

# Visual Basic – základní prvky jazyka

---

© Aleš Keprt – listopad 2008, prosinec 2009, listopad 2010

Vytvořeno pro potřeby výuky na UP Olomouc a MVŠO. Tento text není určen pro samostudium, ale jen jako vodítko pro učitele, takže jeho obsah se může čtenáři zdát stručný, nekompletní či možná i chybný. Použití je povoleno dle vlastní libosti, ale jen na vlastní nebezpečí. ☺ V případě dalšího šíření je NUTNO uvádět původního autora a odkaz na původní dokument. Komentáře můžete posílat e-mailem autorovi (adresu najdete přes Google).

## Základní informace

Každý příkaz se píše na samostatný řádek. Možno také libovolně vkládat prázdné řádky.

Velká a malá písmena se u příkazů nerozlišují.

Násobení se značí hvězdičkou \*, dělení lomítkem /.

Text za znakem ' (apostrofov - na anglické klávesnici je to vedle Enter) je poznámka. Zobrazuje se zeleně a můžeme tam psát cokoliv, není to součást programu.

## Proměnné

Proměnná je pojmenovaná paměťová buňka. Má jméno a hodnotu.

Proměnnou nejprve založíme příkazem **Dim jméno** a pak používáme.

Do proměnné přiřadíme hodnotu pomocí rovnítka, příklad: **a = 23** nebo **a = b + c** nebo **a = a + 2**

Proměnná může obsahovat i text – přesněji textový řetězec, píšeme jej do uvozovek.

Co je v uvozovkách, se matematicky nevyhodnocuje – vyzkoušejte třeba tohle:

**Dim a,b**

**a = 123 + 456** 'součet čísel, výsledek je číslo 579

**b = "123" + "456"** 'spojení textu, výsledek je text 123456

Proměnná může mít tzv. typ. Typ určuje, jaké hodnoty lze do proměnné přiřadit.

**Sub a as Integer** 'celé číslo

**Sub b as Double** 'racionální číslo (celé, desetinné, zlomky...)

**Sub c as String** 'textový řetězec

Typy se v praxi používají na zrychlení programu a snazší odhalování chyb. Vám se mohou hodit k explicitnímu rozlišení čísel a textu a také k vynucení zaokrouhlování čísel na celá (třeba aby po dělení lichého čísla dvěma nebyla ve výsledku polovina, která nám může překážet).

## Psaní na obrazovku

**Console.WriteLine("text" + "další text" + 123)**

Tento příkaz napíše to, co je v závorkách, a odřádkuje (tj. přesune se na další řádek).

Nechceme-li odřádkovat, použijeme místo **WriteLine** jen **Write**:

**Console.Write("Toto nebude odřádkováno")**

Chceme-li naopak jen odřádkovat, do závorek za **WriteLine** nenapíšeme nic:

**Console.WriteLine()** 'nic nevypíše, jen odřádkuje

V závorce může být i nějaký výpočet či volání funkce:

**Console.WriteLine("Hodnota sin(10) je " + Math.Sin(10))**

## Vstup z klávesnice

Základní vstup z klávesnice: Uživatel zadá hodnotu a zmáčkne Enter, přiřadíme ji pak do proměnné. Většinou mu předchází použití **Write**, kde uživateli napíšeme, jakou hodnotu od něj chceme zadat:

```
Console.Write("Zadej nějaké číslo: ")
```

Dim proměnná

```
proměnná = Console.ReadLine()
```

## Opakování kódu

**While** podmínka

*příkazy-k-opakování*

End While

Místo prostředního řádku doplníme jakýkoliv příkaz nebo i víc příkazů. Celý tento „blok“ příkazů se opakuje tak dlouho, dokud platí uvedená podmínka. Jako podmínku uvádíme nějakou nerovnost.

Následující příklad vypíše sudá čísla od 10 do 20:

Dim a = 10

While a <= 20

```
    Console.WriteLine(a)
```

```
    a = a + 2
```

End while

## Podmínka (podmíněné vykonání)

Toto funguje skoro stejně jako příkaz While, kód se však provede jen jednou.

**If** podmínka Then

*'příkazy-když-podmínka-platí*

End If

Navíc můžeme určit kód k vykonání, když podmínka neplatí.

**If** podmínka Then

*'příkazy-když-podmínka-platí*

Else

*'příkazy-když-podmínka-neplatí*

End If

Tento příkaz má ještě krátkou jednořádkovou variantu, kterou můžeme používat tam, kde chceme při splnění podmínky vykonat jen jeden krátký příkaz.

```
If a>0 Then Console.WriteLine("a je kladné")
```

## Podprogramy

Podprogram je pojmenovaná část programu. Používáme ji pak jako nový příkaz. Smyslem podprogramů je, že nám umožní nějakou činnost (řadu příkazů) napsat jen jednou, ale vykonávat opakovaně na různých místech programu. Visual Basic již má řadu takových podprogramů – např.

**Console.WriteLine** je podprogram, který umí psát na obrazovku, **Math.Sin** je podprogram, který

vypočítá hodnotu funkce sinus. Na těchto příkladech je také vidět, jak se hotové podprogramy používají, nyní se je naučíme také tvořit.

Základním typem podprogramu je **procedura** – je to jednoduše několik příkazů za sebou.

**Sub jméno()**

'příkazy

**End Sub**

Dalším typem podprogramu je **funkce**. Od procedury se liší tím, že má výsledek – hodnotu výpočtu.

**Function jméno()**

'příkazy

**return výsledek**

**End Function**

Výsledkem je obvykle nějaké číslo. Může to být jakákoliv hodnota, kterou lze přiřadit do proměnné. Říkame, že „funkce vrací“ nějakou hodnotu.

Podprogramy mohou mít parametry. Parametry slouží k upřesnění, co má podprogram dělat. Jako příklad uvedme funkci, která vrací dvojnásobek hodnoty parametru:

**Function dvojnásobek(ByVal hodnota)**

**return hodnota \* 2**

**End Function**

Parametry mohou mít určen typ, stejně jako proměnné.

Podprogram může mít i víc parametrů, oddělují se čárkami.

Příklad: Procedura vypíše násobek dvou čísel:

**Sub VypišNásobek(ByVal a, ByVal b)**

**Console.WriteLine(a \* b)**

**End Sub**

Parametry používáme v podprogramu stejně jako proměnné – každý má jméno a hodnotu.

Konstrukce **ByVal** tedy jakoby nahrazuje obvyklé **Dim**.

## Příklady

Pro vyzkoušení Basicu zkuste „nakreslit“ v konzoli následující obrazce. Programy se snažte psát obecně, aby fungovaly pro libovolnou velikost obrazců.

```
*      1. Trojúhelník – každý řádek má o jednu hvězdičku více než předchozí
**
***
****
*****
```

```
***** 2. Obrácený trojúhelník – každý řádek má o jednu hvězdičku méně než předchozí
****
***
**
*
```

```

* 3. Trojúhelník zprava – každý řádek má o jednu hvězdičku víc než předchodí a je to zarovnáno doprava
* *
* * *
* * * *
* * * * *

```

```

* 4. Prázdný trojúhelník – hvězdičky jsou jen po obvodu
* *
* * *
* * * *
* * * * *

```

```

* * * * * 5. Prázdný čtverec – hvězdičky jsou jen po obvodu
* *
* *
* *
* * * * *

```

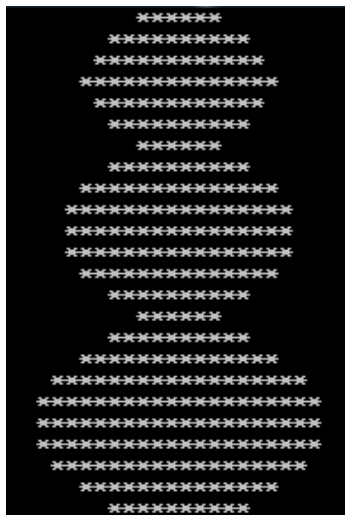
```

* * * * * * 6. Vepsané čtverce – kolik se jich tam vejde, závisí na velikosti obrazce
* *
* * * * *
* * * * *
* * * * *
* * * * *
* *
* * * * *

```

## Vánoční příklady

Sněhuláka uvádím i s řešením, ale zkuste si ho nakreslit sami, jestli to zvládnete.

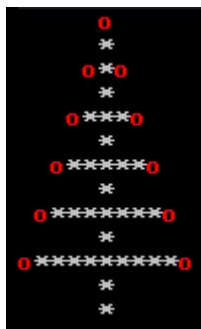


```

Sub Kruh(ByVal prumer)
    Dim radek = 1
    While radek + 1 < prumer
        Dim w As Integer =
            prumer / 1.2 * Math.Sin(radek / prumer * Math.PI)
        Znaky(" ", 40 - w)
        Znaky("*", 2 * w)
        Console.WriteLine()
        radek = radek + 1
    End While
End Sub

Sub Main()
    Kruh(8)
    Kruh(10)
    Kruh(12)
End Sub

```



Nejprve udělejte stromeček z hvězdiček, jako na obrázku. Potom ho už ověste červenými baňkami. Jednu baňku nakreslí následující podprogram, zbytek je váš úkol.

```

Sub Banka()
    Console.ForegroundColor = ConsoleColor.Red
    Console.Write("o")
    Console.ForegroundColor = ConsoleColor.Gray
End Sub

```