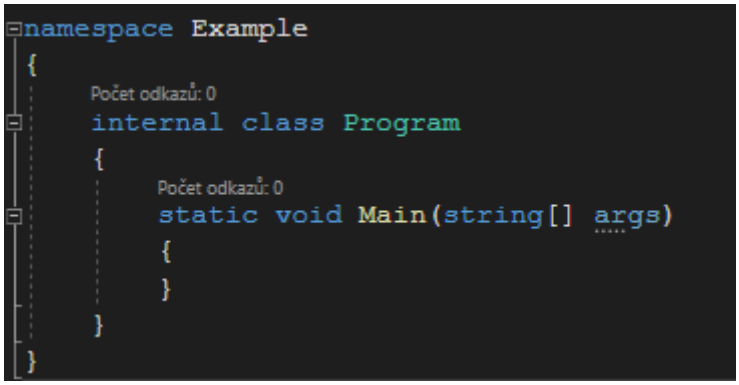
1. Struktura programu v C#, datové typy, přetypování, identifikátory, klíčová slova, základní příkazy jazyka C#. Příklad.

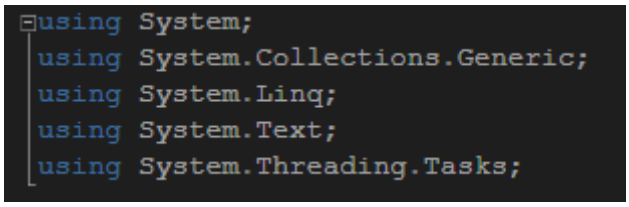
Obsah

Struktura každého programu má „předdefinovanou“ formu a strukturu.

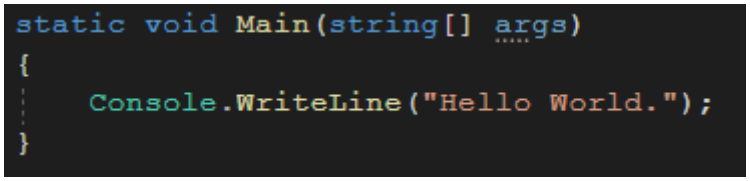
- Deklarace jména prostoru/souboru
  -Slouží k uspořádání a organizaci kódu a prevenci konfliktů jmen.



- Importování „Using“
  -Pomocí klíčového slova Using jsme schopni do našeho kódu přidat různé moduly, které nám umožňují přístup k určitým třídám a funkcím jazyka C#.
  -Například import „using Systém;“ nám umožní přístup k třídě ‚Console‘.



- Metoda ‚Main‘
  -Vstupní bod v programu, který si spouští ihned při otevření/spuštění programu.



Pro lehčí práci s různými hodnotami a textem jsme schopni si vytvořit různé proměnné, u kterých musíme deklarovat datový typ.

Datové typy

- Celá čísla
  - ‚int': 32bitové číslo.
  - ‚long': 64bitové číslo.
  - ‚short': 16bitové číslo.
  - ‚byte': 8bitové číslo.

```
int Int = 999999999;
long Long = 9999999999999999;
short Short = 9999;
byte Byte = 99;
```

- Desetinná čísla
  - ‚float': 32bitové číslo.
  - ‚double': 64bitové číslo.
  - třída Math používá double.

```
float Float = 9.999999999999f;
double Double = 9.99999999999999;
```

- Znak a řetězec
  - ‚char': pouze jeden znak.
  - ‚string': textový řetězec.

```
char znak = 'A';
string text = "Hello, World!";
```

- Boolean (logická hodnota)
  - ‚bool': může nabýt pouze hodnot **True** a **False.**

```
bool pravda = true;
bool nepravda = false;
```

- Datum a čas
  - ‚DateTime': čas našeho zařízení.

```
DateTime dnes = DateTime.Now;
```

- Pole
  - ‚int[]‘: pole celých čísel.
  - ‚string[]‘: pole celých znaků.

```
int[] poleCisel = { 1, 2, 3, 4, 5 };
string[] poleTextu = { "Hello", "World" };
```

- List
  - ‚List<>‘: ukládání a manipulace s kolekcí objektů.
  - List musí obsahovat pouze objekty nebo záznamy stejného datového typu.

```
List<int> seznamCisel = new List<int>();
List<string> seznamTextu = new List<string>();
```

- Přetypování
  - Převedení jedné datové hodnoty na jinou.
  - 2 způsoby

-- Implicitní

- Rozhraní si konverzi provede samo.
- Používáme pouze pokud nehrozí ztráta celé nebo části hodnoty.
- Tato možnost nepotřebuje specifikování datového typu.

```
int celeCislo = 10;
long velkeCislo = celeCislo;  // int na long
```

--Explicitní

- Potřeba specifikovat nový datový typ.
- Používáme pouze pokud typy nejsou implicitně kompatibilní nebo je možnost ztráty hodnoty.

```
double desetinneCislo = 10.5;
int celeCisloExp = (int)desetinneCislo;  // double na int
```

Identifikátory

- Používání pro jmenování proměnných, metod a třeba tříd.
- Jsou limitovány pouze pár pravidly.

- Začínají písmenem nebo podtržítkem:
  - Od A do Z, a do z.
- Diakritika
  - Ačkoli to program vezme jako validní proměnnou tak to paměti zabere o trošku více procesů.
- Nemohou obsahovat speciální znaky:
  - Například ‚@‘ ‚#‘ ‚$‘ nám program nevezme.
- Žádní klíčová slova:
  - Proměnná nesmí mít název klíčového slova.
  - Nejde pojmenovat např. int int = …;
- Velká a malá písmena:
  - Identifikátory dbají i na velké a malé znaky.
  - cislo != Cislo.
- Není omezen počet znaků.

Klíčová slova

- Jsou speciální a vyhrazená slova.
- Každé toto slovo má svůj význam (datový typ, třída, metody ...).
- Nesmíme používat tyto slova jako jména. Program je nebude brát jako validní.
- Nějaké příklady kromě datových typů.

Break	Case	Catch	Class	Else
Finally	For	Foreach	If	New
Null	Object	Override	Private	Public
Return	Switch	Try	Using	void

Základní příkazy v jazyce C#.

- Základní příkazy jsou konstrukce pro ovládání chodu programu.
- Člení se zde třeba volání metod, podmínky, inicializace proměnných a jejich datových typů ...

- Proměnné:

```
int Int = 999999999;
long Long = 9999999999999999;
short Short = 9999;
byte Byte = 99;
```

- Početní operace s proměnnými

```
int cislo = 5;
cislo++; // číslo bude 6
cislo = cislo + celeCisloImp;
cislo = cislo - 15;
```

- Podmínky:

```
int cislo = 5;

if (cislo > 0)
{
    Console.WriteLine("Číslo je kladné.");
}
else if (cislo < 0)
{
    Console.WriteLine("Číslo je záporné.");
}
else
{
    Console.WriteLine("Číslo je nula.");
}
```

- Smyčky

```
for (int i = 0; i < 5; i++)
{
    Console.WriteLine(i);
}

while (cislo > 0)
{
    Console.WriteLine(cislo);
    cislo--;
}
```

- Switch
  - efektivnější verze if, else if při více hodnotách.

```
switch (cislo)
{
    case 1:
        Console.WriteLine("Hodnota je 1.");
        break;
    case 2:
        Console.WriteLine("Hodnota je 2.");
        break;
    default:
        Console.WriteLine("Hodnota není ani 1 ani 2.");
        break;
}
```

- Metody
  - je to zabalený kód, který si můžeme kdykoliv vyvolat pro určitou operaci.

```
void Pozdrav(string jmeno)
{
    Console.WriteLine("Ahoj, " + jmeno + "!");
}

Pozdrav("Jan");
```

- Čtení a zápis do konzole.

```
Console.WriteLine("Zadejte číslo:");
int vstup = int.Parse(Console.ReadLine());
Console.WriteLine("Zadali jste: " + vstup);
```

- Výjimky
  - Ošetření výjimek a bugů v kódu.

```
try
{
    // Kód, který může vyvolat výjimku
}
catch (Exception ex)
{
    Console.WriteLine("Došlo k chybě: " + ex.Message);
}
```