



Vyšší odborná škola obalové techniky  
a střední škola, Štětí

# Digitální učební materiály

Programování - Programování C#

Ivan Pomykacz



evropský  
sociální  
fond v ČR



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání  
pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

# Licence



Digitální učební materiály, jejímž autorem je Ivan Pomykacz, podléhají licenci [Creative Commons: Uvedte autora - Nevyužívejte dílo komerčně - Zachovejte licenci 3.0 Unported](https://creativecommons.org/licenses/by-nc-sa/3.0/).

Vytvořeno na základě tohoto díla: <http://dumy.odbornaskola.cz/pomykacz>

Práva nad rámec této licence jsou popsána zde: <http://dumy.odbornaskola.cz/pomykacz>.

# Disclaimer

Tento PDF dokument byl strojově vygenerován z HTML stránek

<http://dumy.odbornaskola.cz/pomykacz/>.

Je tedy možné, že sazba textu může obsahovat chyby. Jde převážně o vizuální a typografické chyby, které mohou narušit plynulou čitelnost textu. V některých případech může být ohrožena i funkčnost některých komponent (jako vnitřní odkazy).

Vzhledem k tomu, že vypracované materiály nebyly nikdy určeny pro výstupní formát PDF, autor se zříkává jakékoli odpovědnosti za nalezené chyby. Nesnažte se proto v této souvislosti autora kontaktovat.

# **Programování**

## **Programování C#**

# Obsah

- Deklarace metody

# Deklarace metody

<b>Název školy</b>	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
<b>Adresa školky</b>	Kostelní 134, 411 08 Štětí		
<b>IČ</b>	46773509		
<b>Název operačního programu</b>	OP Vzdělávání pro konkurenceschopnost		
<b>Registrační číslo</b>	CZ.1.07/1.5.00/34.1006		
<b>Označení vzdělávacího materiálu</b>	VY_32_INOVACE_23_PRG_460		
<b>Název tématické oblasti (sady)</b>	Programování		
<b>Název materiálu</b>	Deklarace metody		
<b>Anotace</b>	Prezentace deklaráce metody bez/s vstupními argumenty a bez/s návratovou hodnotou. Volání metody.		
<b>Autor</b>	Ivan Pomykacz	<b>Jazyk</b>	čeština
<b>Očekávaný výstup</b>	Popíše metodu dle její deklaráce. Deklaruje metodu. Volá metodu v kódu.		
<b>Klíčová slova</b>	deklaráce metody		
<b>Druh výukového zdroje</b>	Výklad	<b>Věková skupina žáků</b>	15+
<b>Typ interakce</b>	aktivita	<b>Ročník</b>	1.
<b>Speciální vzdělávací potřeby</b>	žádné		
<b>Zhotoveno, (datum/období)</b>	20.11.2013	<b>Celková velikost</b>	

## Obsah

- [Příklad](#)
- [Nová metoda](#)
  - [Implementace](#)
- [Volání metody](#)
  - [Opakované volání](#)

## Příklad

V kódu níže je deklarována jen jedna metoda. Jmenuje se `Main()`.

```
using System;
```

```
namespace Deklarace
```

```
{
    class MainClass
    {
        public static void Main (string[] args)
        {
            Console.WriteLine ("Hello World!");
        }
    }
}
```

```
    }  
  }  
}
```

Můžeme o ní říci, že:

- Je veřejná (`public`).
- Je statická (`static`).
- Vrací `void`, tedy nevrací nic.
- Má jeden argument typu `string[]` a názvem `args`.
- Je součástí třídy (class) `MainClass`, a ta je součástí jmenného prostoru (namespace) `Deklarace`.  
Vše (namespace, class a metoda) má své složené závorky `{}`.

Vzhledem k tomu, že mi se ještě nebudeme zabývat plně objektovým programováním, budeme některá vyhrazená slova (`static` a `public`) tiše přecházet.

Navíc všechny metody budeme psát do jediné třídy, zde `MainClass`.

## Nová metoda

Provedme deklaraci nové metody s názvem `Info`. Metoda zobrazí název našeho programu a základní informace o něm. Nebude žádný vstupní argument a nebude nic vracet.

```
public static void Info()  
{  
  
}
```

Klíčová slova `public` a `static` budeme "tiše" psát před každou nově deklarovanou metodou, dokud si neřekneme, co to znamená.

Dávejte dobrý pozor, do kterého místa v kódu metodu vpisujete.

```
using System;
```

```
namespace Deklarace  
{  
    class MainClass  
    {  
        public static void Main (string[] args)  
        {  
            Console.WriteLine ("Hello World!");  
        }  
  
        public static void Info()  
        {  
  
        }  
    }  
}
```

## Implementace

Implementace spočívá v tom, že napíšeme kód, který provede požadovanou akci.

V našem případě jde o to vypsát informace o programu.

```
public static void Info()  
{  
    Console.WriteLine("Program: Deklarace v1.0");  
    Console.WriteLine("Zdrojový kód programu demonstruje deklaraci nové  
metody a její volání.");  
    Console.WriteLine("-  
-  
-");  
};
```

## Volání metody

Volání metody spočívá v tom, že někde do kódu napíšeme název metody `Info()`, čímž při vyhodnocování kódu dojde k jejímu zavolání = spuštění.

V našem případě se tedy vrátíme do metody `Main()` a dopíšeme volání metody `Info()`.

```
public static void Main (string[] args)  
{  
    Info();  
    Console.WriteLine ("Hello World!");  
}
```

## Opakované volání

Jak jsme již zmiňovali, metodu lze volat opakovaně. U metody `Info()` to nemá zřejmě moc význam, ale ...

```
public static void Main (string[] args)  
{  
    Info();  
    Info();  
    Console.WriteLine ("Hello World!");  
    Info();  
    Info();  
}
```