C#

podstawowe elementy

Piotr Błaszyński

13 lutego 2017

Podstawowe wytyczne (oczywiście jakby coś trzeba było rozjaśnić, to proszę pisać):

użycie przestrzeni nazw, poniższe (użyte na początku pliku źródłowego) daje w obrębie całego pliku dostęp do wszystkich klas, struktur i przestrzeni nazw znajdujących się w przestrzeni nazw **Generic**, która znajduje się w przestrzeni nazw **Collections**, która znajduje się w przestrzeni nazw **System**

```
using System.Collections.Generic;
```

Kod powinno sie umieszczać w przestrzeniach nazw (domyślnie nazwa projektu jest przestrzenią nazw).

```
namespace MyNames
{
//ten kod jest umieszczony w przestrzeni nazw MyNames
}
```

Po użyciu poniższej konstrukcji na początku pliku można używać skrótu:

```
using gen=System.Collections.Generic;

gen.List<double>
```

Nie należy jednak powyższej konstrukcji nadużywać (w zasadzie jest ona przydatna w wypadku, gdy chcemy skrócić zapis, a ta sama nazwa występuje w dwóch używanych przestrzeniach nazw).

Tabelka porównawcza C++ - C#:

Konstrukcja w C/C++	Odpowiednik w C#
vector <double>	List<double> (dokładnie: System.Collections.Generic.List <double>)
vector<double> collection = new vector<double>();	List<double> collection = new List<double>();
od standardu 2011: auto collection = new vector<double>();	od NET 3.0 (VS 2008): var collection = new List<double>();
printf, cout	Console.Writeline oraz dodatkowo String.Format
class ClassName: BaseClass	class ClassName: BaseClass
for(i=0 ; i<10 ; i++)	for(i=0 ; i<10 ; i++)
BOOST_FOREACH(T item, collection) lub for (T item: collection)	foreach (T item in collection)(od wersji .NET 3.0 bardzo promowane foreach (var item in collection))
int count = 0;	int count = 0; (od wersji .NET 3.0 bardzo promowane var count = 0;)
Modyfikatory public, protected, private umieszczane przed fragmentami kodu.	Modyfikatory public, protected, private umieszczane przed konkretnymi elementami kodu (metody, zmienne).
fabs, sin, cos	Math.Abs, Math.Sin, Math.Cos
Linkowanie	Referencje w projekcie (w efekcie też linkowanie)
int main(int argc, char *argv[])	static void Main(string[] args)
. lub -> (wyciągamy ze zmiennej lub spod wskaźnika)	.
int arr[10][10];	int [,]arr = new int[10,10];
metaprogramowanie, makra, itp.	Atrybuty, np.: [CLSCompliant (true)]
Pliki nagłówkowe	Partial classes (oraz using i referencje)