

Základní programové konstrukce

Roman Princ
Patrik Mrnka

Obsah

Proměnná

Datový typ

Reference

Hodnota

Program

Podprogram

Podmínky

Cykly

Operátory

Proměnná

Místo v paměti, kam se ukládají data

Musí se v kódu deklarovat

Je složená z:

- Datový typ (int, string, float, ...)
- Název
- Hodnota (odpovídá datovému typu)

Datový typ

- Hodnotové (přímo proměnná)
 - čísla (int, float, double...)
 - Logické (bool)
- Referenční (odkazují na místo v paměti)
 - Class (object, string,...)
 - Interface
 - Array

Reference

- Odkaz na proměnnou nebo instanci objektu

```
int MultiplyByTen(int parameter)
{
    return parameter * 10;
}

int param = 10;

Console.WriteLine(MultiplyByTen(param));
```

Hodnota

- Entita programu, se kterou může program pracovat

```
int value = 0;
```

Program

- Proces v projektu, který řídí fungování aplikace
- V C# Program.cs

Podprogram

- Část programu
- Lze ji opakovaně využívat v programu
- Typicky funkce

Podmínky

- **Složený příkaz**
 - Příkaz se provádí za sebou
- **Příkaz úplný podmíněný (2 větve)**
 - If/else – pokud platí podmínka, proved' příkaz; jinak proved' příkaz 2

```
[-] if(value < 0)
    {
        value++;
    }
[-] else
    {
        value--;
    }
```

Podmínky

- **Příkaz neúplný podmíněný (1 větev)**
 - If – pokud platí podmínka, proved' příkaz
- **Více násobné větvení**
 - Pokud se hodnota rovná x, proved' příkaz 1; pokud se rovná y, proved' příkaz 2...

```
if(value < 0)
{
    value++;
}
```

```
switch (value)
{
    case 1:
        value++;
        break;
    case 0:
        value--;
        break;
    default:
        break;
}
```

Cykly

- **Podmínka na začátku**

- While – příkaz nemusí proběhnout, může běžet do nekonečna

```
int c = 0;
while (c < 10)
{
    Console.WriteLine(c);
    c++;
}
```

- **Podmínka na konci**

- Do-while: vykonej příkaz, opakuj zda platí podmínka

```
int c = 0;
do
{
    Console.WriteLine(c);
    c++;
} while (c > 0);
```

Boxing / Unboxing

Boxing

- Převod proměnné typu hodnota (int...) na typ reference (objekt)

```
int a = 23;  
Object Obj = a; // Boxing
```

Unboxing

- Převod proměnné typu reference (objekt) na typ hodnota (int...)

```
int a = 23;  
Object Obj = a; // Boxing  
int b = (int)Obj; // Unboxing
```

Konverze typů

- Převedení jednoho datového typu na jiný

Implicitní

- Přetypování se provede automaticky
 - Operand s nižší prioritou se konvertuje na operand s vyšší prioritou

```
int a = 3;
double b = 2.123;
double c;
c = a + b; // a se převede na double hodnotu 3.000
```

Explicitní (vynucené)

- Provede se tam, kde implicitní konverze není možná

```
double a = 5.3;
int b;
b = (int)a; // do proměnné b se uloží oříznuté a = tj. 5
```

Tabulka priorit

Int
Unsigned int
Long
Unsigned long
Float
Double
Long double

Struct

- Datový typ hodnotového typu
- Jedna proměnná obsahující související data různých typů dat

```
struct Knihy  
{  
    public string Název;  
    public string Autor;  
    public int Id;  
}
```

Cykly

- Známý počet průchodů
 - For – opakuj tolikrát, kolikrát je zadáno v podmínce

```
for (int i = 0; i < 20; i++)  
{  
    Console.WriteLine(i);  
}
```

Cykly

- Iterace prvků kolekce
 - Foreach – provádí se pro každý prvek přítomný v poli

```
int[] pole = { 'A', 'H', 'O', 'J' };  
  
foreach(int i in pole)  
{  
    Console.WriteLine(i);  
}
```


Operátory

Aritmetické

$+$, $-$, $*$, $/$

Logické

$\&\&$, $||$, $!$

Bitové

$\&$, $|$, \sim

Relační

$<$, $>$, $==$

Přístupové

$[]$, $..$, $()$

Ostatní

$=$, $=>$