

# Пространства имен

**Пространства имен (*namespace*)** — это способ, благодаря которому .NET избегает конфликтов имен между типами (классами, структурами и т.д.)

Пространство имен — это не более чем группа типов данных, но дающая тот эффект, что имена всех типов данных в пределах пространства имен автоматически снабжаются префиксом - названием пространства имен.

Пространства имен можно вкладывать друг в друга.

Платформа .NET требует, чтобы все имена были объявлены в пределах пространства имен;

**Внимание!!!** Если пространство имен не указано явно, тип будет добавлен к безымянному глобальному пространству имен.

Глобальное пространство имен соответствует имени проекта или настраивается в свойствах проекта.

**Рекомендации:** необходимо применять хотя бы два вложенных пространства имен:

- **первое** — наименование вашей компании,
- **второе** — название технологии или пакета программного обеспечения, к которому относится класс, чтобы это выглядело примерно так:

***MyCompany.SomeNamespace.MyClass.***

В большинстве случаев такой подход защитит классы вашего приложения от возможных конфликтов с именами классов, написанных разработчиками из других компаний.

Пространство имен в .NET	Описание
<b>System</b>	Внутри пространства имен <i>System</i> содержится множество полезных типов, позволяющих иметь дело с внутренними данными, математическими вычислениями, генерированием случайных чисел, переменными среды и сборкой мусора, а также ряд наиболее часто применяемых исключений и атрибутов
<b>System.Collections</b> <b>System.Collections.Generic</b>	В этих пространствах имен содержится ряд контейнерных типов, а также несколько базовых типов и интерфейсов, которые позволяют создавать специальные коллекции
<b>System.Data</b> <b>System.Data.Common</b> <b>System.Data.EntityClient</b> <b>System.Data.SqlClient</b>	Эти пространства имен применяются для взаимодействия с базами данных с помощью ADO.NET
<b>System.IO</b> <b>System.IO.Compression</b> <b>System.IO.Ports</b>	В этих пространствах содержится много типов, предназначенных для работы с операциями файлового ввода-вывода, сжатия данных и манипулирования портами
<b>System.Reflection</b> <b>System.Reflection.Emit</b>	В этих пространствах имен содержатся типы, которые поддерживают обнаружение типов во время выполнения, а также динамическое создание типов

<b>System.Windows.Forms</b>	В этих пространствах имен содержатся типы, применяемые для построения настольных приложений с использованием исходного набора графических инструментов .NET ( <u>Windows Forms</u> )
<b>System.Windows</b> <b>System.Windows.Controls</b> <b>System.Windows.Shapes</b>	Пространство <i>System.Windows</i> является корневым среди этих нескольких пространств имен, которые представляют собой набор графических инструментов Windows Presentation Foundation (WPF)
<b>System.Linq</b> <b>System.Xml.Linq</b>	В этих пространствах имен содержатся типы, применяемые при выполнении программирования с использованием API-интерфейса LINQ
<b>System.Web</b>	Это пространство имен является одним из многих, которые позволяют создавать веб-приложения ASP.NET
<b>System.ServiceModel</b>	Это пространство имен является одним из многих, которые позволяет применять для создания распределенных приложений с помощью API-интерфейса Windows Communication Foundation (WCF)
<b>System.Threading</b> <b>System.Threading.Tasks</b>	В этом пространстве имен содержатся многочисленные типы для построения многопоточных приложений, способных распределять рабочую нагрузку среди нескольких ЦП.
<b>System.Xml</b>	В этом ориентированном на XML пространстве имен содержатся многочисленные типы, которые можно применять для взаимодействия с XML-данными

# Использование using для подключения статических членов

(C# 6.0)

```
using System;
using static System.Console;
using static System.Math;
using static HelloApp.Geometry;
namespace Example5UsingStatic
{
    class Program
    {
        static void Main(string[] args)
        {
            double radius = 50;
            double result = GetArea(radius);    // Geometry.GetArea
            WriteLine(result);                  // Console.WriteLine
            Read();                             // Console.Read
        }
    }
    class Geometry
    {
        public static double GetArea(double radius)
        {
            return PI * radius * radius;      // Math.PI
        }
    }
}}
```