

Programování

Funkce a jejich atributy, přetížení funkcí

Hlavní funkce programu - opakování

- ▶ Hlavní Main funkce
- ▶ Prvně spouštěná funkce při zapnutí aplikace
- ▶ Vlastní kód a volání dalších funkcí píšeme do složených závorek
- ▶ Vstupní argumenty hlavní funkce, které se vkládají jako parametry při spuštění - pole řetězců
- ▶ Podle argumentů hlavní funkce můžeme odlišit chování aplikace

```
static void Main(string[] args)
{
    ...
}
```

Metoda / funkce

- ▶ = blok kódu, který vykonává určitou úlohu

```
public void PrintInfo()  
{  
    Console.WriteLine("Příprava programu...");  
}
```

- ▶ Před samotným voláním libovolné funkce je nutná její deklarace

```
public void PrintInfo()  
{  
}
```

- ▶ Deklarovanou funkci již můžeme volat

```
PrintInfo();
```

Deklarace funkce - viditelnost

```
public void PrintInfo()  
{  
}
```

- ▶ Jako první uvádíme viditelnost funkce v našem programu
 - ▶ **Public** - k funkci lze přistupovat z jakéhokoliv místa
 - ▶ **Private** - k funkci lze přistupovat pouze ze stejné třídy nebo struktury
 - ▶ **Protected** - k funkci lze přistupovat pouze ve stejné třídě nebo ve třídě odvozené z původní třídy
 - ▶ **Internal**
 - ▶ **Protected internal**
 - ▶ **Private protected**
- ▶ Stejné parametry viditelnosti můžeme využít i u vlastností tříd

Deklarace funkce - návratový typ

```
public void PrintInfo()  
{  
}
```

- ▶ Jako druhý údaj uvádíme návratový datový typ
- ▶ Návratový typ může být jakýkoliv jednoduchý datový typ
 - ▶ Int, double, bool, string, ...
- ▶ Rovněž lze vracet i složené konstrukce
 - ▶ Pole, seznamy, objekty, ...
- ▶ Pokud nám má funkce něco vracet, musí se v ní objevit klíčové slovo **return**
- ▶ V případě, že funkce nic nevrací, uvádíme jako návratový typ void

```
public int Funkce()  
{  
    ...  
    return 42;  
}
```

Deklarace funkce - název funkce

```
public void PrintInfo()  
{  
}
```

- ▶ Třetí povinnou položkou u deklarace uvádíme název funkce
- ▶ Stejné podmínky jako pro deklaraci proměnné
 - ▶ Bez diakritiky
 - ▶ Nesmí začínat číslem
 - ▶ Nesmí obsahovat mezery
 - ▶ Víceslovné názvy oddělujeme podtržítkem nebo velkým písmenem na počátku slova
- ▶ Oproti proměnné funkce začínají velkým písmenem

```
Počet odkazů: 0  
public void Funkce_o_vice_slovech()  
{  
...  
}
```

```
Počet odkazů: 0  
public void FunkceOViceSlovech()  
{  
...  
}
```

Deklarace funkce - vstupní proměnné

- ▶ Vstupní proměnné funkce nám slouží jako vstupy do funkce
- ▶ Na základě vstupních proměnných můžeme ovlivnit celé chování funkce
- ▶ Vstupní proměnné uvádíme jako seznam oddělený čárkami
 - ▶ Datový typ + název proměnné, pod kterým ji budeme používat v těle funkce

```
public void Funkce(int a, double b, string c, bool d)
```

- ▶ Při volání funkce, která očekává vstupní proměnné musíme dodržet
 - ▶ Počet proměnných
 - ▶ Jejich datový typ
 - ▶ Správné pořadí

```
Funkce(1, 12.7, "text", false);
```

Přenesení hodnoty vstupních proměnných do hlavního těla programu

- ▶ Proměnné, které používáme uvnitř těla funkce nejsou dosažitelné mimo tělo funkce!
- ▶ Pokud uvnitř těla funkce změníme hodnotu vstupní proměnné a nevrátíme do hlavního vláknu programu tuto změnu proměnná se nám nezmění!

Počet odkazů: 1

```
private static void Increase1(int a)
{
    ++a;
}
```

Počet odkazů: 1

```
private static int Increase2(int a)
{
    return ++a;
}
```

```
int cislo = 10;
Increase1(cislo);
Console.WriteLine(cislo); // vypise 10
cislo = Increase2(cislo);
Console.WriteLine(cislo); // vypise 11
```


Přetížení funkcí

- ▶ U proměnných platí, že nelze opakovat proměnné se stejným jménem
- ▶ U funkcí lze opakovaně vytvářet funkce se stejným názvem, ale pouze pokud platí jedna z možností:
 - ▶ Různý počet vstupních proměnných
 - ▶ Stejný počet vstupních proměnných, ale vstupní proměnné mají jiný datový typ
- ▶ Přetížením můžeme získat větší variabilitu kódu

Počet odkazů: 0

```
public void Foo() { }
```

Počet odkazů: 0

```
public void Foo(int a) { }
```

Počet odkazů: 0

```
public void Foo(string s) { }
```

Vytváření vlastních funkcí - procvičení

1. Funkce, která přijímá jako vstupní parametry dvě čísla a vrací jejich součet
2. Funkce, která rozhodne, zda celé číslo, které obdrží je dělitelné 6
3. Funkce, která přijme jméno a věk hráče a vypíše na výstup hráčovu přezdívku, která bude ve tvaru Jméno_Věk
4. Funkce, která přijme na vstup pole celých čísel a celé číslo. Funkce nám vrátí pole čísel, kde každé číslo v poli je vynásobeno číslem, které funkce obdržela
5. Funkce, která načte celé číslo a řetězec. Funkce nám zadaný řetězec vypíše tolikrát, kolikrát jsme zadali. Pokud je vstupní číslo 0 nebo záporné, tak vypíšeme „neplatný vstup“
6. Funkce, která načte pole čísel a vrátí aritmetický průměr hodnot