



Programování

Letem světem programkem - pokračování



Funkce

- Funkce/metody, které používáme pro volání podprogramů
- Knihovny MSDN obsahují mnoho užitečných funkcí
- Lze vytvářet i vlastní funkce
- Náležitosti každé funkce:
 - Jméno (začíná velkým písmenem)
 - Návrátový datový typ
 - Viditelnost funkce
 - Volitelně u funkce uvádíme seznam vstupních parametrů a další speciální modifikátory funkce

Enumerace

- Výčtový datový typ, speciální datový typ
- Umožňuje programátorům vytvořit lépe čitelný seznam hodnot, které jsou reprezentovány jako číselné hodnoty
- Pro vytvoření enumerace využíváme klíčové slovo **enum**
- Jakmile máme definovaný výčtový typ, lze vytvořit proměnnou odpovídající tomuto výčtu
- Bez dodatečných zásahů jsou jednotlivé položky číslovány od 0
 - Máme ovšem možnost změnit výchozí hodnotu
- Následující hodnota je automaticky větší o hodnotu 1



Struktura

- Datový typ, který nám umožňuje vytvořit vlastní strukturu
- Struktura je podobná třídě (OOP), ale má omezené možnosti
- Vhodná pro definici vlastních komplexnějších datových typů
- Struktura obsahuje několik atributů, které ji charakterizují
 - Atributy jsou definovány pomocí jednoduchých datových typů
- Pro vytváření struktury využíváme klíčového slova **struct**
- Struktura nám umožňuje vytvořit vlastní přidružené funkce se strukturou spojené včetně konstruktoru
 - Konstruktory jsou funkcí, které nám vytváří novou instanci struktury

Vlastnosti a metody Pole (Array)

- **Length** – atribut pole obsahující počet prvků v poli
- **Copy()** – metoda, která zkopíruje obsah pole do jiného pole
- **Sort()** – metody, která seřadí prvky v poli
- **IndexOf()** – metoda, sloužící k nalezení pozice daného prvku v poli
 - Pokud se zde prvek nenachází vrací se hodnota -1
- Všechny uvedené metody se volají skrze třídu Array, která tyto funkce obsahuje
- Vstupním parametrem těchto funkcí je pole
- Např.: `Array.Sort();`

Vlastnosti a metody String

- **string** nepatří mezi jednoduché datové typy
- Spojování řetězců lze provádět pomocí operátoru **+**
- **Length** – atribut délky řetězce
- **Substring()** – metoda pro nalezení podřetězce v řetězci
- **Equals()** – metoda pro porovnání řetězců
- **Split()** – metoda, která na základě rozdělovacího znaku rozdělí řetězec a vloží jej do příslušného pole
- Všechny funkce datového typu **string** se volají jako metody konkrétního řetězce
- Např.: `"message".Split();`



Vlastnosti a metody Seznamu (List)

- Dynamická datová struktura
- **Count** – vlastnost seznamu, obsahující počet položek v listu
- **Add()** – vložení položky na konec seznamu
- **Insert()** – vložení položky na konkrétní místo
- **Remove()** – odstranění prvního výskytu položky
- **RemoveAt()** – odstranění položky na konkrétní pozici
- **Contains()** – metoda zjišťující, zda se v seznamu nachází konkrétní položka
- **Clear()** – odstranění všech položek ze seznamu
- Všechny funkce List jsou volány skrze konkrétní seznam



Rekurze

- Programovací princip jehož hlavní myšlenkou je opakování něčeho
- V našem případě pro rekurzi platí, že funkce ve svém těle volá samu sebe
- Musíme si dát pozor na zacyklení
- Vždy musí existovat něco, čemu říkáme rekurzivní zarážka
- Soubor dat, který se v rekurzi volá se nesmí zvětšovat
- Funkce můžeme volat rekurzivně buď přímo nebo nepřímo



Třídy Math, Random

- Další z tříd, které nám ulehčují práci
- Třída Math obsahuje matematické konstanty a běžně používané funkce
- Goniometrické funkce, mocniny, statistické funkce, ...
- Třída Random nám umožňuje vytvářet náhodné a pseudonáhodné hodnoty
- Vhodné jsou například pro generování náhodného pořadí
- Na tyto třídy lze pohlížet jako na knihovny funkcí a konstant