



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 - cvičení

Co když - cvičení

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_453		
Název tématické oblasti (sady)	Programování		
Název materiálu	Co když - cvičení		
Anotace	Soubor úloh pro procvičování práce s řídicími konstrukcemi.		
Autor	Ivan Pomykacz	Jazyk	čeština
Očekávaný výstup	Tvoří výrazy. Tvoří jednoduché, jednoúčelové aplikace, které dokáží podmíněně vykonávat konkrétní bloky kódu dle vyhodnoceného výrazu. Navrhuje a realizuje vlastní větvení kódu.		
Klíčová slova	podmínka, výraz, větvení kódu		
Druh výukového zdroje	Pracovní list	Věková skupina žáků	15+
Typ interakce	aktivita	Ročník	1.
Speciální vzdělávací potřeby	žádné		
Zhotoveno, (datum/období)	07.10.2013	Celková velikost	

Cvičení VI - Kalkulačka v3.0

Obsah

- [Příklad](#)

Příklad

```
using System;
```

```
namespace Kalkulacka3
```

```
{
```

```
    class MainClass
```

```
    {
```

```
        public static void Main (string[] args)
```

```
        {
```

```
            Console.WriteLine("Program: Kalkulačka v3.0\n");
```

```
        /**
```

```
* Vytvořte program podobný Kalkulačce v2.0, ale
* na místo konstruktů switch použijte if-else
*
* V programu bude navíc ošetřeno dělení nulou.
*
*/

double x;
double y;
double vysledek;
string op;

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

Console.Write("Zadej číslo: ");
y = double.Parse(Console.ReadLine());

Console.Write("Zadej operaci (*,/,+,-): ");
op = Console.ReadLine();

if (op == "+") {
    vysledek = x + y;
    Console.WriteLine(x + " " + op + " " + y + " = " + vysledek);
    //Console.WriteLine("{0} {1} {2} = {3}", x, op, y, vysledek);
}
else if (op == "-") {
    vysledek = x - y;
    Console.WriteLine(x + " " + op + " " + y + " = " + vysledek);
}
else if (op == "*") {
    vysledek = x * y;
    Console.WriteLine(x + " " + op + " " + y + " = " + vysledek);
}
else if (op == "/") {
    if (y == 0) {
        Console.WriteLine("Zadané číslo nemůže být rovno 0.");
    }
    else {
        vysledek = x / y;
        Console.WriteLine(x + " " + op + " " + y + " = " +
vysledek);
    }
}
else {
    Console.WriteLine("Neznámý operátor!");
}

Console.Write("Press any key to continue . . . ");
Console.ReadKey(true);
}
```

```
}  
}
```

Cvičení VII - Diagram 1

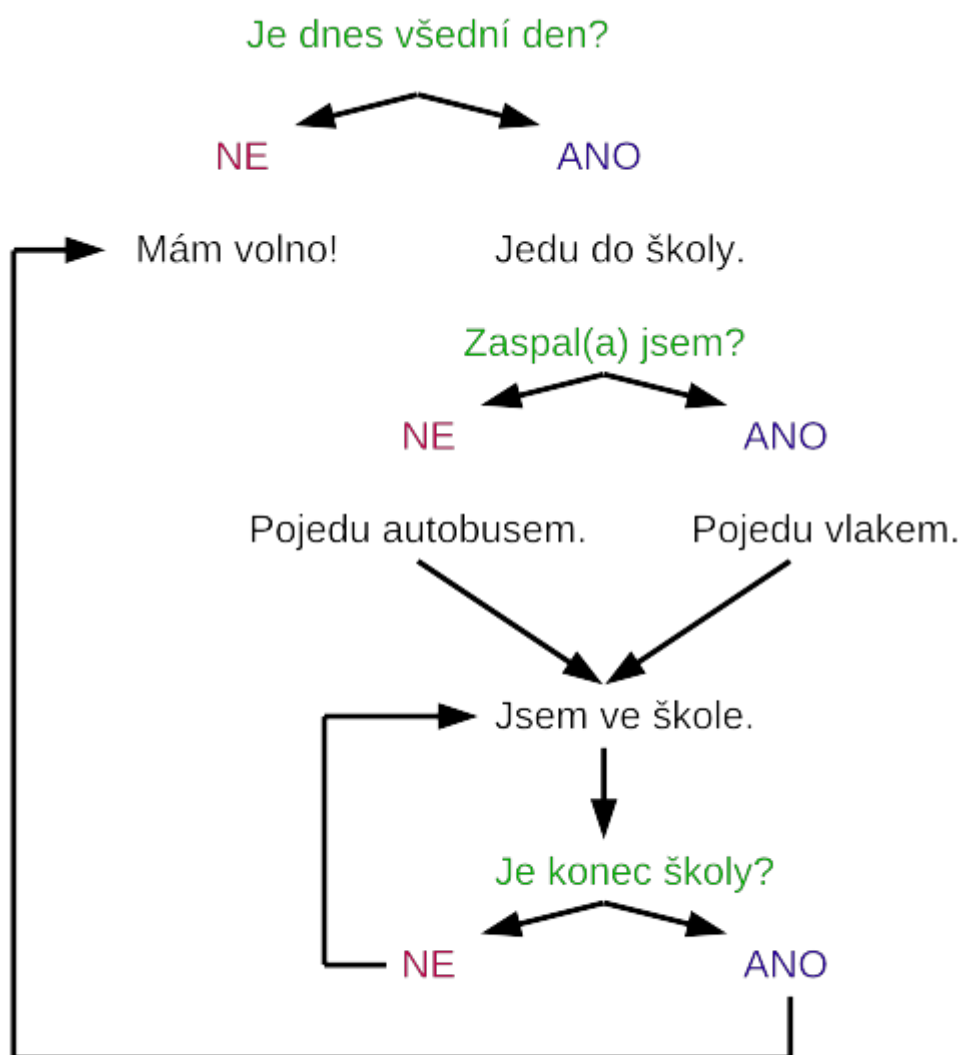
Obsah

- Příklad
 - Řešení

Příklad

Provedte implementaci diagramu na obrázku níže.

Použijte dosavadní znalosti konstrukce `if-else`.



Řešení

```
Console.WriteLine ("Program: Do školy v1.0");

Console.WriteLine("Na otázky odpovídej \"ano\" nebo \"ne\".");

string volba;

Console.Write("Je dnes všední den? ");
volba = Console.ReadLine();
if (volba == "ano") {
    Console.WriteLine("Jedu do školy.");
    Console.Write("Zaspal(a) jsem? ");
    volba = Console.ReadLine();
    if (volba == "ano") {
        Console.WriteLine ("Pojedu vlakem.");
    }
    else if (volba == "ne") {
        Console.WriteLine ("Pojedu autobusem.");
    }
    else {
        Console.WriteLine ("Neumím rozhodnout.");
    }

    Console.WriteLine ("Jsem ve škole.");
    Console.Write("Je konec školy? ");
    volba = Console.ReadLine();
    if (volba == "ano") {
        Console.WriteLine("Mám volno!");
    }
    else if (volba == "ne") {
        Console.WriteLine("Jsem stále vše škole!");
    }
    else {
        Console.WriteLine ("Nevím, co se děje.");
    }
}
else if (volba == "ne") {
    Console.WriteLine("Mám volno!");
}
else {
    Console.WriteLine("Neplatná volba!");
}
```

Cvičení VIII - Diagram 2

Obsah

- Pomocné metody
 - Using
- Příklad
 - Řešení

Pomocné metody

Následující kód obsahuje tři pomocné metody.

Celý kód zkopírujte a vložte do svého projektu takovým způsobem, aby nově vložený kód byl na stejné úrovni, jako je metoda `Main`.

```
public static bool Otazka() {
    string klav;
    bool odpoved;

    while (true) {
        klav = Odpoved();
        if (klav == "ano") {
            odpoved = true;
            break;
        }
        else if (klav == "ne") {
            odpoved = false;
            break;
        }
        else {
            NapisPomalu("Nerozumím, zadej buď \"ano\" nebo \"ne\".");
        }
    }

    return odpoved;
}

public static void Pauza(int doba = 200) {
    Thread.Sleep(doba);
}

public static string Odpoved() {
    Console.Write(": ");
    return Console.ReadLine();
}

public static void NapisPomalu(string text) {
    int rychlost = 100;

    foreach (char znak in text.ToCharArray()) {
        if (rychlost > 50) {
            Thread.Sleep(rychlost);
        }
        Console.Write(znak);
    }
}
```

```

    }
    Console.WriteLine("\n");
}

```

Pokud použijete sbalovací tlačítka vlevo od kódu `-`, na všechny metody, měli byste vidět podobný přehled, jako na obrázku. (Na obrázku už je kód i zarovnaný)

```

1  using System;
2
3  namespace diagram2
4  {
5  class MainClass
6  {
7  public static void Main (string[] args)...
11
12 public static bool Otazka()...
33
34 public static string Odpoved()...
38
39 public static void NapisPomalu(string text)...
50 }
51 }

```

Všimněte si hlavně mezi jakými závorkami je kód vložený. Až doposud jsme totiž vkládali kód jen do metody `Main`.

Using

Nakonec ještě dopište do horní části kódu, hned za `using System;` nový řádek: `using System.Threading;`

```

using System;
using System.Threading;

```

Otestujte klávesou F8, zda je vše OK.

Příklad

Vytvořte si vlastní diagram s otázkami a odpověďmi. Následně proveďte jeho implementaci s využitím metod `NapisPomalu` a `Otazka`.

Řešení

```
NapisPomalu("Program: Pomalu v1.0");
```

```
bool odp;
```

```

NapisPomalu("Píše se ti rychle?");
odp = Otazka();
if (odp == true) {
    NapisPomalu("Tak OK, není co řešit");
}

```

```
}  
else {  
    NapisPomalu("A chceš zrychlit?");  
    odp = Otazka();  
    if (odp == true) {  
        NapisPomalu("Co psaní všemi deseti?");  
        odp = Otazka();  
        if (odp == true) {  
            NapisPomalu("Na netu je celkem dost kurzů i zdarma.");  
        }  
        else {  
            NapisPomalu("Je to moc těžký?");  
            odp = Otazka();  
            if (odp == true) {  
                NapisPomalu("Jo, souhlasím.");  
            }  
            else {  
                NapisPomalu("No, pokud si o nějaký prst přišel, tak mě to  
mrzí.");  
            }  
        }  
    }  
    else {  
        NapisPomalu("Hmm, tak nic.");  
    }  
}
```

Test III

Obsah

- Test
 - Ukázka testu
 - Vyhodnocení testu
 - Náplň testu
 - Bonus
- Odevzdání

Test

Vytvořte program "Test", který uživateli předloží několik (min 4) testových otázek. Jde o formu klasického (někdy i zatracovaného) testu, kde testovaný subjekt má na výběr ze tří (nebo více) variant.

Pro test platí následující pravidla:

- U každé otázky je správná pouze jedna odpověď.
- U každé otázky existuje správná odpověď.

- Za správnou odpověď je možné získat právě jeden bod.
- Není možné se vracet k již vyplněným odpovědím.
- Není možné otázky přeskokovat.
- Test není možné přerušit.

Ukázka testu

Stáhněte si a spusťte [ukázku testu \(zip verze\)](#), abyste věděli, jak by měl program vypadat.

Vyhodnocení testu

Na konci testu se zobrazí celkový počet získaných bodů. Následně se vypíše hodnotící text (ten si můžete vymyslet).

Hodnotící text se váže k počtu bodů:

1. Nikdo není dokonalý ... Ale tohle je kacířství!
2. Mezi nebem a zemí jsou věci, o kterých zatím nevíš.
3. Neutrální ... Nezvažuješ snad temnou stranu síly - nevědomost?
4. Správně, není dobré vynášet před sebe hned všechny karty.
5. Evidentně jsi IT GURU (bow).

Náplň testu

Můžete si vymyslet svůj vlastní test (při dodržení zadání). Nebo můžete použít následující texty:

Co je to CSS?

- a) Computer Science Senatorium
- b) Common Sense Stubbornness
- c) Cascading Style Sheets

Co je to Fedora?

- a) Nový engine pro Crysis V (pouze pro PC)
- b) Operační systém
- c) Zásilková služba (v USA)

Co je to Gnu?

- a) Goniometrická funkce (Gonus)
- b) Hnutí za globální svobodu
- c) Pakůň

$001+010+011+100+101+110+111 = ?$

- a) 28
- b) 32
- c) 34

Bonus

Můžete použít metody `NapisPomalu()` a `Pauza()`, které jsou uvedeny ve *Cvičení VIII - Diagram 2*.

Odevzdání

Odevzdejte pouze soubor se zdrojovým kódem.

Mini příklady I

Obsah

- Cvičení
 - Úloha 1
 - Úloha 2
 - Úloha 3
 - Úloha 4
 - Úloha 5
 - Úloha 6
- Odevzdání

Cvičení

Úloha 1

Program vyzve uživatele k zadání dvou čísel. Následně program čísla porovná a vyhodnotí, zda-li první číslo je větší, menší nebo rovno druhému z čísel.

Úloha 2

Program vyzve uživatele k zadání čísla od 1 do 5. Zadá-li uživatel číslo mimo zadaný rozsah, program vypíše chybovou hlášku.

Úloha 3

Vytvořte program na výpočet objemu kváдру ($V = a*b*c$). Zajistěte aby program upozornil uživatele na chybně zadanou hodnotu. Chybně zadaná hodnota je:

- pokud je číslo nula;
- je-li číslo záporné.

Nastane nějaká z chyb, program neprovede výpočet.

Úloha 4

Vytvořte program, který bude počítat přibližný čas potřebný k cestování z místa A do místa B.

Program nejprve vyzve uživatele k zadání vzdálenosti mezi místem A a B (km). Následně se dotáže (nechá uživatele vybrat z možností) na způsob dopravy. Možnosti dopravy a jejich průměrné rychlosti budou:

- pěšky (4 km/h);
- autem (70 km/h);
- letadlem (600 km/h);
- teleportem (1 079 252 848 km/h).

Program poté spočítá potřebný čas (v hodinách) k dosažení cíle a vypíše tuto informaci na obrazovku. Pokud by bylo číslo příliš malé, převede se čas na vteřiny.

V programu ošetřete stavy, kde by uživatel mohl zadat nesmyslnou hodnotu (např. záporná vzdálenost). Pokud takový stav nastane program uživatele varuje a skončí nebo jej nechá vložit údaj znovu, záleží na vás. (Vzorec na výpočet potřebného času je $t = s / v$)

Úloha 5

Obyčejný film je promítán o snímkové frekvenci 25 snímků za vteřinu. Pokud film trvá jednu minutu, obsahuje 60*25 snímků ($1\text{min} = 60\text{s} \Rightarrow 60*25 = 1500$).

Velikost jednoho snímku u full HD rozlišení je 5.9MB (pokud snímek není komprimován).

Nechť program vyzve uživatele k zadání *názvu filmu* a následně jeho *délky v minutách* (např. 2 hodinový film bude mít přibližně 120min).

Program pak spočítá velikost takového filmu (vynásobením velikosti snímku a délkou filmu ve vteřinách a ještě krát počet snímků). Program by měl hlídat takové chybové stavy, kde délka by neměla být záporná.

Následně program nabídne možnost uložení filmu na:

- DVD (8.55 GB);
- Blu-ray (50 GB);
- X-ray (1000 GB).

Bude potřeba převést velikost filmu (která je nyní v MB, na GB, tj. vydělit 1024). Poté vydělením velikost filmu a velikostí média zjistit odpovídající počet médií.

Úloha 6

Program vyzve uživatele k zadání textového řetězce (např jména). Následně program zjistí délku textu (vizte ukázkou níže) a rozhodne, zda-li nebyla překročena maximální povolená délka. Maximální povolená délka řetězce je 10 znaků.

```
string s = "Lorem ipsum";  
int delka;  
// do prom. delka se uloží délka řetězce s  
delka = s.Length;
```

Odevzdání

Odevzdejte pouze soubory `.cs`, které vhodně pojmenujete.

Mini příklady II

Obsah

- [Úloha 1 - Barevná hloubka](#)
- [Úloha 2 - Pixmapa](#)

Úloha 1 - Barevná hloubka

Program spočítá na základě počtu barev, kolika bitová barevná hloubka tyto barvy pojme. Např. uživatel zadá, že chce barevnou hloubku pro 16 barev. Program výpočtem zjistí, že tomu odpovídá 4bitová barevná hloubka.

Pro výpočet můžete použít funkci `Math.Log()`, která přebírá dva parametry, číslo a základ logaritmu. Např. u příkladu se 16 barvami bude funkce vypadat `Math.Log(16, 2)`. Číslo 2 je základem proto, že bit může nabývat dvou hodnot.

Program pak bude navíc výsledek zaokrouhlovat, aby nevycházela čísla jako 3.9bit. K tomu můžete použít funkci `Math.Ceiling()`, která přebírá jeden vstupní argument (číslo) a vrací upravenou hodnotu..

Program ohlídá takové stavy, jako třeba záporné hodnoty.

Testovací hodnoty:

- -5
- 15
- 16
- 65000
- 15000000

[demo](#)

Úloha 2 - Pixmapa

Program spočítá velikost pixelové mapy (bitmapy) na základě vstupních údajů, které mu předloží uživatel. Program bude potřebovat informaci o tom,

- jaký je rozměr mapy (šířku a výšku),
- jakou má použít barevnou hloubku (kolik bitů).

Výsledek se zobrazí v jednotkách Byte. Pokud ovšem bude číslo příliš veliké, přepočítá se na kB, případně MB. Příliš veliké znamená, že bude 1024 krát větší než je nejbližší násobek.

Např. pixmapa o rozměrech 1024x768 při hloubce 1bit bude zabírat: 98 304 Byte (číslo bylo děleno 8 - bit na Byte).

Tzn., že výsledek je větší jak 1024 Byte, ale menší jak 1024 KiloByte (což by bylo 1 048 576 Byte), proto se číslo převede na kB, tedy 96kB.

Testovací hodnoty (Š x V x H):

- 1024x768x1
- 1920x1080x24

demo
