



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

- Co když

Co když

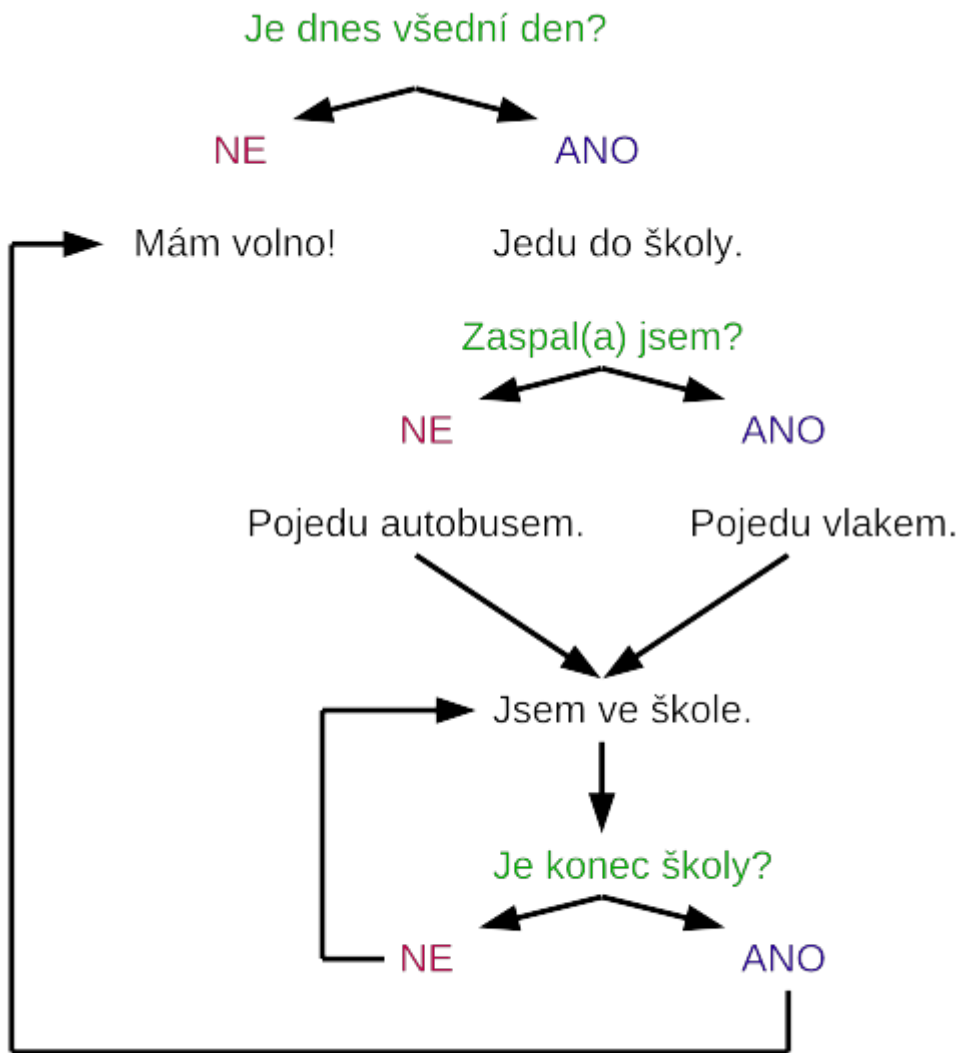
Název školy	Vyšší odborná škola obalové techniky a Střední škola, Štětí, příspěvková organizace		
Adresa školky	Kostelní 134, 411 08 Štětí		
IČ	46773509		
Název operačního programu	OP Vzdělávání pro konkurenceschopnost		
Registrační číslo	CZ.1.07/1.5.00/34.1006		
Označení vzdělávacího materiálu	VY_32_INOVACE_23_PRG_452		
Název tématické oblasti (sady)	Programování		
Název materiálu	Co když		
Anotace	Text uvádí řídicí konstrukci if-else. Na pseudo vývojovém diagramu demonstruje větvení kódu. Zavádí operátory porovnání, které bývají součástí výrazů v podmínkách. Předkládá cvičení na pravdivostní hodnoty true a false. Nakonec předkládá příklady zápisu řídicí konstrukce v základní podobě a následně rozšířené o bloky else-if-else.		
Autor	Ivan Pomykacz	Jazyk	čeština
Očekávaný výstup	Chápe význam hodnot true a false při vyhodnocování výrazů. Sestavuje jednoduché podmínky pro ovlivnění způsobu vykonávání kódu. Ovlivňuje, který kód se dle vyhodnocení výrazu vykoná a který ne.		
Klíčová slova	podmínka, výraz, větvení kódu		
Druh výukového zdroje	Výklad	Věková skupina žáků	15+
Typ interakce	aktivita	Ročník	1.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	04.10.2013	Celková velikost	

Obsah

- [Shrnutí](#)
- [Operátory porovnání](#)
- [Pravda nepravda](#)
- [Příklad](#)
- [Konstrukce if](#)
- [Konstrukce if-else](#)
 - [Muž a žena](#)

Shrnutí

Řídicí konstrukce umožňují usměrňovat tok kódu, tj. vykonávání některých příkazů a jiné třeba přeskočit (vynechat). Jsou to právě tyto konstrukce, díky kterým můžeme tvořit různé algoritmy a výsledné programy pak skutečně něco "dělají".



Operátory porovnání

Používají se v podmínkách při testování nějakého stavu, např. je-li něco rovno něčemu jinému a nebo je-li něčeho více, atp.

Patrně je znáte už z matematiky, ale přeci jen pro úplnost toho, jak vypadají v jazyce C#.

- `==` .. rovná se
- `!=` .. je různý od (nerovná se)
- `>` .. je větší
- `>=` .. je větší nebo rovno
- `<` .. je menší
- `<=` .. je menší nebo rovno

Pravda nepravda

- Pokuste se přiřadit hodnotu *True* nebo *False* na základě následujících výrazů: `bool bool1;`
`bool bool2;`
`bool bool3;`
`bool bool4;`

```
bool bool5;

// 17 < 18/100
bool1 =

// 100 == 33 * 3 + 1
bool2 =

// 19 <= Math.Pow(2,4)
bool3 =

// -22 >= -18
bool4 =

// 99 != 98 + 1
bool5 =

Console.WriteLine(bool1);
Console.WriteLine(bool2);
Console.WriteLine(bool3);
Console.WriteLine(bool4);
Console.WriteLine(bool5);
```

- Pokuste se zjistit hodnotu *True* nebo *False* na základě poněkud složitějších výrazů: **bool** bool1;

```
bool bool2;
bool bool3;
bool bool4;
bool bool5;

// 20 + -10 * 2 > 10 / 3 / 2
bool1 =

// Math.Pow(10 + 17, 2) == Math.Pow(3,6)
bool2 =

// Math.Pow(1,Math.Pow(2,3)) <= -(-(-1))
bool3 =

// 40 / 20 * 4 >= Math.Pow(-4,2)
bool4 =

// Math.Pow(100,0.5) != 6 + 4
bool5 =

Console.WriteLine(bool1);
Console.WriteLine(bool2);
Console.WriteLine(bool3);
Console.WriteLine(bool4);
Console.WriteLine(bool5);
```


Příklad

Cvičení výše bylo na pochopení stavů *Pravda*, *Npravda*, a toho, že není nic mezi tím. Pokračujme praktičtějším příkladem.

Od uživatele si necháme z klávesnice zadat číslo. A následně rozhodneme, zda-li toto číslo je kladné nebo záporné, případně rovno 0.

```
int x;

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

if (x > 0) {
    Console.WriteLine("Číslo je kladné");
}

if (x < 0) {
    Console.WriteLine("Číslo je záporné");
}

if (x == 0) {
    Console.WriteLine("Číslo je nula");
}
```

Konstrukce if

Konstrukt *if* začíná právě slovem `if ()` společně s kulatými závorkami, do kterých se zapisuje výraz, jenž se bude vyhodnocovat.

```
if (vyraz) {
    // příkazy, které se mají provést, je-li výsledek výrazu true (pravda)
}
```

Ihned za kulatými závorkami následují složené závorky `{ a }`, které ohraničují příkazy, které se vykonají pouze tehdy, je-li výraz v kulatých závorkách vyhodnocen jak `true` (a je to tedy typ *bool*).

Výrazem tedy může být téměř cokoli, co vrátí typ *bool* (tedy `true`, `false`):

- proměnná typu *bool*;
- metoda, která vrací typ *bool*;
- výraz, jehož součástí je alespoň jeden z operátorů porovnání (viz výše);
- jeden z booleanovských operátorů (`and` - `&&`, `or` - `||`);

Konstrukce if-else

Konstrukt lze ještě rozšířit o blok `else`. Stojí v podstatě v opozici proti tvrzení `if`.

```
if (vyraz) {
    Console.WriteLine("Výsledek výrazu je true");
}
else {
    Console.WriteLine("Výsledek výrazu je false");
}
```

Blok `else` může být jen jeden a vždy na konci. Proč? No, buď je výraz pravdivý, a pak se vykonají příkazy uvnitř bloku `if`, a nebo pravdivý není, nic mezi tím není.

Muž a žena

Mějme program, který vyhodnocuje zadané údaje, konkrétně, zda-li uživatel vyplnil správně informaci o tom, zda-li je "muž" nebo "žena".

Pokud souhlasíte, tak existují pouze tyto dva stavy: *je-li uživatel muž, nemůže být žena* a obráceně, *je-li uživatel žena, nemůže být muž*.

Potom, mějme proměnnou `volba`, do které budeme ukládat uživatelskou odpověď, tedy to, co napíše na klávesnici. A podle výše domluveného, platí-li výraz `volba == "muž"`, pak jde o muže. Jelikož další jediná možná odpověď je "žena", můžeme kód zapsat např. takto:

```
string volba;
```

```
Console.WriteLine("Jsi muž nebo žena? Odpověz \"muž\" nebo \"žena\": ");
volba = Console.ReadLine();
```

```
if (volba == "muž") {
    Console.WriteLine("Eviduji muže.");
}
else {
    Console.WriteLine("Eviduji ženu.");
}
```

Všimněte si, že testujeme pouze to, zda-li proměnná obsahuje řetězec "muž". Pokud výraz není pravda, tzn. proměnná obsahuje jiný řetězec, uživatel bude automaticky vyhodnocen jako "žena". Vyzkoušíte-li výše uvedený kód, tak zjistíte, že pokud například napíšete do odpovědi třeba jen "Muž" s velkým počátečním písmenem, dojde ke konstatování "Eviduji ženu".

Jak z toho ven? Mohli bychom blok `else` vynechat a napsat další `if` pro detekci řetězce `volba == "žena"`.

```
if (volba == "muž") {
    Console.WriteLine("Eviduji muže.");
}

if (volba == "žena") {
```

```
    Console.WriteLine("Eviduji ženu.");  
}
```

A tím jsme ošetřili chybné vyhodnocení muže a ženy.

Ale co se teď stane se všemi chybnými odpověďmi? Tedy s takovými odpověďmi, které obsahují překlepy nebo schválně nesmyslná slova?

Program by měl být schopen tyto vstupy rozpoznat a adekvátně na ně zareagovat, aby se uživatel dozvěděl, zda-li to, co zadal, je OK.

V takových případech se hodí provést tzv. zřetězení bloků `if` do jednoho mega bloku.

```
string volba;
```

```
Console.Write("Jsi muž nebo žena? Odpověz \"muž\" nebo \"žena\": ");  
volba = Console.ReadLine();
```

```
if (volba == "muž") {  
    Console.WriteLine("Eviduji muže.");  
}  
else if (volba == "žena") {  
    Console.WriteLine("Eviduji ženu.");  
}  
else {  
    Console.WriteLine("Chyba zadání.");  
}
```

Všimněte si, že uvedená konstrukce obsahuje dvakrát slovo `else`, ale poprvé na `else` navazuje `if`.

Jak se to pak celé chová?

- **Jestliže [if]** `volba == "muž"`, vypiš "Eviduji muže",
- **jinak jestliže [else if]** `volba == "žena"`, vypiš "Eviduji ženu",
- **jinak [else]** vypiš "Chyba zadání".