



# Programování

Vícerozměrná pole

# Pole

- ▶ Statická homogenní datová struktura
  - ▶ umožňuje uchovávat více hodnot stejného datového typu
  - ▶ Po stanovení velikosti pole již nelze v průběhu měnit počet prvků
- ▶ Dle datového typu, jednotlivých položek pole lze využívat speciální funkce – řazení, suma, průměr, vyhledávání, ...
- ▶ Pro práci s konkrétní položkou musíme uvést její index (začínáme od 0)

```
double[] novePole = new double[4];

bool[] pravdivostniPole = { true, false, true, true, false };

string[] poleSlov = { "ahoj", "svete", "jsem", "tady" };

int[] poleCisel = new int[3];
poleCisel[0] = 1;
poleCisel[2] = 4;
```



# Dvourozměrné pole pravoúhlé

- U běžného pole víme, že obsah jednotlivých položek musí být homogenní
- Konkrétní položkou v poli může být další pole
- V případě, že délky polí na jednotlivých indexe základního pole jsou shodné, bavíme se tzv. pravoúhlém poli

```
int[,] pravouhlePole = new int[2,3];
```

```
int[,] ctvercovePole = { { 1, 2 }, { 3, 4 } };
```

# Průchod pravoúhlým polem

- Využijeme deklarace pole z předešlého snímku

```
for (int i = 0; i < ctvercovePole.GetLength(0); i++)  
    for (int j = 0; j < ctvercovePole.GetLength(1); j++)  
        Console.WriteLine(ctvercovePole[i, j]);
```

- Na výstup se postupně vypíše čísla 1, 2, 3, 4
- Pomocí funkce **GetLength()** získáváme délku pole pro konkrétní dimenzi

# Dvourozměrné pole - zubaté

- Obdobně jako u pravoúhlého pole jsou pod jednotlivými prvky další pole
- Jednotlivá pole mohou mít rozdílnou délku

```
int[][] zubatePole = new int[2][];  
zubatePole[0] = new int[5];  
zubatePole[1] = new int[3];
```



# Průchod zubatým polem

```
for (int i = 0; i < zubatePole.Length; i++)  
    for (int j = 0; j < zubatePole[i].Length; j++)  
        Console.WriteLine(zubatePole[i][j]);
```

```
for (int i = 0; i < zubatePole.GetLength(0); i++)  
    for (int j = 0; j < zubatePole[i].GetLength(0); j++)  
        Console.WriteLine(zubatePole[i][j]);
```

## Uložení vícerozměrných polí v paměti

- U jednorozměrného pole víme, že jednotlivé položky jsou v paměti uloženy přesně za sebou
- U vícerozměrného je tomu obdobně za sebou jsou však uloženy jednotlivé "sloupce" daného pole
- V paměti je tedy nejprve uložen sloupec A a následně sloupce B, C
- V některých jazycích můžeme za pomoci aritmetiky přímo přistupovat na konkrétní pole bez nutnosti znát oba indexy pole

A	B	C
15	3	9
2	3	1
8	7	16
5		13
3		