



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

- Vstup

Vstup

Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_23_PRG_449		
Název tématické oblasti (sady)	Programování		
Název materiálu	Vstup		
Anotace	Text představuje možnost zadávání dat do běžící aplikace pomocí klávesnice. Zavádí pojem návratová hodnota. Rozebírá problematiku při ukládání textových dat a čistě číselných dat z klávesnice do proměnných.		
Autor	Ivan Pomykacz	Jazyk	čeština
Očekávaný výstup	Načte stisknuté klávesy a uloží do proměnné. Jde-li o číslo, provede parsování hodnoty, tak aby mohl být použitý vhodný datový typ.		
Klíčová slova	vstup z klávesnice, parsování řetězce, návratová hodnota		
Druh výukového zdroje	Výklad	Věková skupina žáků	15+
Typ interakce	aktivita	Ročník	1.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	23.09.2013	Celková velikost	

Obsah

- [Shrnutí](#)
- [Console](#)
 - [Návratová hodnota](#)
 - [Vstup a číslo](#)

Shrnutí

Pod vstupem u programu si můžeme představit cokoli, co se dostalo do našeho programu námi definovaným způsobem. Např. uživatel mohl napsat něco na klávesnici. Stisknuté klávesy se pak předaly do našeho programu a ten na jejich základě provedl nějakou akci. (Klávesnice je typickým vstupním zařízením)

Dalším vstupním zařízením by mohla být myš, nebo čtečka čárových (nebo QR) kódů, nebo i scanner. Jiným vstupem by mohl být soubor, ze kterého bude program načítat data. Dalším pak síťová komunikace, kde program přijímá data po síti (ať už z Internetu nebo lokální sítě).

Console

Zaměříme se na první popisovaný vstup, tj. vstup z klávesnice. V C# je dostupný objekt `Console`, který obsahuje metodu `ReadLine()`.

Metoda `ReadLine()` pozastaví běh programu a čeká, dokud uživatel stiskne klávesu `Enter`.

```
string jmeno;  
Console.Write("Napiš své jméno: ");  
  
jmeno = Console.ReadLine();  
  
Console.WriteLine("Ahoj " + jmeno + "!");  
Console.WriteLine("Bohužel neumím skloňovat ...");
```

Všimněte si, jak je v ukázce metoda `Console.Write()` a `Console.WriteLine()`. Při použití `Write()` nedojde k odřádkování a uživatel pak zadává své jméno ihned za dvojtečku (v řetězci metody `Write()`).

Návratová hodnota

Co je kritické k pochopení u tohoto příkladu je to, že metoda `ReadLine()`, vůbec jako první námi použitá metoda, vrací nějakou hodnotu. V tomto případě jde o text, který napsal uživatel na klávesnici.

Aniž bychom cokoli vysvětlili, automaticky je v příkladu použita proměnná typu `string`. Automaticky proto, že cokoli co napíšeme na klávesnici je vlastně text, a to i v případě, že napíšeme nějaké číslo. Prostě vše, co napíšete na klávesnici je text, a tedy typ `string`.

Metody mohou, ale nemusí mít návratovou hodnotu. Např. metoda `WriteLine()` nemá žádnou návratovou hodnotu (přesněji řečeno má, nazývá se `void`).

To, jestli a jakou má metoda návratovou hodnotu se lze dočíst v referenční příručce jazyka (hledejte klíčové slovo *return*).

Vstup a číslo

Z toho, co bylo právě popsáno **není možné použít následující** kód:

```
double cislo;  
Console.Write("Zadej číslo: ");  
cislo = Console.ReadLine();
```

Kompilátor okamžitě zahlásí chybu, že není možné přetypovat proměnnou `string` na typ `int`.

Jenomže mi potřebujeme pracovat s čísly, a chceme čísla zadávat z klávesnice.

Proto existují metody, které dokáží analyzovat textový řetězec, a pokud v něm rozpoznají číslo, tak provedou jakousi konverzi na požadovaný datový typ (`int` nebo `double`). Vizte následující ukázkou:

```
int cislo;  
string text;
```

```
Console.Write("Zadej celé číslo: ");  
text = Console.ReadLine();  
  
cislo = int.Parse(text);
```

Nejprve jsme do proměnné `text` uložili řetězec, který pravděpodobně bude obsahovat číslo. Následně jsme použili metodu `int.Parse()`, která má vstupní argument typu `string` a vrací typ `int`.

Celý příklad lze ještě zjednodušit o to, že nebudeme zbytečně deklarovat proměnnou typu `string`.

```
int cislo;  
  
Console.Write("Zadej celé číslo: ");  
cislo = int.Parse(Console.ReadLine());
```

Obdobně bychom mohli parsovat čísla na typ `double`. V příkladu výše stačí nahradit `int` za `double`.

Pozor! Metoda `Parse()` může též tzv. "vyhodit (throw)" výjimku (*exception*). To se může stát např. tehdy, zadáte-li takový řetězec, který neobsahuje číslo (nebo číslo jinak zkomolené). Prozatím nechme tento stav neošetřený, a vrátíme se k němu později.