

Технология компонентного программирования

Лекция 2.5

Составитель: Эверстов В.В.

Дата составления: 23/02/2010

Дата модификации: 25/02/2010

Определения

- При создании приложений очень удобно группировать взаимосвязанные типы в специальные пространства имен. В С# подобное делается с помощью ключевого слова **namespace**

namespace

```
using System;
namespace MyShapes
{
    //Класс Circle
    class Circle{/* методы и свойства*/}
    //Класс Hexagon
    class Hexagon{/* методы и свойства*/}
    //Класс Square
    class Square{/* методы и свойства*/}
}
```

```
//File circle.cs
using System;
namespace MyShapes
{
    //Класс Circle
    class Circle{/* методы и свойства*/}
}

//File hexagon.cs
using System;
namespace MyShapes
{
    //Класс Hexagon
    class Hexagon{/* методы и свойства*/}
}

//File square.cs
using System;
namespace MyShapes
{
    //Класс square
    class Square{/* методы и свойства*/}
}
```

ИСПОЛЬЗОВАНИЕ

```
// Используем типы, определенные  
// в пространстве имен MyShape.  
using System;  
using MyShapes;  
  
namespace MyApp  
{  
    class ShapeTester  
    {  
        static void Main(string[] args)  
        {  
            Hexagon h = new Hexagon();  
            Circle c = new Circle();  
            Square s = new Square();  
        }  
    }  
}
```

Полностью квалифицированное ИМЯ ТИПА

```
// Обратите внимание на то, что мы больше не используем
// ключевое слово using для ссылки на пространство
// имен MyShapes.
using System;

namespace MyApp
{
    class ShapeTester
    {
        static void Main(string[] args)
        {
            MyShapes.Hexagon h = new MyShapes.Hexagon();
            MyShapes.Circle c = new MyShapes.Circle();
            MyShapes.Square s = new MyShapes.Square();
        }
    }
}
```

Полностью квалифицированное ИМЯ ТИПА

```
// Еще одно пространство имен с классами фигур...
using System;
namespace My3DShapes
{
    // Класс Circle, способный визуализировать круг
    // в трехмерном формате.
    class Circle{ }

    // Класс Hexagon, способный визуализировать многоугольник
    // в трехмерном формате.
    class Hexagon{ }

    // Класс Square, способный визуализировать квадрат
    // в трехмерном формате.
    class Square{ }
}
```

```
// Полно неоднозначных моментов!
using System;
using MyShapes;
using My3DShapes;

namespace MyApp
{
    class ShapeTester
    {
        static void Main(string[] args)
        {
            // О каком пространстве имен идет речь?
            Hexagon h = new Hexagon(); // Компилятор выдаст ошибку!
            Circle c = new Circle(); // Компилятор выдаст ошибку!
            Square s = new Square(); // Компилятор выдаст ошибку!
        }
    }
}
```

// Теперь мы устранили неоднозначность.

```
static void Main(string[] args)
{
    My3DShapes.Hexagon h = new My3DShapes.Hexagon();
    My3DShapes.Circle c = new My3DShapes.Circle();
    MyShapes.Square s = new MyShapes.Square();
}
```

Вложенные пространства

```
// Вложенные пространства имен
Namespace Lecture3.My3DShapes
{
    //класс Circle
    class Circle{....}
    // класс Hexagon
    class Hexagon{...}
    // класс Square
    class Square{....}
}
```

Сборки в .NET

- Сборка - самоописываемый двоичный файл, исполняемый на CLR.
- Приложения в .NET создаются за счет складывания вместе некоторого числа сборок. Преимущества использования сборок:
 - Сборки повышают возможность повторного использования кода,
 - Сборки определяют границы типов
 - Сборки являются самоописываемыми
 - Сборки поддаются конфигурации

Однофайловая сборка

