

Algoritmy a programování III

Pole

- `typ[] nazevPole;`

int[] cisla;

string[] jmena;

Pole

Deklarace pole

```
int[] ciska = new int[10];
```

- Pole 10 prvků typu int bez inicializace. Prvky se v takovém případě inicializují hodnotou 0 a nemusí se dále inicializovat.

```
int[] ciska = new int[] { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

```
int[] ciska = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };
```

Pole

```
int[] ciska = new int[5];
```

```
for (int i = 0; i < 5; i++){  
    ciska[i] = i + 1;}
```

Délka pole – Length

```
int[] ciska = new int[5];
```

```
for (int i = 0; i < ciska.Length; i++){  
    ciska[i] = i + 1;}
```

Pole

- Vypis

```
int[] ciska = new int[5];  
for (int i = 0; i < ciska.Length; i++){  
    ciska[i] = i + 1;  
    Console.Write("{0} ", ciska[i]);}
```

Pole

- Foreach

foreach (datovotyp promenna in kolekce) { // příkazy }

```
public static void Main(string[] args)
{
    int[] cisla = new int[10];
    for (int i = 0; i < 10; i++)
        cisla[i] = i + 1;
    foreach (int i in cisla)
        Console.WriteLine("{0} ", i);
    Console.ReadKey();
}
```

Cvičení

- Naplňte pole pomocí vstupu od uživatele a vypište pole

Pole

Užitečné metody:

- Min() - nejmenší prvek
- Max(), - největší prvek
- Average() - průměr
- Sum() - součet
- using System.Linq;

```
static void Main() {  
    int[] ciska = {3, 4, 5, 6};  
    Console.WriteLine(ciska.Sum());  
}
```


Cvičení

- Spočítejte průměr uživatelových známek

Objektově orientované programování

Objektově orientované programování

Základní pojmy

- **Třída** - představuje skupinu objektů, které nesou stejné vlastnosti.
- **Objekt** - jeden konkrétní jedinec (reprezentant, entita) příslušné třídy pro konkrétní objekt nabývají vlastnosti deklarované třídou konkrétních hodnot

Příklad:

- Třída Clovek má vlastnost jmeno => objekt Jan Novák

Objektově orientované programování

Proměnné

- atributy, charakteristika objektu

Metody

- Dovednosti objektu

Objektově orientované programování

Třída

Class NovaTrida{ //kod}

- Internal - výchozí, viditelná pouze v daném projektu
- Public - veřejná, viditelná všude
- Partial - Rozložena na více částí
- Abstract, static – nelze vytvořit instanci
- Sealed – v hierarchii dědění poslední.

Instanční proměnné

- Public
- Protected - druhý "level" oprávnění
- Private – pouze v třídě
- Static – lze volat pomocí tečkové notace (samozřejmě musí být veřejný)

Konstruktory

- Speciální metoda, která má stejný název jako třída
- volá ve chvíli vytváření instance této třídy.