

ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКОЙ БИБЛИОТЕКИ WPFOBJECTS В PASCALABC.NET ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

Описывается подход к обучению основам объектно-ориентированного и событийного программирования, базирующийся на векторном графическом модуле WPFOjects системы программирования PascalABC.NET.

We describe an approach how to teach the basics of object-oriented events-based programming. This approach is based on vector graphic module WPFOjects of PascalABC.NET programming system.

Тема графики занимает важное место при обучении программированию начинающих. Данная работа продолжает тематику использования современных графических модулей при обучении программированию школьников. В качестве основы обучения программированию используется система программирования PascalABC.NET, разрабатываемая под научным руководством автора, реализующая современный Паскаль. Основанная на этой системе методика обучения многократно представлялась на научных и методических конференциях и в центральной печати [1-4].

В PascalABC.NET на протяжении более 10 лет для начального обучения объектно-ориентированному программированию использовался модуль векторной графики ABCObjects. В 2017-18 годах был создан усовершенствованный модуль WPFOjects, о котором пойдёт речь в данной статье. Он основан на современной графической библиотеке WPF, что позволило сделать его существенно проще устаревшего модуля ABCObjects, насыщенным возможностями современного обучения программированию и заточенным на создание компактных и эффектных анимаций, а также простых 2D-игр. Данный модуль активно используется в учебной программе юных программистов в Воскресной компьютерной школе Института математики, механики и компьютерных наук ЮФУ [5].

При создании модуля WPFOjects преследовались следующие цели:

1. Сделать максимально простым создание векторных графических объектов и управление их свойствами. Реализовать простую возможность задания перемещения и поворота векторного графического объекта.
2. Реализовать глобальные события мыши и клавиатуры для создания интерактивных программ и простейших игр.
3. Реализовать взаимодействие между графическими объектами: попадание точки в объект, пересечение графических объектов. Рассмотрим основные новшества на примерах.

1. Создание объектов и простейшая анимация. Для создания графического объекта используется традиционный в ООП вызов конструктора:

```
var s := new SquareWPF(100,200,50,Colors.Green);
```

При этом на графическом экране рисуется квадрат, и далее им можно управлять с помощью изменения свойств и вызова методов. Используя свойство `s.Left`, будем перемещать объект вправо в цикле, а используя метод `s.Rotate`, дополнительно вращать его вокруг центра. Код соответствующей программы выглядит крайне просто:

```
loop 400 do
begin
s.Left += 1;
s.Rotate(1);
end;
```

При этом анимация осуществляется без мерцания.

2. Использование событий мыши и клавиатуры. В модуле `WPFObjects` используются глобальные события мыши и клавиатуры, позволяющие добавить в программу простую интерактивность. Следующий код позволяет управлять графическим объектом с клавиатуры, перемещая и вращая его:

```
var s := new SquareWPF(100,200,50,Colors.Green);
OnKeyDown := procedure(k) -> begin
case k of
Key.Left: s.MoveOn(-2,0);
Key.Right: s.MoveOn(2,0);
Key.Up: s.MoveOn(0,-2);
Key.Down: s.MoveOn(0,2);
Key.A: s.Rotate(-2);
Key.D: s.Rotate(2);
end;
end;
```

В программе используется событие `OnKeyDown`, срабатывающее при нажатии клавиши. К нему с помощью присваивания привязан обработчик, задаваемый лямбда-выражением – безымянной процедурой, создаваемой на лету в коде основной программы.

Далее приведем код, который позволяет перетаскивать объекты мышью. Для реализации такой программы нам потребуется определить два обработчика: в обработчике события `OnMouseDown` мы в переменную `obj` захватим объект под курсором, а в обработчике `OnMouseMove` если удерживается кнопка мыши и объект `obj` не пуст, то его центр следует за координатами курсора мыши:

```
var s := new SquareWPF(100,200,50,Colors.Green);
var s1 := new SquareWPF(300,200,50,Colors.Red);
var ob: ObjectWPF;
OnMouseDown := procedure(x,y,mb) ->
ob := ObjectUnderPoint(x,y);
OnMouseMove := procedure(x,y,mb) ->
if (mb = 1) and (ob<>nil) then
ob.Center := Pnt(x,y);
```

3. Взаимодействие между объектами. Каждый WPF объект может определять список тех объектов, с которыми он в данный момент

пересекается, используя метод `IntersectionList`. Создадим вначале в графическом окне 500 случайных прямоугольников:

```
loop 500 do
    new RectangleWPF(Random(800), Random(600),
        Random(200), Random(200), RandomColor);
```

Сбросим на них бомбу, представляющую большую черную окружность по центру экрана:

```
var Destroyer := new CircleWPF(Window.Center, 100, Colors.Black);
```

Бомба должна уничтожить все объекты, которых она касается. Для этого воспользуемся следующим циклом:

```
foreach var o in Destroyer.IntersectionList do
    o.Destroy;
```

В конце программы мы можем выдать количество оставшихся объектов:

```
Print(Objects.Count);
```

Рассмотренные примеры иллюстрируют лёгкость и компактность применения высокоуровневых графических средств школьниками для создания простых анимационных и игровых программ, позволяя писать им интуитивно ясный и простой код.

Модуль `WPFObjets` и приведенные примеры используются в программе занятий учеников первого года обучения программированию в Воскресной компьютерной школе мехмата ЮФУ [5]. Практика их использования показала хорошее усвоение темы «Объектно-ориентированное программирование», а также повышение мотивации учащихся при обучении программированию за счет воплощения своих креативных идей в виде простого кода на языке программирования `PascalABC.NET`.

Литература

1. Михалкович С.С. Система программирования `PascalABC.NET`: 10 лет развития / Труды XX Научно-методической конференции «Современные информационные технологии в образовании: Южный Федеральный округ». Ростов н/Д: Изд-во ЮФУ, 2013. С. 69–71.

2. Бондарев И.В., Михалкович С.С. Система программирования `PascalABC.NET`: 15 лет развития / Труды XXV Научно-методической конференции «Современные информационные технологии в образовании: Южный Федеральный округ». Ростов н/Д: Изд-во ЮФУ, 2018. С. 69-71.

3. Бордюгова Т.Н., Михалкович С.С. Анализ необходимости внедрения системы программирования `PascalABC.NET` в школьный курс информатики на базовом и углубленном уровнях / Труды XXIII Научно-методической конференции «Современные информационные технологии в образовании: Южный Федеральный округ». Ростов н/Д: Изд-во ЮФУ, 2016. С. 72-75.

4. Абрамян М.Э., Михалкович С.С. Веб-среда разработки и обучения // Открытые системы. СУБД. 2012, № 10. С. 56–59.

5. Сайт Воскресной компьютерной школы мехмата ЮФУ. <http://компьютернаяшкола.рф> (электронный ресурс, дата обращения 08.09.18).