже готовый к выполнению файл, записанный на машинном языке, можно запускать на выполнение. Достоинством компиляторов является большая скорость выполнения программы, а недостатком — трудоемкость отладки, так как невозможно пошаговое выполнение программы.

Система объектно-ориентированного программирования Visual Basic позволяет работать в режиме как интерпретатора, так и компилятора. На этапе разработки и отладки программы используется режим интерпретатора, а для получения готовой исполнимой программы — режим компилятора.

Система программирования Visual Basic 2005. Система программирования Visual Basic 2005 предоставляет пользователю удобный графический интерфейс в процессе разработки проектов.

После запуска системы программирования ввести команду [**Файл-Создать проект...**] (**File-New project...**). В появившемся диалоговом окне **Новый проект (New project)** выбрать тип создаваемого проекта **Приложение Windows (Windows Application)** и в текстовое поле **Имя (Name)** ввести название проекта, например **Первый проект**. Щелкнуть по кнопке **OK** (рис. 2.2).

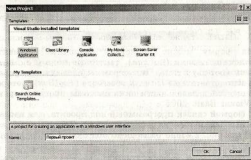


Рис. 2.2. Создание проекта в системе программирования Visual Basic 2005

После этого появится окно системы программирования, включающее несколько окон (рис. 2.3).

Визуальное конструирование графического интерфейса проекта производится в окне **Конструктор форм (Designer)**. Оно располагается в центре окна разработки проектов и содержит **форму** (в данном случае **Form1**), являющуюся основой графического интерфейса проекта.



Рис. 2.3. Система программирования Visual Basic 2005

На форму можно поместить различные элементы управления: кнопки (Button), текстовые поля (TextBox), надписи (Label) и т. д. Пиктограммы элементов управления располагаются на *Панели объектов (ToolBox)*, которая размещается в левой части окна системы программирования Visual Basic 2005.

С формой связан программный код проекта, для ввода и редактирования которого служит окно *Программный код (Code)* (на рис. 2.3 оно размещено под окном *Конструктор форм*).

Для перехода в окно *Программный код* применяется команда [Вид-Код] ([View-Code]), а для обратного перехода в окно конструирования графического интерфейса *Конструктор форм* применяется команда [Вид-Конструктор] ([View-Designer]).

Справа располагается окно *Свойства (Properties)*. Окно содержит список свойств, относящихся к выбранному объекту (форме или элементу управления на форме). В левом столбце находятся названия свойств, а в правом — их значения. Установленные по умолчанию значения могут быть изменены.

Этапы разработки проекта. Создание проектов в системе объектно-ориентированного визуального программирования Visual Basic реализуется в режиме [design]. Создание проекта можно разделить на несколько этапов.

1. Создание графического интерфейса проекта. В окне *Конструктор форм* с использованием *Панели объектов* на форму помещаются элементы управления, которые должны обеспечить взаимодействие проекта с пользователем.
2. Установка значений свойств объектов графического интерфейса. С помощью окна *Свойства* задаются значения свойств элементов управления, помещенных ранее на форму.
3. Создание и редактирование программного кода. В окне *Редактор кода* производится ввод и редактирование программного кода проекта.
4. Сохранение проекта. Так как проекты включают в себя несколько файлов, необходимо каждый проект сохранять в отдельной папке. Сохранение проекта производится командой [Файл-Сохранить все] ([File-Save all]).

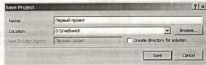



Рис. 2.4. Сохранение проекта

В появившемся диалоговом окне *Сохранить проект (Save Project)* в текстовом поле *Имя: (Name:)* можно уточнить имя проекта. Ввести путь к папке проекта можно в текстовом поле *Расположение: (Location:)* или выбрать ее расположение в файловой системе после щелчка по кнопке *Обзор... (Browse...)*.

Выполнение проекта. Загрузка проекта в систему программирования Visual Basic 2005 производится путем активации в папке проекта основного файла проекта (файла с расширением vbproj).

Запуск проекта на выполнение производится командой [Отладка-Начать] ([Debug-Start]) или щелчком по кнопке на панели инструментов окна системы программирования. После этого система программирования переходит в

режим выполнения проекта *run*, в котором редактирование графического интерфейса или программного кода невозможно.

Для окончания выполнения проекта и перехода в режим конструирования проекта [*design*] необходимо ввести команду [*Отладка-Остановить отладку*] (*[Debug-Stop Debugging]*) или щелкнуть по кнопке  на панели инструментов окна системы программирования.

Контрольные вопросы


1. Перечислите основные окна системы программирования Visual Basic 2005 и объясните их назначение.
2. Каковы основные этапы разработки проекта на языке Visual Basic 2005 и порядок его сохранения?
3. Какие существуют режимы работы с проектом в системе программирования Visual Basic 2005?

Задания для самостоятельного выполнения

- 2.3. *Практическое задание.* Запустить систему объектно-ориентированного визуального программирования Visual Basic 2005, создать новый проект и сохранить его.

2.2.2. Объекты: свойства и методы

Объекты (Objects). Как конструирование графического интерфейса, так и разработка программного кода базируется на использовании программных объектов. Каждый объект обладает определенным набором свойств и может использовать определенные методы обработки данных. Если говорить образно, то объекты — это существительные, свойства объекта — это прилагательные, а методы объекта — это глаголы.

 Программные объекты обладают свойствами и могут использовать методы обработки данных.

Классы объектов являются «шаблонами», определяющими наборы свойств, методов и событий, по которым создаются объекты. Основными классами объектов являются объекты, реализующие графический интерфейс объектов.

Объект, созданный по «шаблону» класса объектов, является экземпляром класса и наследует весь набор свойств, методов и событий данного класса. Каждый экземпляр класса объектов имеет уникальное для данного класса имя.

Основой для создания графического интерфейса проекта является объект «форма» (рис. 2.5).

На основании класса объектов *Form* можно создавать экземпляры объектов «форма», которые получают имена *Form1*, *Form2* и т. д.



Рис. 2.5. Объект «форма»

Свойства объекта (Properties). Каждый класс объектов обладает определенным набором свойств. Так, например, класс объектов *Form* обладает несколькими десятками различных свойств, которые определяют размеры объекта «форма», цвет формы, положение на экране монитора и т. д. (табл. 2.1).

Таблица 2.1. Некоторые свойства объекта «форма»

Свойство	Значение по умолчанию	Комментарий
Name	Form1	Имя объекта, используется в программном коде для обращения к объекту
Text	Form1	Текст в левом верхнем углу формы
BackColor	Control	Серый цвет фона формы
Font	MS Sans Serif, обычный, 8	Шрифт, его начертание и размер

Различные экземпляры класса объектов обладают одинаковым набором свойств, однако значения свойств у них могут отличаться. Первоначальные значения свойств объектов можно установить с использованием диалогового окна *Свойства (Properties)* системы программирования (см. рис. 2.3).

Так, для объекта «форма» *Form1* можно установить требуемое значение любого свойства. Для этого необходимо выбрать свойство из списка и изменить его значение.

Значения свойств объектов можно изменять в программном коде. Для присваивания свойству объекта нового значения в левой части строки программного кода необходимо указать имя объекта и затем — название свойства, которые в соответствии с **правилами точечной нотации** разделяются между собой точкой. В правой части строки необходимо записать конкретное значение свойства:



```
Объект.Свойство = ЗначениеСвойства
```

Например, новая надпись «Первый проект» в левом верхнем углу объекта Form1 (значение свойства Text) появится в результате выполнения программного кода:

```
Form1.Text = "Первый проект"
```

Методы объекта (Methods). Объекты могут использовать различные методы обработки данных. Методы имеют аргументы, которые позволяют задать значения параметров выполняемых действий.

Для использования метода в строке программного кода необходимо указать имя объекта и затем метод, которые в соответствии с правилами точечной нотации разделяются между собой точкой. В скобках при необходимости записываются аргументы метода, разделяемые запятыми:



```
Объект.Метод (arg1, arg2)
```

Например, с помощью метода Scale(x, y) можно изменить размеры формы или элемента управления. Аргументы метода x и y являются коэффициентами масштабирования по горизонтали и вертикали, т. е. позволяют увеличить или уменьшить ширину и высоту элемента управления. Например, можно в два раза увеличить размер объекта по оси X и в два раза его уменьшить по оси Y:

```
Me.Scale(2, 0.5)
```



Если производятся операции над самой формой, то вместо ее имени (например, Form1) в программном коде используется имя **Me**.

Контрольные вопросы

1. Что можно изменить в выбранном объекте: набор свойств, набор методов, значения свойств?
2. Каким образом можно изменить значения свойств программного объекта?

2.2.3. Графический интерфейс проекта и событийные процедуры

Графический интерфейс проекта. Графический интерфейс необходим для реализации интерактивного диалога пользователя с запущенным на выполнение готовым проектом. Основой для создания графического интерфейса разрабатываемого проекта является объект «форма». Форма представляет собой окно, на котором размещаются другие объекты — **элементы управления**.

Элементы управления имеют различное назначение. Текстовые поля (TextBox) используются для ввода и вывода данных, надписи (Label) — для вывода данных и пояснительных текстов, графические окна (PictureBox) — для вывода графики, кнопки (Button) — для запуска событийных процедур (рис. 2.6).

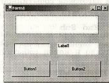


Рис. 2.6. Элементы управления на форме



Графический интерфейс проекта представляет собой форму, на которой размещены управляющие элементы.

Визуальное конструирование графического интерфейса проекта состоит в том, что на форме с помощью мыши «рисуются» те или иные элементы управления. Выбрав щелчком мыши нужный элемент на **Панели объектов**, мы можем поместить его на форму разрабатываемого проекта. Процесс размещения на форму элементов управления аналогичен рисованию графических примитивов с использованием графического редактора.

Событийные процедуры. Событие (Event) представляет собой действие, распознаваемое элементом управления. Событие может создаваться пользователем (например, щелчок мышью или нажатие клавиши) или быть результатом воздействия других программных объектов.

Каждый объект реагирует на определенный набор событий. Например, кнопка реагирует на щелчок (Click), нажатие (MouseDown) и отпускание (MouseUp) мыши или нажатие определенной клавиши на клавиатуре (KeyPress).

Для каждого события можно запрограммировать отклик, т. е. реакцию объекта на произошедшее событие. Если пользователь производит какое-либо воздействие на элемент графического интерфейса (например, щелчок), в качестве отклика выполняется **событийная процедура**, представляющая собой программу.

Для того чтобы создать заготовку событийной процедуры, необходимо в режиме разработки проекта осуществить двойной щелчок мышью по объекту. Например, после щелчка по кнопке Button1 в окне *Программный код* будет создана заготовка событийной процедуры:

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

```
End Sub
```

Служебные слова Private Sub и End Sub обозначают начало и конец событийной процедуры. Имя событийной процедуры Button1_Click(...) включает в себя имя объекта и имя события.

Далее необходимо ввести в событийную процедуру программный код, который реализует определенный алгоритм.



Событийная процедура представляет собой программу, которая начинает выполняться после реализации определенного события.

Проект «Первый проект». Создать проект, в котором в первой событийной процедуре свойству Text всех объектов будут присвоены новые значения, а во второй событийной процедуре будут изменены размеры формы с помощью метода Scale(), очищено текстовое поле с помощью метода Clear() и скрыто графическое поле с помощью метода Hide().

Создадим графический интерфейс проекта (рис. 2.7).

1. Поместить на форму две кнопки Button1 и Button2, текстовое поле TextBox1, надпись Label1 и графическое поле PictureBox1.
 2. Последовательно выделить объекты PictureBox1 и Label1 и с помощью диалогового окна *Свойства* установить для свойства BackColor (цвет фона) значение White (белый).
- Создадим событийные процедуры. Последовательно осуществим двойной щелчок по обеим кнопкам и внесем в заготовки событийных процедур программные коды.

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

```
Me.Text = "Первый проект"
```

```
TextBox1.Text = "Текстовое поле"
```

```
Label1.Text = "Надпись"
```

```
Button1.Text = "Свойства"
```

```
Button2.Text = "Методы"
```

```
End Sub
```

```
Private Sub Button2_Click(...)
```

```
Me.Scale(2, 0.5)
```

```
TextBox1.Clear()
```

```
PictureBox1.Hide()
```

```
End Sub
```

4. После запуска проекта на выполнение и щелчков по кнопкам событийные процедуры будут выполнены (см. рис. 2.7).

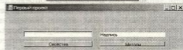


Рис. 2.7. Результаты выполнения событийных процедур

Проект хранится в папке
..informatika9\Первый проект\

Windows-CD