

# Datový typ pole – vícerozměrné, jednorozměrné

Do pole můžeme uložit více proměnných stejného typu.

```
type[] arrayName;
```

Počet prvků musí být nastaven při vytvoření pole a poté nejde změnit.

Je indexováno od 0.

```
// Declare a single-dimensional array of 5 integers.
int[] array1 = new int[5];

// Declare and set array element values.
int[] array2 = [1, 2, 3, 4, 5, 6];

// Declare a two dimensional array.
int[,] multiDimensionalArray1 = new int[2, 3];

// Declare and set array element values.
int[,] multiDimensionalArray2 = { { 1, 2, 3 }, { 4, 5, 6 } };

// Declare a jagged array.
int[][] jaggedArray = new int[6][];

// Set the values of the first array in the jagged array structure.
jaggedArray[0] = [1, 2, 3, 4];
```

K výpisu používáme cykly

```
int[] pole = new int[10];
pole[0] = 1;
for (int i = 0; i < 10; i++)
    pole[i] = i + 1;
foreach (int i in pole)
    Console.WriteLine("{0} ", i);
Console.ReadKey();
```

```
1 2 3 4 5 6 7 8 9 10
```

## Dvourozměrné pole

První údaj je řádek, druhý je sloupec

X:	0	1	2	3	4
Y: 0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	1	0	0
3	0	1	1	1	0
4	1	1	1	1	1

Lze si představit jako tabulku

Deklarace `int[,] kinosal = new int [5, 5];`

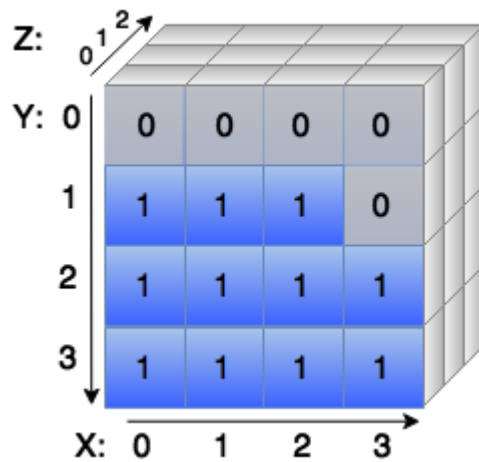
Vypíšeme dvěma cykly

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

```
00000
00000
00100
01110
11111
```

## N rozměrné pole

Např. 3D



```
int[, ,] kinosaly = new int [4, 4, 3];
```

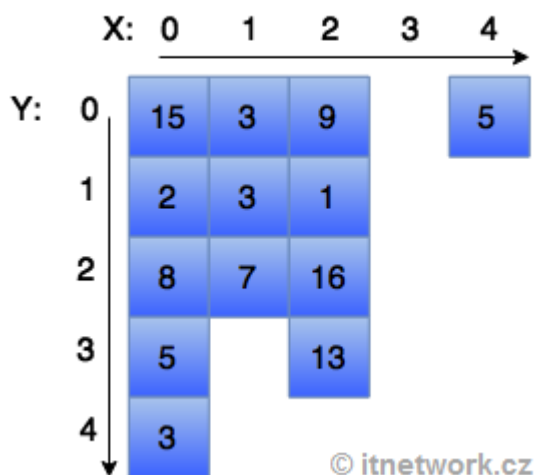
```
kinosaly[3, 2, 1] = 1; // Druhý kinosál, třetí řada, čtvrtý sloupec
```

Pokud metodě `GetLength()` zadáme parametr s hodnotou `2`, získáváme počet "pater" (kinosálů).

## Pole polí – jagged

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

Do každého řádku si můžeme vytvořit, jak velké chceme pole = šetříme paměť



Nevýhodou je složité zadávání

```
for (int i = 0; i < kinosal.Length; i++)  
{  
    kinosal[i] = new int[5];  
}
```

Také získat velikost pole je komplikované

```
int sloupcu = kinosal.Length;  
int radku = 0;  
if (sloupcu != 0)  
    radku = kinosal[0].Length;
```

K hodnotám přistupujeme pomocí indexu, když zadáme jenom 1 vrátí se všechny hodnoty v sloupečku

## Zkrácená inicializace

Lze inicializovat podle hodnot

```
int[,] kinosal = new int[,] {  
    { 0, 0, 0, 0, 1 },  
    { 0, 0, 0, 1, 1 },  
    { 0, 0, 1, 1, 1 },  
    { 0, 0, 0, 1, 1 },  
    { 0, 0, 0, 0, 1 }  
};
```

```
int[][] zubatePole = new int[][] {  
    new int[] {15, 2, 8, 5, 3},  
    new int[] {3, 3, 7},  
    new int[] {9, 1, 16, 13},  
    new int[] {},  
    new int[] {5}  
};
```

Příklad: vytvořte program který vytvoří obdélníček z \*

```
Console.WriteLine("Zadejte počet řádků: ");  
int radky = int.Parse(Console.ReadLine());  
Console.WriteLine("Zadejte počet sloupců: ");  
int sloupce = int.Parse(Console.ReadLine());  
  
string[,] pole = new string[radky, sloupce];  
  
for (int i = 0; i < radky; i++)  
{  
    for (int j = 0; j < sloupce; j++)  
    {  
        if ((i == 0) || (i == radky - 1) || (j == 0) || (j == sloupce - 1))  
        {  
            pole[i, j] = "*";  
        }  
        else  
        {  
            pole[i, j] = " ";  
        }  
    }  
}  
  
for (int i = 0; i < radky; i++)  
{  
    for (int j = 0; j < sloupce; j++)  
    {  
        Console.Write(pole[i, j] + " ");  
    }  
    Console.WriteLine();  
}  
Console.ReadKey();
```

Zadejte počet řádků:

5

Zadejte počet sloupců:

4

\* \* \* \*

\*       \*

\*       \*

\*       \*

\* \* \* \*