

Microsoft® Small Basic

Изучение фигур

Предполагаемое время работы с этим уроком: 1 час



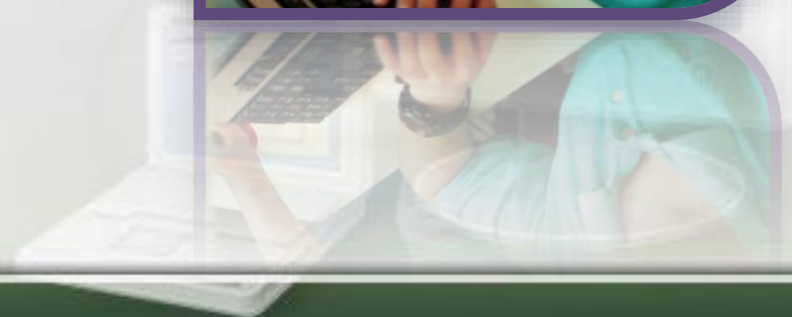
Изучение фигур

В этом уроке вы изучите следующее.

Создание фигур с использованием объекта **Shapes**.

Использование различных операций объекта **Shapes**.

Анимирование фигур на экране.

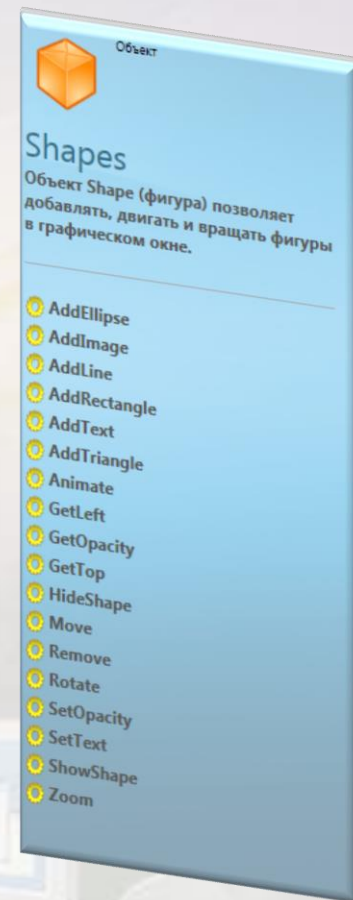


Знакомство с объектом Shapes

Пока вы изучили создание шаблонов в Small Basic с использованием объектов **GraphicsWindow** и **Turtle**.

В этом уроке будет представлен объект **Shapes** Small Basic! Этот объект позволяет рисовать, поворачивать и анимировать фигуры в графическом окне.

Можно назначить фигурам цвета, используя определенные свойства объекта **GraphicsWindow**.



Операции объекта Shapes

Благодаря использованию определенных операций объекта **Shapes** созданным фигурам можно придать яркий внешний вид. Поддерживаются следующие операции:

- **AddImage**
- **AddRectangle**
- **HideShape**
- **ShowShape**
- **SetOpacity**
- **GetOpacity**
- **Move**
- **Animate**
- **Zoom**



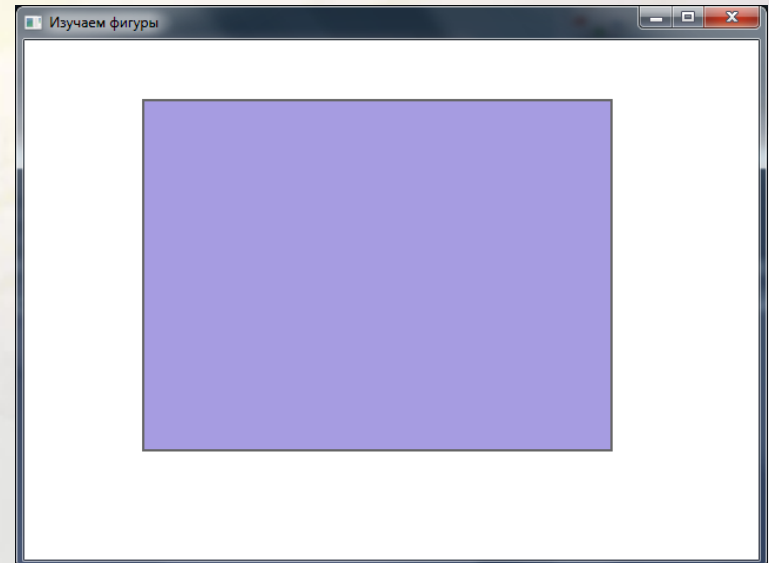
Операции объекта Shapes

Рассмотрим пример демонстрации этих операций...

```
rectangle = Shapes.AddRectangle(400, 300)
Shapes.Move(rectangle, 100, 50)
For i = 1 To 10
    Shapes.ShowShape(rectangle)
    Program.Delay(1000)
    Shapes.HideShape(rectangle)
    Shapes.SetOpacity(rectangle, 100 - i * 10)
    Program.Delay(800)
EndFor
```

В этом примере мы использовали операции **ShowShape**, **HideShape**, и **SetOpacity** объекта **Shapes** для выполнения различных действий с прямоугольником.

ВЫВОД



Операции объекта Shapes

Рассмотрим эти операции подробно...

AddRectangle— с помощью этой операции можно определить прямоугольник и указать имя, ширину и высоту прямоугольника.

```
rectangle = Shapes.AddRectangle(150, 100)
```

```
Shapes.HideShape(rectangle)  
Shapes.ShowShape(rectangle)
```

HideShape— с помощью этой операции можно скрыть фигуру и указать ее имя.

ShowShape— с помощью этой операции можно отобразить фигуру и указать ее имя.

SetOpacity— с помощью этой операции можно установить прозрачность фигуры и указать ее имя и уровень прозрачности от 0 до 100.

GetOpacity— с помощью этой операции можно вернуть прозрачность фигуры и указать ее имя.

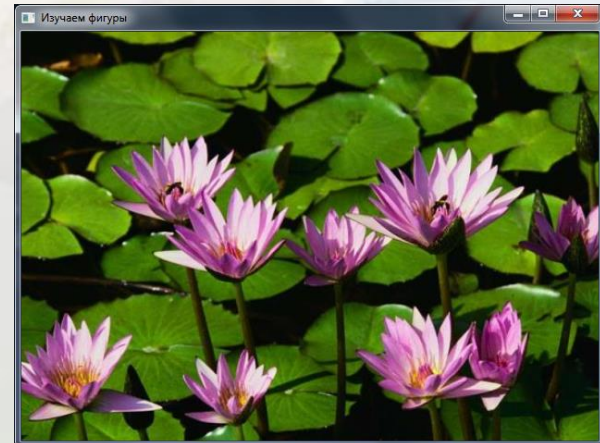
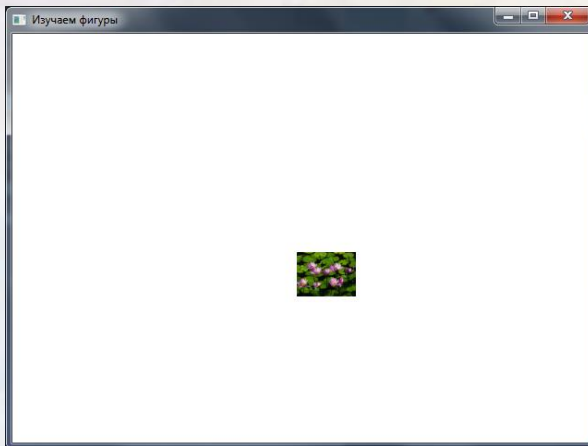
```
Shapes.SetOpacity(rectangle, 50)  
Shapes.GetOpacity(rectangle)
```

Операции объекта Shapes

Рассмотрим другой пример, чтобы продемонстрировать другие операции...

```
imagepath = "C:\Small Basic\Water lilies.jpg"  
image = Shapes.AddImage(imagepath)  
Shapes.Move(image, 5, 5)  
Shapes.Animate(image, 20, 20, 1000)  
Shapes.Zoom(image, 0.1, 0.1)  
For i = 0 To 1 Step 0.1  
    Program.Delay(1000)  
    Shapes.Zoom(image, 0.1 + i, 0.1 + i)  
EndFor
```

В этом примере мы использовали операцию **AddImage** для отображения изображения. Затем мы использовали операции **Move**, **Animate** и **Zoom** для выполнения различных действий с изображением.



Операции объекта Shapes

AddRectangle - с помощью этой операции можно добавить прямоугольник, который появится в графическом окне.

```
rectangle = Shapes.AddRectangle(150, 100)
```

```
Shapes.Move(rectangle, 125, 125)
```

Move — с помощью этой операции можно переместить фигуру в другое место графического окна. Необходимо указать имя фигуры и координаты x и y нового местоположения.

Animate — эта операция используется для анимации фигуры при перемещении в другое местоположение. Необходимо указать имя фигуры, координаты x и y нового местоположения и продолжительность анимирования.

```
Shapes.Animate(rectangle, 30 * i, 150, 5000)
```

```
Shapes.Zoom(rectangle, 2, 2)
```

Zoom — операция Zoom используется для увеличения или уменьшения фигуры до определенного уровня. Необходимо указать имя фигуры и уровень масштаба от 0.1 до 20.

Операции объекта Shape

Можно использовать объект **Shapes** для добавления различных типов фигур в программе.

Затем можно использовать различные операции для объекта **Shapes**, такие как перемещение фигуры, установка прозрачности или добавления эффекта масштабирования. Рассмотрим пример...

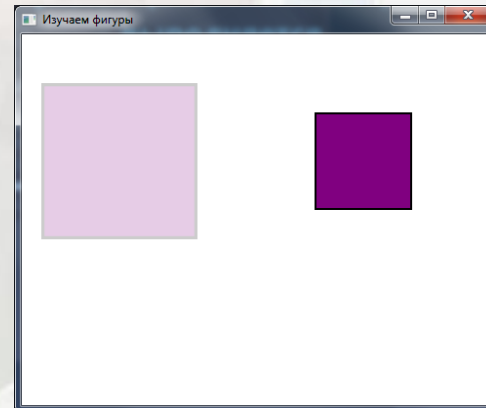
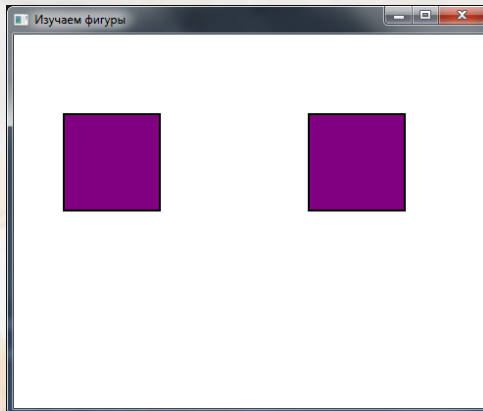
Нажмите



Запуск (F5)

на панели инструментов.

```
GraphicsWindow.PenWidth = 2
GraphicsWindow.PenColor = "Black"
GraphicsWindow.BrushColor = "Purple"
rectangle1 = Shapes.AddRectangle(100, 100)
Shapes.Move(rectangle1, 50, 80)
rectangle2 = Shapes.AddRectangle(100, 100)
Shapes.Move(rectangle2, 300, 80)
For i = 1 To 4
    Program.Delay(1000)
    Shapes.Zoom(rectangle1, i * 0.4, i * 0.4)
    Shapes.SetOpacity(rectangle1, i * 5)
EndFor
```

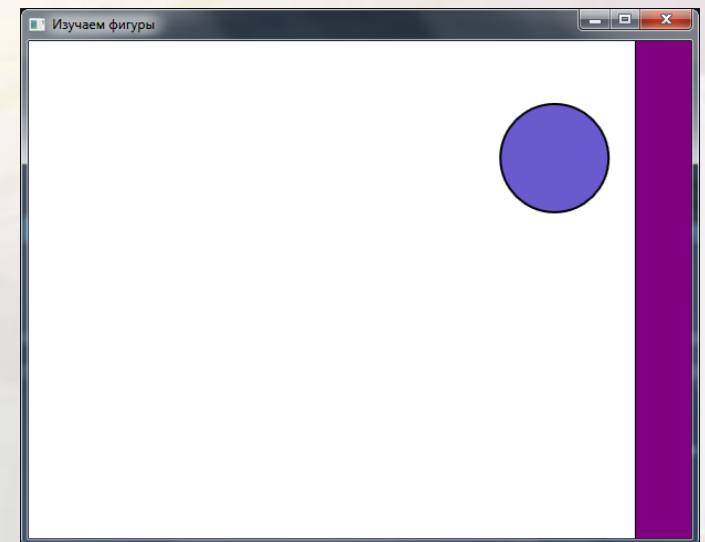


Анимация фигуры

Рассмотрим пример анимации фигуры с использованием объекта **Shapes**.

В этом примере выполняется анимация фигуры из исходного местоположения в другое и обратно в графическом окне.

```
Sball = Shapes.AddEllipse(100, 100)
Shapes.Move(Sball, 0, 340)
x = 450
GraphicsWindow.DrawRectangle(550, 0, 80, 450)
GraphicsWindow.BrushColor = "Purple"
GraphicsWindow.FillRectangle(550, 0, 80, 450)
Shapes.Animate(Sball, x, 40, 490)
Program.Delay(500)
If (Shapes.GetLeft(Sball) = x) Then
    Shapes.Animate(Sball, 0, 340, 500)
EndIf
```



Поворот фигуры

Рассмотрим некоторые другие операции объекта **Shapes**, написав программу для поворота фигуры.

В этом примере используется цикл **For** для поворота фигуры в исходном положении в графическом окне.

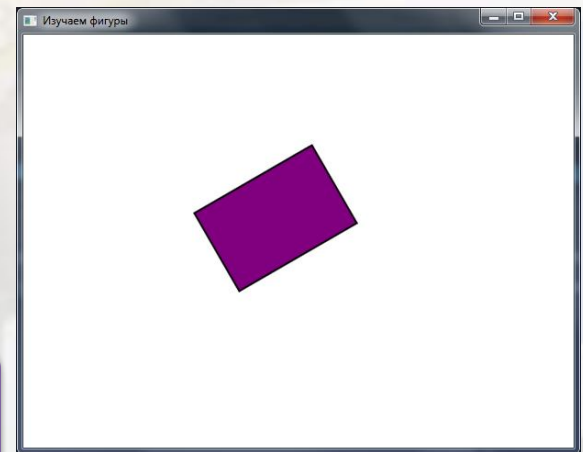
```
GraphicsWindow.BrushColor = "Purple"  
rotateshape = Shapes.AddRectangle(150, 100)  
Shapes.Move(rotateshape, 200, 150)  
For i = 0 To 12  
    Shapes.Rotate(rotateshape, 30 * i)  
    Program.Delay(1000)  
EndFor
```

Нажмите  на панели инструментов.

Запуск (F5)

При выполнении программы в графическом окне поворачивается прямоугольник.

ВЫВОД



Эксперименты с фигурами

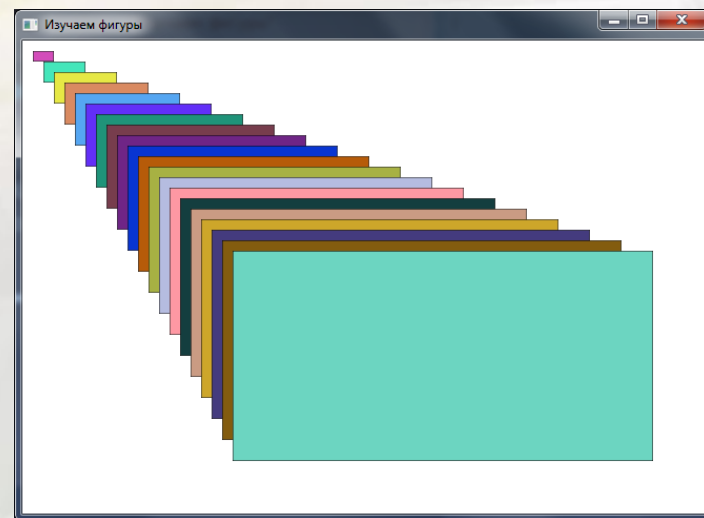
Помимо рисования фигур различных стилей и размеров, также можно создавать уникальные фигуры, используя условия и циклы в программе.

Например, можно использовать цикл **For** для создания нескольких прямоугольников случайных цветов...



```
For i = 0 To 20  
    GraphicsWindow.PenWidth = "0.5"  
    GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()  
    rectangle1 = Shapes.AddRectangle(i * 20, i * 10)  
    Shapes.Move(rectangle1, i * 10, i * 10)  
EndFor
```

ВЫВОД





Поздравляем! Вы изучили следующее.

- ✚ Создание фигур с использованием объекта **Shapes**.
- ✚ Использование различных операций объекта **Shapes**.
- ✚ Анимация фигур на экране.

Продемонстрируйте свои знания

Написание программы для отображения графического окна и выполнения следующих действий.

- ❖ Добавление линии и круга в окно.
- ❖ Установка цвета, размера и местоположения фигур.
- ❖ Анимация круга, чтобы он перемещался по линии от левой к правой части графического окна.

