

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

Материалы подготовлены учителем информатики: Белкиным В.В.

Задания для самостоятельного выполнения

Windows-CD 

- 2.23. *Задание с развернутым ответом.* Начертить блок-схемы алгоритмических структур «цикл со счетчиком» и «цикл с условием».
- 2.24. *Практическое задание.* В системе программирования Visual Basic 2005 создать проект «Коды символов». Использовать для вывода числовых кодов и символов элементы управления ComboBox.
- 2.25. **Практическое задание.* В системе программирования Visual Basic 2005 создать проект «Слово-перевертыш». Предусмотреть проверку, является ли слово-перевертыш палиндромом.

2.7. Графические возможности языка программирования Visual Basic 2005

2.7.1. Графические методы языка Visual Basic 2005

На форме и управляющих элементах можно рисовать линии, прямоугольники, окружности и другие графические фигуры. Для рисования необходимо определить объекты Graphics (Область рисования), Pen (Перо) и Brush (Кисть).

Область рисования. Объект Graphics (Область рисования) позволяет выбрать в качестве области рисования определенный элемент управления и обладает методами рисования графических фигур. Сначала необходимо в разделе объявления переменных определить имя объекта, например:

```
Dim Graph1 As Graphics
```

Затем в программном коде событийной процедуры необходимо указать определенный элемент управления в качестве области рисования. Обычно в качестве области рисования выбирается размещенное на форме графическое поле (например, PictureBox1):

```
Graph1 = Me.PictureBox1.CreateGraphics()
```

Перо. Объект Pen (Перо) определяет цвет и ширину линии рисования. Сначала необходимо в разделе объявления переменных определить имя объекта (например, Pen1), установить цвет (например, красный Color.Red) и ширину линии в пикселях (например, 3):

```
Dim Pen1 As New Pen(Color.Red, 3)
```

Затем в программном коде событийной процедуры можно установить новые значения цвета и ширины линии, например:

```
Pen1.Color = Color.Green  
Pen1.Width = 15
```

Кисть. Объект Brush (Кисть) определяет цвет и стиль закрашивания прямоугольников, окружностей и других замкнутых фигур. Сначала необходимо в разделе объявления переменных определить имя объекта (например, Brush1) и установить тип закрашки и цвет (например, сплошная закрашка синего цвета SolidBrush(Color.Blue)):

```
Dim Brush1 As New SolidBrush(Color.Blue)
```

Затем в программном коде событийной процедуры можно установить новый цвет закрашки (например, пурпурный):

```
Brush1.Color = Color.Magenta
```

Графические методы. Графические фигуры рисуются с использованием графических методов. Замкнутые фигуры, такие как прямоугольники или эллипсы, состоят из двух частей — из контура и внутренней области. Контур рисуется с использованием заданного пера, а внутренняя область закрашивается с использованием заданной кисти.

DrawLine() — метод рисования линии, аргументами которого являются перо определенного цвета и толщины (например, Pen1), а также координаты концов линии X1, Y1 и X2, Y2:

```
Graph1.DrawLine(Pen1, X1, Y1, X2, Y2)
```

DrawRectangle() — метод рисования прямоугольника, аргументами которого являются перо определенного цвета и толщины (например, Pen1), а также координаты левого верхнего угла X1, Y1, ширина Width и высота Height:

```
Graph1.DrawRectangle(Pen1, X1, Y1, Width, Height)
```

FillRectangle() — метод закрашки прямоугольника с использованием кисти определенного цвета. Например:

```
Graph1.FillRectangle(Brush1, X1, Y1, Width, Height)
```

DrawEllipse() — метод рисования окружности или эллипса, аргументами которого являются перо определенного цвета и толщины (например, Pen1), а также координаты левого верхнего угла описанного прямоугольника X1, Y1, его ширина Width и высота Height:

```
Graph1.DrawEllipse(Pen1, X1, Y1, Width, Height)
```

`FillEllipse()` — метод закраски окружности или эллипса с использованием кисти определенного цвета. Например:

```
Graph1.FillEllipse(Brush1, X1, Y1, Width, Height)
```

Для рисования точки с заданными координатами $X1$ и $Y1$ можно использовать методы `DrawRectangle(Pen1, X1, Y1, 1, 1)` или `DrawEllipse(Pen1, X1, Y1, 1, 1)`, в которых аргументы `Width` и `Height` равны 1.

`Graph1.Clear()` — метод, заданным цветом стирающий изображения в области рисования. Например, белым цветом:

```
Graph1.Clear(Color.White)
```

Цвет. Цвет устанавливается как значение свойства `Color`. Можно установить цвет с использованием нескольких десятков цветовых констант. Ниже приведены примеры установки зеленого цвета для объекта `Pen1` (перо) и желтого цвета для объекта `Brush1` (кисть):

```
Pen1.Color = Color.Green
Brush1.Color = Color.Yellow
```

Для установки цвета в 24-битовой палитре цветов RGB используется метод `Color.FromArgb(Red, Green, Blue)`, аргументами которого являются три числа в диапазонах от 0 до 255 (интенсивности красного, зеленого и синего цветов). Например, так можно установить пурпурный цвет для объекта `Brush1` (Кисть):

```
Brush1.Color = Color.FromArgb(255, 0, 255)
```

Рисование текста. Метод `DrawString()` позволяет выводить текст в область рисования. Аргументами метода являются строка текста, шрифт, кисть и координаты начала строки. Объекты «шрифт» (например, `drawFont`) и «кисть» (например, `drawBrush`) необходимо объявить:

```
Dim drawFont As New Font("Arial", 16)
Dim drawBrush As New SolidBrush(Color.Black)
```

Рисование текста в поле рисования можно осуществить так:

```
Graph1.DrawString("Текст", drawFont, drawBrush,
10, 10)
```

Система координат. Рисование линий, прямоугольников и других фигур производится в компьютерной системе координат, начало которой расположено в верхнем левом углу формы или элемента управления. Ось X направлена вправо,

а ось Y направлена вниз. Единицей измерения по умолчанию является точка (пиксель). Компьютерная система координат графического поля шириной 300 точек и высотой 200 точек приведена на рис. 2.30.

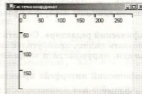


Рис. 2.30. Компьютерная система координат графического поля

При геометрических построениях и построении графиков функций удобнее использовать математическую систему координат, начало которой обычно находится в центре области рисования. Ось X направлена вправо, а ось Y направлена вверх. Математическая система координат графического поля шириной 300 точек и высотой 200 точек приведена на рис. 2.31.

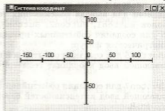


Рис. 2.31. Математическая система координат графического поля

Для преобразования компьютерной системы координат в математическую систему координат используется метод масштабирования и поворота осей `ScaleTransform()` и метод сдвига начала координат `TranslateTransform()`.

Метод `Graph1.ScaleTransform(1, -1)` обеспечивает поворот оси Y .

Метод `Graph1.TranslateTransform(150, -100)` обеспечивает сдвиг по оси X на 150 точек вправо и сдвиг по оси Y на 100 точек вниз.

В системе координат можно также изменить единицу измерения путем установки значения свойства PageUnit.

Graph1.PageUnit = GraphicsUnit.Document — устанавливает размер единицы измерения равным 1/300 дюйма.

Graph1.PageUnit = GraphicsUnit.Millimeter — устанавливает размер единицы измерения равным 1 миллиметру.

Проект «Графический редактор». Создать проект, который позволит рисовать линии, прямоугольники, закрашенные прямоугольники, окружности и закрашенные окружности.

Создадим графический интерфейс проекта (рис. 2.32).

1. Поместить на форму:

- графическое поле PictureBox1, которое будет использоваться в качестве области рисования;
- четыре текстовых поля TextBox1, TextBox2, TextBox3 и TextBox4 для ввода значений переменных X1, Y1 и X2, Y2, содержащих координаты графических примитивов;
- три текстовых поля TextBox5, TextBox6 и TextBox7 для ввода значений переменных Red, Green, Blue, содержащих числовые коды цветов;
- семь надписей для вывода поясняющих текстов;
- пять кнопок Button1, Button2, Button3, Button4, Button5 для создания событийных процедур рисования геометрических фигур;
- кнопку Button6 для создания событийной процедуры, обеспечивающей ввод координат геометрических фигур;
- кнопку Button7 для создания событийной процедуры, обеспечивающей ввод интенсивностей базовых цветов в системе RGB.

Последовательно выберем на форме все текстовые поля и с помощью диалогового окна *Свойства* присвоим свойству Text значение 0.

- Для большей понятности программного кода будем вводить в него комментарии, которые начинаются с символа апостроф «'».

Объявим объекты Graphics (Область рисования), Pen (Перо) и Brush (Кисть), а также переменные, которые будут использоваться в проекте.

```
2. Dim Graph1 As Graphics
   Dim Pen1 As New Pen(Color.Red, 3)
   Dim Brush1 As New SolidBrush(Color.Black)
   Dim X1, X2, Y1, Y2, Red, Green, Blue As Integer
```

Создадим программный код событийной процедуры рисования линии.

```
3. 'Линия
   Private Sub Button1_Click(...)
   Pen1.Color = Color.Red
   Graph1.DrawLine(Pen1, X1, Y1, X2, Y2)
   End Sub
```

Создадим программный код событийной процедуры рисования прямоугольника.

```
4. 'Прямоугольник, где переменная X2 содержит значение ширины, а переменная Y2 - значение высоты прямоугольника
   Private Sub Button2_Click(...)
   Pen1.Color = Color.Green
   Graph1.DrawRectangle(Pen1, X1, Y1, X2, Y2)
   End Sub
```

Создадим программный код событийной процедуры рисования закрашенного прямоугольника.

```
5. 'Закрашенный прямоугольник, где переменная X2 содержит значение ширины, а переменная Y2 - значение высоты прямоугольника
   Private Sub Button3_Click(...)
   Pen1.Color = Color.Blue
   Pen1.Width = 10
   Graph1.DrawRectangle(Pen1, X1, Y1, X2, Y2)
   Brush1.Color = Color.FromArgb(Red, Green, Blue)
   Graph1.FillRectangle(Brush1, X1, Y1, X2, Y2)
   End Sub
```

Создадим программный код событийной процедуры рисования окружности.

```
6. 'Окружность, где переменная X2 содержит значение ширины, а переменная Y2 значение высоты описанного прямоугольника
   Private Sub Button4_Click(...)
   Pen1.Color = Color.Magenta
   Graph1.DrawEllipse(Pen1, X1, Y1, X2, Y2)
   End Sub
```

Создадим программный код событийной процедуры рисования закрашенной окружности, цвет закрашки которой задается пользователем из палитры RGB путем ввода интенсивностей базовых цветов в текстовые поля.

7. *Закрашенная окружность, где переменная X2 содержит значение ширины, а переменная Y2 - значение высоты описанного прямоугольника*

```
Private Sub Button5_Click(...)
Red = TextBox5.Text
Green = TextBox6.Text
Blue = TextBox7.Text
Pen1.Color = Color.Blue
Graph1.DrawEllipse(Pen1, X1, Y1, X2, Y2)
Brush1.Color = Color.FromArgb(Red, Green, Blue)
Graph1.FillEllipse(Brush1, X1, Y1, X2, Y2)
End Sub
```

Создадим программный код событийной процедуры ввода координат графических фигур.

8. *Ввод координат*

```
Private Sub Button6_Click(...)
Graph1 = Me.PictureBox1.CreateGraphics()
X1 = TextBox1.Text
Y1 = TextBox2.Text
X2 = TextBox3.Text
Y2 = TextBox4.Text
End Sub
```

Создадим программный код событийной процедуры ввода интенсивностей базовых цветов в системе RGB.

9. *Цвет кисти*

```
Private Sub Button7_Click(...)
Red = Val(TextBox5.Text)
Green = Val(TextBox6.Text)
Blue = Val(TextBox7.Text)
End Sub
```

10. Запустить проект. Ввести в текстовые поля значения координат графических фигур и щелкнуть по кнопке *Координаты*.

Ввести в текстовые поля интенсивности базовых цветов и щелкнуть по кнопке *Цвет кисти*.

Нарисовать графические фигуры, последовательно щелкнув по кнопкам *Линия*, *Прямоугольник*, *Закрашенный прямоугольник*, *Окружность* и *Закрашенная Окружность* (см. рис. 2.32).



Рис. 2.32. Проект «Графический редактор»

Проект «Графический редактор» хранится в папке `..\informatika9\графический редактор\` Windows-CD

Проект «Система координат». Создать проект, который обеспечит рисование осей и печать шкалы в компьютерной системе координат (см. рис. 2.30) и математической системе координат (см. рис. 2.31).

Создадим графический интерфейс проекта.

1. Разместить на форме:

- 1 графическое поле PictureBox1, которое будет использоваться в качестве области рисования;
- 2 кнопки Button1 и Button2 для создания событийных процедур.

Установим размеры графического поля PictureBox1.

2. Присвоить свойству Size значение 300;200.

Создадим событийную процедуру рисования осей и печатки шкал в компьютерной системе координат.

3. *Объявление переменных*

```
Dim Graph1 As Graphics
Dim Pen1 As New Pen(Color.Red, 3)
Dim drawBrush As New SolidBrush(Color.Black)
Dim drawFont As New Font("Arial", 10)
Dim X, Y As Integer
Private Sub Button1_Click(...)
Graph1 = Me.PictureBox1.CreateGraphics()
```

```

Graph1.Clear(Color.White)
'Рисование шкал компьютерной системы координат
Graph1.DrawLine(Pen1, 0, 0, 300, 0) 'Ось X
Graph1.DrawLine(Pen1, 0, 0, 0, 200) 'Ось Y
For X = 0 To 300 Step 50 'Засечки на оси X
Graph1.DrawLine(Pen1, X, 0, X, 10)
Next X
For Y = 0 To 200 Step 50 'Засечки на оси Y
Graph1.DrawLine(Pen1, 0, Y, 10, Y)
Next Y
'Печать шкалы оси X
For X = 0 To 300 Step 50
Graph1.DrawString(X, drawFont, drawBrush, X, 10)
Next X
'Печать шкалы оси Y
For Y = 0 To 200 Step 50
Graph1.DrawString(Y, drawFont, drawBrush, 10, Y)
Next Y
End Sub

```

Создадим событийную процедуру рисования осей и печати шкал в математической системе координат.

```


4. Private Sub Button1_Click(...)
Graph1 = Me.PictureBox1.CreateGraphics()
Graph1.Clear(Color.White)
'Печать шкал математической системы координат
в компьютерной системе координат
For X = -150 To 150 Step 50
Graph1.DrawString(X, drawFont, drawBrush,
X + 150, 80)
Next X
For Y = 0 To 200 Step 50
Graph1.DrawString(Y - 100, drawFont, drawBrush,
150, 200 - Y)
Next Y
'Преобразование компьютерной системы координат
в математическую систему координат
Graph1.ScaleTransform(1, -1) 'Леворот оси Y
Graph1.TranslateTransform(150, -100) 'Сдвиг по
осям X и Y
'Рисование осей с засечками в математической
системе координат
Graph1.DrawLine(Pen1, -150, 0, 150, 0) 'Ось X
Graph1.DrawLine(Pen1, 0, -100, 0, 100) 'Ось Y
For X = -150 To 150 Step 50 'Засечки на оси X
Graph1.DrawLine(Pen1, X, -5, X, 5)

```

```

Next X
For Y = -100 To 100 Step 50 'Засечки на оси Y
Graph1.DrawLine(Pen1, -5, Y, 5, Y)
Next Y
End Sub

```

Проект «Система координат» хранится в папке `..informatika9\Система координат` Windows-CD 

Контрольные вопросы

1. Перечислите методы рисования графических фигур и их аргументы.
2. Каким образом можно изменить систему координат формы или графического поля?

Задания для самостоятельного выполнения

Windows-CD 

- 2.26. *Практическое задание.* В системе программирования Visual Basic 2005 создать проект «Графический редактор». Предусмотреть очистку графического поля с использованием метода `Graph1.Clear(Color.White)`.
- 2.27. *Практическое задание.* В системе программирования Visual Basic 2005 создать проект «Система координат».

2.7.2. *Анимация в языке Visual Basic

Для создания анимации (иллюзии движения на экране какого-либо объекта) применяется принцип смены кадров (изображений), как это делается в мультипликации. Для этого необходимо с определенной частотой рисовать объект в поле рисования, причем координаты объекта должны каждый раз изменяться на определенную величину.

Проект «Анимация». Разработать проект, в котором реализуется «полет бабочки». Для создания иллюзии взмаха крыльями два изображения бабочки («с развернутыми крыльями» и «с свернутыми крыльями») с определенной частотой выводятся в поле рисования. Для создания иллюзии движения при каждом взмахе координаты изображения изменяются на определенную величину.

Создадим графический интерфейс проекта (рис. 2.33).

1. Поместить на форму:

- графическое поле `PictureBox1`, которое будет использоваться в качестве области рисования;
- объект `Timer1`, для периодического вывода изображения на графическое поле, который вызывает событие `Tick` через определенные пользователем интервалы времени.

Периодичность события `Tick` может быть задана в свойстве `Interval`, измеряемом в миллисекундах (может изменяться от 0 до 65 535).

2. Выделить объект `Timer1` и с помощью диалогового окна *Свойства* присвоить свойству `Interval` значение 100, а свойству `Enabled` — значение `True`.

Создадим заготовку событийной процедуры щелчком по объекту `Timer1`.

```
3. Private Sub Timer1_Tick(...) End Sub
```

```
End Sub
```

Выберем в качестве области рисования графическое поле `PictureBox1`.

Используем в качестве изображения бабочки «с развернутыми крыльями» графический файл `bfly1.bmp`, а изображения «со свернутыми крыльями» — графический файл `bfly1.bmp`.

Объявим графические объекты `Image1` и `Image2` и создадим их из файлов с помощью функции `Image.FromFile()`. Аргументом функции является путь к графическому файлу, а также его имя.

Для того чтобы при каждом событии `Tick` выводить попеременно то одно, то другое изображение бабочки, используем логическую переменную `flg1`. При каждом событии с помощью оператора `If-Then-Else` в однострочной форме будем менять ее значение с `True` на `False` или, наоборот, с `False` на `True`. Затем с использованием оператора `If-Then-Else-End If` в многострочной форме будем выводить в зависимости от значения логической переменной `flg1` то или иное изображение бабочки.

Вывод изображения на область рисования будем осуществлять с помощью метода `DrawImage(Image1, X, Y)`,

Глава 3

аргументами которого являются изображение и координаты его левого верхнего угла на области рисования.

Для реализации перемещения из нижнего левого угла графического поля в правый верхний угол сместим ось координат `Y` вниз на 200 точек и при каждом событии будем увеличивать координату `X` и уменьшать координату `Y`.

4. В результате событийная процедура примет следующий вид:

```
Dim Graph1 As Graphics
Dim Image1, Image2 As Image
Dim X, Y As Single
Dim flg1 As Boolean
Private Sub Timer1_Tick(...)
    Graph1 = Me.PictureBox1.CreateGraphics()
    Image1 = Image.FromFile("\informatika9\
Анимация\bfly1.bmp")
    Image2 = Image.FromFile("\informatika9\
Анимация\bfly2.bmp")
    If flg1 Then flg1 = False Else flg1 = True
    Graph1.TranslateTransform(0, 200)
    If flg1 Then
        Graph1.DrawImage(Image1, X, Y)
    Else
        Graph1.DrawImage(Image2, X, Y)
    End If
    X = X + 2
    Y = Y - 2
End Sub
```

5. Запустить проект. В графическом поле начнется «порхание» бабочки из нижнего левого угла графического поля в правый верхний угол.


Для изменения частоты «порхания» необходимо изменить `y` объекта `Timer1` значение свойства `Interval`.

Для изменения скорости полета необходимо изменить шаг изменения координат (см. рис. 2.33).



Рис. 2.33. Проект «Анимация»

Проект «Анимация» хранится в папке
..\informatika9\Анимация

Windows-CD 

Контрольные вопросы

1. Каковы основные этапы создания анимации движения объекта?

Задания для самостоятельного выполнения

Windows-CD 

- 2.28. **Практическое задание.* В системе программирования Visual Basic 2005 создать проект «Анимация». Изменить частоту «порхания» и скорость полета бабочки.

**Практические работы компьютерного практикума,
рекомендуемые для выполнения в процессе
изучения главы 2**

Компьютерный практикум

- № 7. Разработка проекта «Выбор цвета» в системе программирования Visual Basic.
- № 8. Разработка проекта «Текстовый редактор» в системе программирования Visual Basic.
- № 9. Разработка проекта «Движение Земли» в системе программирования Visual Basic.