



Programování

Opakování – datové struktury



Datová struktura

- = konkrétní způsob uložení dat v paměti
- Homogenní datové struktury
 - Všechny položky uložené mají stejný typ/formát
 - Pole, spojovaný seznam, zásobník, strom, halda
- Nehomogenní datové struktury
 - Každá z položek může být různého typu
- V závislosti na použité datové struktuře lze ovlivnit rychlost práce algoritmu
 - Rozlišujeme rychlost vložení, odstranění, nalezení hodnoty, seřazení struktury

Pole

indexy	0	1	2	3	4	5	6	7
	15	3	21	8	3	12	5	3

- Statická homogenní datová struktura
- První položka je uložena na indexu 0
- Vytvoření nového pole
- Atribut **Length** = délka pole
- poleInts.Length

// deklarace prázdného pole int

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

// deklarace a inicializace pole double

```
double [] poleDoubles = {1.3,2.8,3.6};
```

// vícerozměrné pole

```
bool[,] pole2D = new bool[4,4];
```

Class Array

- Třída umožňující vytvoření a práci s polem
- V závislosti na datovém typu pole se liší funkce použitelné
- Každé pole lze neohledě na typ seřazovat, vyhledávat v poli
- Pole obsahující číselné hodnoty lze navíc sčítat, vyhledávat extrémy, ...

```
int[] pole = {14,5,78,-10,0,1};  
  
int soucet = pole.Sum();  
  
int max = pole.Max();  
  
// seřazení pole  
pole.Sort();
```

Seznam

- Dynamická homogenní datová struktura
- Nově přidávané položky jsou umisťovány na poslední pozici
- Atribut **Size** = počet prvků v seznamu
- osoby.Size

```
// vytvoření novéh seznamu řetězců  
List<string> osoby = new List<string>();  
  
// přidání hodnot do seznamu  
osoby.Add("Kamil");  
osoby.Add("Radim");
```

Class List

- Třída umožňující vytvoření a práci se seznamem
- V seznamech lze rovněž vyhledávat, řadit je a převádět na pole
- Oproti polím můžeme odstranit položku na určité pozici
 - Jednotlivé položky v seznamu se přeskládají a napojí
- Pokud jsou seznamy číselného typu, rovněž lze aplikovat stejné metody jako u pole
 - Součet hodnot
 - Hledání extrémů

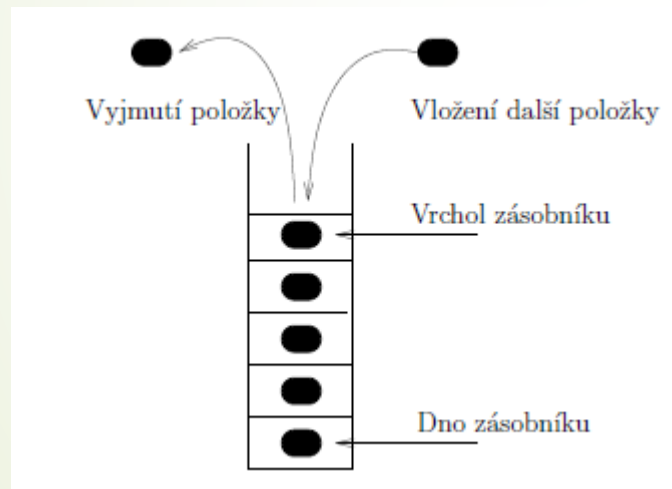
```
List<int> cisla = new List<int>();
```

```
int sum = cisla.Sum();
```

```
int max = cisla.Max();
```

Datové struktury fronta a zásobník

LIFO - zásobník



FIFO - fronta

