KURZ PROGRAMOVÁNÍ

podklady pro 1. část

Použitý software

- MS Visual Studio Community 2017
- ke stažení zdarma:
  - https://www.visualstudio.com/vs/

- použitý programovací jazyk: C#
  - objektově orientovaný jazyk

Něco málo k objektům

- knihovna objektů
  - třída
    - instance

- objekt má…
  - atributy (vlastnosti)
  - metody (co umí udělat)
    - nevrací/nevrací hodnotu
    - má/nemá argumenty (parametry)

Struktura programu v jazyce C#

```
// příklad P_1_01
// Výstupní příkazy - metoda WriteLine

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace P_1_01
{
    class P_1_01
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Ahoj, programuju v C#");
            Console.ReadKey();
        }
    }
}
```

komentáře

použité knihovny

hlavní metoda
(spouští se při startu programu)

náš programový kód

Program v C#

- příkaz
  - je vždy ukončen středníkem
  - může být rozložen do více řádků
  - na jednom řádku může být více příkazů
- blok příkazů
  - příkazy, které se postupně vykonají
  - ve složených závorkách { }
- identifikátor
  - název proměnné, příkazu, metody…
  - case senzitiv = rozlišuje malá a velká písmena!
  - bez diakritiky, mezer

Komentáře v programu

- neprovádí se
- použití:
  - vysvětlení kódu (přehlednost programu)
  - zakomentování části programu, která se nemá vykonat (při ladění)

- 2 typy komentářů
  - // platnost od místa vložení do konce řádku

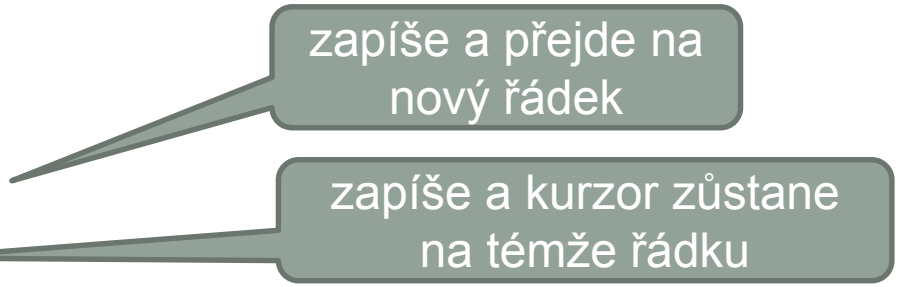
  - /* část řádku
    nebo i více řádků */

Metoda Main

- „hlavní program“, tj. místo, kam budeme zatím psát svůj kód
- při startu našeho programu se začnou vykonávat příkazy v metodě Main
- parametry v závorce za metodou
  - parametry příkazového řádku
  - zatím nevyužijeme

Konzolový výstup

- výstupní příkazy:
  - Console.WriteLine (*parametr*)
  - Console.Write (*parametr*)



  - *parametr* je textový řetězec (uzavřený do uvozovek)

Příklad použití Write a WriteLine

- oba dva kódy dělají totéž:

```
Console.WriteLine("Ahoj, programuju v C#");

Console.Write("Ahoj,");
Console.Write(" programuju");
Console.WriteLine(" v C#");
```

- výstup:

Proměnná

- místo v paměti (velikost podle datového typu)
- můžeme jí přiřazovat hodnotu
- můžeme tuto hodnotu použít
  - použitím identifikátoru proměnné v programu

- přiřazovací příkaz =

proměnná = výraz

  - nejprve se vyhodnotí výraz na pravé straně příkazu
  - potom se tato hodnota přiřadí (uloží) do proměnné na levé straně příkazu

Datové typy

- liší se
  - velikostí použité paměti
  - způsobem použití
- každé proměnné musíme deklarací přiřadit před prvním použitím datový typ

```
int cislo;
```

- deklarace může být spojena s přiřazením hodnoty

```
int cislo = 15;
string jmeno = Console.ReadLine();
```

Celočíselné datové typy

datový typ	velikost v paměti	rozsah hodnot	poznámka
int	4 B = 32 b (4 byty = 32 bitů)	-2 147 483 648 …2 147 483 647	nejčastější typ pro celá čísla
uint	4 B = 32 b	0 … 4 294 967 295	u = unsigned = bez znaménka
byte	1 B = 8 b	0 … 255	
sbyte	1 B = 8 b	-128 … 127	s = signed = se znaménkem
short	2 B = 16 b	-32 768 … 32 767	
ushort	2 B = 16 b	0 … 65 535	
long	8 B = 64 b	-9 223 372 036 854 775 808 … 9 223 372 036 854 775 807	
ulong	8 B = 64 b	0 … 18 446 744 073 709 551 615	

Datové typy – desetinná čísla

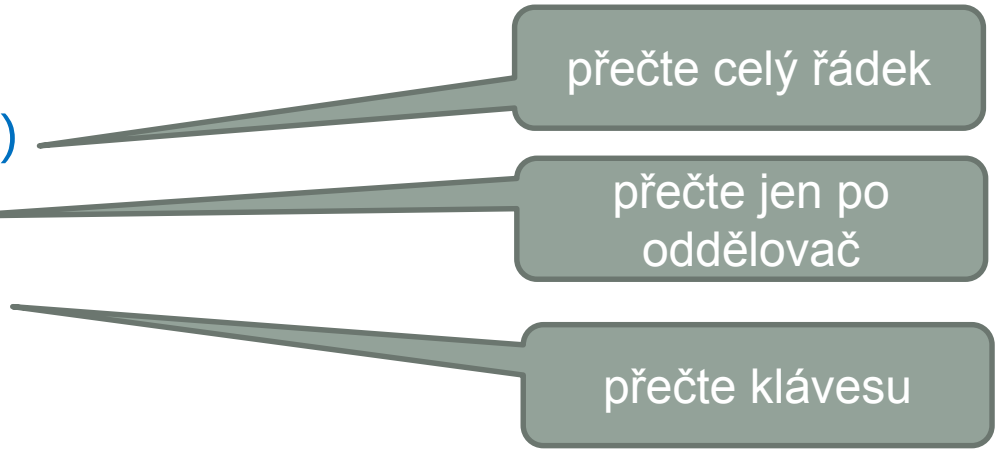
datový typ	velikost v paměti	rozsah hodnot	přesnost
float	4 B = 32 b	$+-1.5*10^{-45}$…$+-3.4*10^{38}$	7 číslic
double	8 B = 64 b	$+-5.0*10^{-324}$…$+-1,7*10^{308}$	15-16 číslic
decimal	16 B = 128 b	$-7.9*10^{28}$…$+7.9*10^{28}$ /10^{0} až 28	28-29 číslic

Další datové typy

datový typ	velikost v paměti	rozsah hodnot	poznámka
bool	1 B = 8 b	true / false	logické hodnoty
char	1 B = 8 b		1 znak
string			testový řetězec

Konzolový vstup

- vstupní příkazy
  - Console.ReadLine ()
  - Console.Read ()
  - Console.ReadKey()



  - hodnotu načtenou s pomocí ReadLine a Read většinou přiřazujeme do proměnné nebo je součástí výrazu
  - ReadKey často používáme jako poslední příkaz v konzolovém programu
    (čeká na stisk klávesy – tj. dokud ji nestiskneme, vidíme výstup našeho programu; po stisku klávesy program skončí a okno konzoly se zavře)

Parsování

- metody Read a ReadLine vrací textový řetěze, ten musíme převést na číslo:

```
int cislo = int.Parse(Console.ReadLine());
```

- opačný případ:
  - převod čísla na text se provádí automaticky

```
string slovo = cislo;
```

  - nebo lze použít konverzní metodu:

```
string slovo = cislo.ToString();
```

Operace s číselnými proměnnými

- přiřazení hodnoty
  - celočíselná proměnná: `int cislo = 15;`
  - proměnná typu float: `float cislo = 15.6F;`
  - proměnná typu double: `double cislo = 15.6D; // D nemusí být`
  - proměnná typu decimal: `decimal cislo = 15.6m;`

- desetinná čísla:
  - v zápisu programu používáme desetinnou tečku
  - v konzole používáme tečku nebo čárku podle nastavení OS

Aritmetické operace

- Přehled aritmetických operátorů:

operace	operátor	příklad
sčítání	+	c1 + c2
odčítání	-	c1 – c2
násobení	*	c1 * c2
dělení	/	c1 / c2
zbytek po celočíselném dělení	%	c1 % c2

Dělení

- operace / s celými čísly – výsledek je celá část podílu:

```
int c1 = 7;
int c2 = 3;
int podil = c1 / c2;            // vysledek je 2
int zbytek= c1 % c2;            // vysledek je 1
```

- protože 7 : 2 = 2 a zbytek 1

- „běžné“ dělení:
  - jeden z operandů musí být desetinný typ
  - přetypování
  - pokračování předchozího kódu:

```
double podil2 = (double)c1 / c2;
                // převod c1 na typ double
```

Priorita operátorů

- násobení/dělení má přednost před sčítáním/odečítáním
- pořadí lze ovlivnit závorkami – pouze kulaté ( )

- příklady:

```
int vysledek = 10 + 5 * 3;              // 25
vysledek = 10 + (5 * 3);        // 25, závorky zbytečné
vysledek = (10 + 5) * 3;                // 45
vysledek = ((10 + (5 * 3)) / 5          // 5
```

Proměnná na obou stranách výrazu

- přiřazovací příkaz NENÍ ROVNICE
- nejprve se vyhodnotí pravá strana přiřazovacího příkazu
  - na pravé straně může být hodnota nebo výraz
- tato výsledná hodnota se dosadí na levou stranu přiřazovacího příkazu
  - na levé straně musí být proměnná
- příklad

```
// ztrojnásobí hodnotu proměnné pocet:
int pocet = 10;
počet = pocet * 3;       // nová hodnota je 30
```

Další aritmetické operátory

operátor	význam	příklad	totéž jako…
+=	přičtení k proměnné	c += 5	c = c + 5
-=	odečtení od proměnné	c -= 5	c = c - 5
*=	znásobení proměnné	c *= 5	c = c * 5
/=	vydělení proměnné	c /= 5	c = c / 5

Operátory inkrementace a dekrementace

operátor	význam	příklad	totéž jako…

preinkrementace
++promenna
	zvětšení proměnné o 1 před vyhodnocením výrazu	`c = 5;`
		`v = 10 - ++c;`
			`c = 5;`
			`c = c + 1;`
			`v = 10 - c;`

postinkrementace
promenna++
	zvětšení proměnné o 1 po vyhodnocení výrazu	`c = 5;`
		`v = 10 - c++;`
			`c = 5;`
			`v = 10 - c;`
			`c = c + 1;`

predekrementace
--promenna
	zmenšeníproměnné o 1 před vyhodnocením výrazu	`c = 5;`
		`v = 10 - --c;`
			`c = 5;`
			`c = c - 1;`
			`v = 10 - c;`

postdekrementace
promenna--
	zvětšení proměnné o 1 po vyhodnocení výrazu	`c = 5;`
		`v = 10 - c--;`
			`c = 5;`
			`v = 10 - c;`
			`c = c - 1;`

Metody pro práci s čísly

metoda	**význam**	**příklad** (předpokládáme double c, d)

Round	zaokrouhlení	`c = 5.6789;`
		`d = Math.Round(c);  // 6 (0 des. míst)`
		`d = Math.Round(c, 2);  // 5.68 (2 des. místa)`

Pow	mocnina	`d = Math.Pow(c, 2)  // c na druhou` `d = Math.Pow(c, -1/3)  // 3. odmocnina z c`
Sqrt	2. odmocnina	`d = Math.Sqrt(c)  // odmocnina z c`

- další metody si najděte **ve třídě Math**

Operace s textovými řetězci

- Spojování textových řetězců
  - operátor +
  - příklad:

```
string celeJmeno = "Petr " + " Novák";
string jmeno, prijmeni;
celeJmeno = jmeno + prijmeni;
```

  - číslo se zde automaticky převede na text:
  - příklad:

```
string adresa = "Krátká " + 25;
```

- formátovací řetězec
  - zabuduje do textového řetězce hodnoty proměnných
  - příklady:

```
string s = string.Format("Součet {0} a {1} je {2}", a, b, a + b);
Console.WriteLine(s);
```

stejné jako:

```
Console.WriteLine("Součet " + a + " a " + b + " je " + (a + b));
```

Operace s textovými řetězci

- Metody a atributy použité v programu P_1_05:

metoda	význam
Trim	ořeže mezery na začátku a na konci textu
ToUpper	převede text na velká písmena
ToLower	převede text na malá písmena
Length	délka textového řetězce (počet znaků) není to metoda, ale atribut (nemá závorky)
	(najděte si význam dalších…)

Operace s textovými řetězci

- přístup k jednotlivým znakům řetězce
  - hranaté závorky [ ]
  - znaky počítáme od 0
  - příklad – první znak řetězce, třetí znak řetězce:

```
string slovo = "počítač";
char pismeno = slovo[0];     // p
pismeno = slovo[2];          // č
```

Relační operátory

operátor	význam
>	větší
<	menší
>=	větší nebo rovno
<=	menší nebo rovno
==	rovno
!=	nerovno

Logické operátory

operátor	význam
&&	logický součin (AND)
||	logický součet (OR)
!	negace (NOT)