

ИСПОЛЬЗОВАНИЕ НОВОЙ ГРАФИЧЕСКОЙ БИБЛИОТЕКИ GRAPHWPF В PASCALABC.NET ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ ШКОЛЬНИКОВ

Описывается подход к обучению школьников основам программирования графики, базирующийся на графическом модуле GraphWPF системы программирования PascalABC.NET.

We describe an approach how to teach the basics of graphic programming for middle school pupils. This approach is based on graphic module GraphWPF of PascalABC.NET programming system.

Тема графики занимает особое место при обучении программированию начинающих. Так, Logo – один из первых языков, создававшихся специально для обучения программированию, – в своей основе имеет команды черепашьей графики. Разумеется, роль графики при обучении программированию менялась от просто рисования векторных картинок до визуализации результатов и всевозможных анимаций.

Система программирования PascalABC.NET реализует современный Паскаль, многократно представлялась на научных и методических конференциях и в центральной печати [1-4]. В PascalABC.NET на протяжении последних 10 лет для обучения написанию графических программ использовался упрощенный модуль растровой графики GraphABC. Однако изменение графических средств в других языках привело к необходимости переписать этот модуль с целью его осовременивания и лучшего встраивания в учебную программу юных программистов в Воскресной компьютерной школе Института математики, механики и компьютерных наук ЮФУ [5]. В 2017 году был создан усовершенствованный модуль GraphWPF, о котором пойдёт речь в данной статье. Он основан на современной графической библиотеке WPF, что позволило сделать его не только компактным и простым, но и насыщенным новыми возможностями.

При создании данного модуля преследовались следующие цели:

1. Упростить написание графических программ с учетом следующих средств, используемых для обучения начинающих: подпрограммы, динамические массивы, объекты и точечная нотация, обработчики событий и действия, хранящиеся в переменных.
2. Обеспечить при изучении графики повторение по всем изученным до этого разделам.
3. Сделать предельно простым создание анимационных программ
4. Сделать предельно простым рисование в традиционной для школьников декартовой системе координат с крупным масштабом (15-20 единиц на графическое окно).

Рассмотрим основные новшества на примерах.

1. Вещественные координаты. Все параметры графических команд – вещественные, что существенно облегчает масштабирование и деление на несколько частей. Рассмотрим код, рисующий многоугольник, оформленный в виде процедуры:

```
procedure Многоугольник(x0,y0,r: real; n: integer);  
begin  
  var a := Pi / 2;  
  MoveTo(x0 + r * Cos(a), y0 - r * Sin(a));  
  loop n do  
    begin  
      a += 2 * Pi / n;  
      LineTo(x0 + r * Cos(a), y0 - r * Sin(a));  
    end;  
end;
```

Нетрудно убедиться, что использование вещественных координат убирает необходимость округления и упрощает код.

2. Математическая система координат с отображением координатной сетки. Для перехода к математической системе координат с отображением координатной сетки используется простейший вызов:

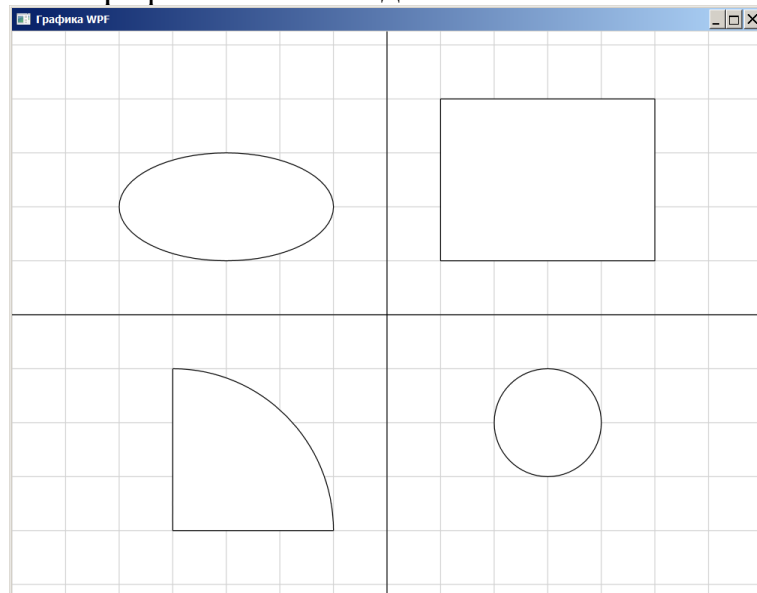
```
SetMathematicCoords(-7, 7);
```

Параметры здесь задают отображаемый диапазон координат по оси OX. После этого рисование графических примитивов существенно упрощается.

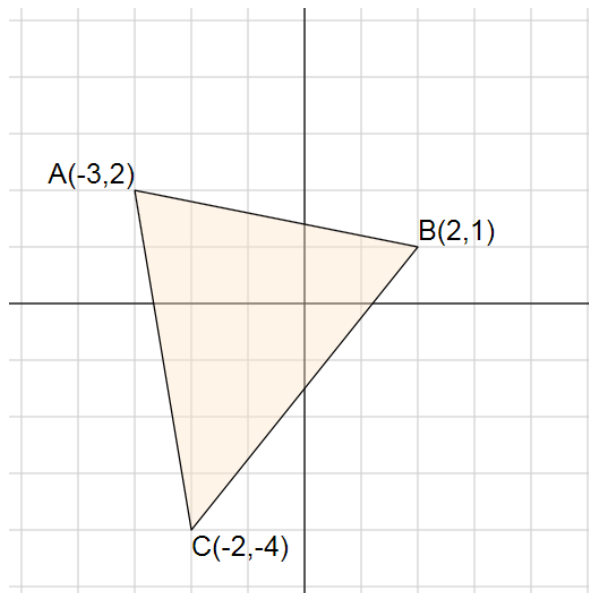
Так, следующий простой код

```
Circle(3, -2, 1);  
Ellipse(-3, 2, 2, 1);  
Rectangle(1, 1, 4, 3);  
Sector(-4, -4, 3, 0, 90);
```

позволяет получить графический вывод:



3. Использование кортежей для задания точек. Для рисования произвольных многоугольников используется процедура Polygon с параметром, являющимся массивом точек. Точки могут быть представлены двухэлементными кортежами, например: (1,2). Так, следующий треугольник



МОЖНО ПОЛУЧИТЬ, ИСПОЛЬЗУЯ КОД:

```
SetMathematicCoords;
Polygon(Arr((-3,2), (2,1), (-2,-4)));
TextOut(-3, 2, 'A(-3,2)', Alignment.RightBottom);
TextOut(2, 1, 'B(2,1)', Alignment.LeftBottom);
TextOut(-2, -4, 'C(-2,-4)', Alignment.LeftTop);
```

В данном примере следует также обратить на константы выравнивания текста, сосредоточенные в типе `Alignment`, позволяющие легко получить реалистичный графический вывод, похожий на изображения в учебниках по геометрии.

3. Анимация на основе кадра. Для анимаций средней сложности, содержащих достаточно большое количество объектов, проблемой является производительность и отсутствие мерцания. Одним из распространенных приёмов анимации в этом случае является анимация на основе кадра, когда пишется специальная процедура, рисующая текущий кадр на пустом экране, после чего она автоматически вызывается графической системой. Приведем простейший пример, позволяющий анимировать круг, перемещая его влево:

```
var x: integer := 0;
procedure Draw;
begin
  Circle(x,250,20);
  x += 1;
end;
begin
  BeginFrameBasedAnimation(Draw);
end.
```

С использованием лямбда-выражений и модификации, нумерующей кадры, весь код можно перенести в основной блок и существенно упростить, записав его одной строкой:

```
BeginFrameBasedAnimation(x->Circle(x,250,20));
```

Здесь на вход процедуре `BeginFrameBasedAnimation` передаётся действие `Draw`, но с одним параметром `x`, задающим номер кадра.

Приведем еще один пример, иллюстрирующий создание слайд-шоу из фотографий, сменяющих друг друга каждую секунду:

```
var a := Arr('Pic1.png', 'Pic2.png', 'Pic3.png');  
BeginFrameBasedAnimation(  
  i->DrawImage(0,0,200,200,a[i mod a.Length]),1);
```

Здесь имена файлов изображений хранятся в массиве `a`, процедура `DrawImage` рисует текущее изображение в прямоугольнике размера 200x200, а смена индекса по закону `i mod a.Length` призвана обеспечить циклическое воспроизведение изображений в массиве `a`.

Рассмотренный модуль `GraphWPF` и приведенные примеры используются в программе занятий учеников первого года обучения программированию в Воскресной компьютерной школе мехмата ЮФУ [5]. Практика их использования показала существенно лучшее усвоение темы «Программирование графики» за счет наличия высокоуровневых концепций, позволяющих писать интуитивно ясный и простой код.

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: новые возможности 2015-16 гг. / Труды XXIII Научно-методической конференции «Современные информационные технологии в образовании: Южный Федеральный округ». Ростов н/Д: Изд-во ЮФУ, 2016. 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 10.09.17).