

Střední škola pedagogická, hotelnictví a služeb, Litoměřice, příspěvková organizace

Předmět:	Vývoj aplikací
Téma:	Pole
Vyučující:	Ing. Milan Káža
Třída:	EK3
Hodina:	14
Číslo:	V/5

Programování v jazyce C a C#

12. Pole

Pole je strukturovaný homogenní (všechny prvky v poli jsou stejné) datový typ. To znamená že, obsahuje několik prvků stejného datového typu se kterými je možné pracovat. K jednotlivým prvkům pole se přistupuje postupně a také se s nimi tak pracuje. Pole může mít různou velikost, začátek je dán ukazatelem, konec pole ale není dán proto je nutné ho hlídat.

Pole není záhada ani složitost, pole jsou zase jen data

Pole je možné rozdělit na **statické a dynamické** dále je možné rozdělit na **jednorozměrné a vícerozměrné**. V jazyce C se s poli pracuje jinak než v jazyce C#. S poli je možné pracovat i v PHP a práce s nimi je blíže k původnímu jazyku C.

12.1. Pole v jazyce C

V jazyce C je pole v podstatě pointer (byl popsán v předcházející části bude) obsahující adresu nultého prvku pole. V jazyce C začíná pole vždy nultým prvkem a to je z důvodu pro zvýšení efektivity pro přístup do pole.

Jednorozměrné pole

```
int cisla[10]; //pole pro 10 čísel
char pole_znaku[20]; //pole pro 20 znaků
```

tímto způsobem se zadává jednorozměrné pole, a je to pole statické, protože je znám počet prvků v poli. Jednorozměrné pole je možné zapsat i tímto způsobem:

```
int pocet[5]={1,2,3,4,5}; //počet prvků v poli je tedy 5
char slovo[]="Ahoj lidi"; //zadání řetězce
```

Pokud se zadává slovo tak program porovnává shodnost zda se zadané slovo nachází někde v textu např. v souboru.

Přístup k prvkům

K jednotlivým prvkům pole se přistupuje pomocí operátoru indexu [], který je uveden za identifikátorem pole. Uvnitř závorek operátoru je pak potřeba uvést celé číslo odpovídající indexu prvku v poli. V jazyce C je to nula.

Příklad ukazuje jak se přistupuje j jednotlivým prvkům pole:

```
int w;  
int prvky[10];  
  
for(w=0;q<10; q++){  
    prvky[q]=q+1;  
}  
for(q=0; q<10; q++){  
    printf("%i",prvky[q]);  
}
```

Příklad je vhodné si vyzkoušet a ve vývojovém prostředí Borland C si program odkrokovat.

Vícerozměrné pole

Jazyk C umožňuje, aby pole bylo vícerozměrné, tedy bylo možné zadat dvě a více polí. Nejčastěji se zadává dvourozměrné pole i pro vícerozměrná pole platí, že začínají nulou.

Zadání vícerozměrného pole:

```
int znak[2][3];
```

Zde je inicializace dvojrozměrného pole, které bude mít velikost 2 a 3, tedy alokuje v paměti 2*3 prvků.

Uložení prvků pole v paměti

Prvky pole jsou v paměti ukládány ve stejném pořadí, v jakém se uvádějí při inicializaci pole. Například u pole `int znak[2][3]` jsou v paměti uloženy v tomto pořadí: `x[0][0]`, `x[0][1]`, `x[0][2]`, `x[1][0]`, `x[1][1]`, `x[1][2]`.

Na n -rozměrné pole se lze také dívat jako na jednorozměrné pole, jehož prvky jsou $(n-1)$ -rozměrná pole. Výše uvedené dvourozměrné pole `x` tedy může být chápáno jako jednorozměrné pole typu `int [3]` o velikosti 2.

12.2. Pole v jazyce C#

Pole v jazyce C# jsou indexována od nuly jako v jazyce C, jinak jsou mezi poli v jazyce C a C# rozdíly. Hlavní rozdíly je možné shrnout do těchto bodů:

- S poli se zachází prostřednictvím odkazů a vytváří se pouze pomocí `new`. O zrušení pole se stará automatická správa paměti.
- S polem se v C# pracuje jako s celkem.
- Při přístupu k prvkům se vždy zkontroluje, zda leží požadovaný index v rozmezí určeném deklarací, v případě že ne, tak vzniká výjimka typu: **`System.IndexOutOfRangeException`**

Pole jsou v C# odvozeny od třídy **`System.Array`** a je možné ji použít s příkazem **`foreach`**.

Jednorozměrné pole

Jednorozměrné pole je s jedním indexem a má tuto deklaraci: `int [] pole=new int[10]`

`int[]` je typ prvku pole zde se neuvádí v hranatých závorkách číslo

`pole` je identifikátor

`new` je instance pro jednorozměrné pole za `new` znovu typ např. `int` a do hranatých závorek se napíše počet prvků pole

V programu se deklarace jednorozměrného pole zadává tímto způsobem

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

Nula se musí napsat, protože pole v C# začíná vždy od nuly.

K jednotlivým prvkům se přistupuje pomocí operátoru indexování jako v jazyce C, ale na rozdíl od jazyka C se v jazyce C# se kontroluje, zda index leží v mezích dané deklarace.

Jazyk C# nabízí možnost zjistit délku pole a to pomocí příkazu `Lenght`. Tento příkaz je možné použít např. při výpisu délky pole, pokud není známá jeho délka. Zde je ukázka kódu pro zjištění délky pole.

```
public static void Vypis(double[] pole)
{
    for (int q=0; q<pole.Lenght;q++)
        Console.WriteLine(pole[q]);
}
```

Tento příklad zjistí jak dlouhý je text v souboru, nebo kolik má věta slov.

Pole je možné také zkopírovat do jiného pole a to příkazem **CopyTo** a do kódu programu se zapisuje tímto způsobem:

```
pole.CopyTo(wpole, 3);
```

Tento kód zajistí, že se prvky z **pole** okopírují do prvků **wpole** a uloží se v něm prvky počínaje indexem 3.

Vícerozměrné pole

Specifikace vícerozměrného pole se provádí stejně jako u jednorozměrného pole pouze se v hranatých závorkách napíše čárka.

Inicializace vícerozměrného pole se zapisuje tímto způsobem:

```
int[,] pole2=new int[2,3]
```

Pole se skládá ze dvou prvků a přistupuje se k němu po jednotlivých prvcích.

Dynamická pole

Tento typ polí se objevuje především v jazyce C. Ve většině případů ví programátor jak velké bude pole, ale může nastat případ, kdy tato skutečnost není známá a proto se vytváří dynamické pole. Důležité je aby všechny prvky pole byly stejného typu např. **int**. Je nutné předem alokovat paměť pro uložení dynamického pole a to se děje až za chodu programu.

Dynamické pole se zadefinuje pomocí ukazatele **int *p_cislo** tak je do programového kódu zadáné, že někde bude identifikátor tohoto datového typu a dále se musí alokovat paměť, k alokování paměti se použije příkaz `malloc()` a celý zápis do kódu programu se zapíše: `p_cislo = (int *) malloc(POCET * sizeof(int));`

malloc - přidělování paměti během běhu programu.

sizeof - vrací počet bajtů, které obsadila alokace pole v paměti. Dle počtu prvků a typu použitých prvků je dána celková alokovaná oblast. Zabere se určitý počet Bajtů v paměti.

POCET - je buď přímá hodnota nebo symbolická konstanta.

Zde je vidět výhoda dynamických polí, že se rozhodne až během běhu programu o velikosti přidělené paměti a to podle aktuální situace. Neplýtvá se tedy paměť.

Po ukončení práce s dynamickým pole je nutné pole zrušit **p_cislo=0**; tím je zajištěno vynulování pole a uvolnění paměti.

S touto formou pole je možné se setkat v jazyce **Borland C++ 3.1** a v jeho dalších produktech, dynamické pole se využívala především v době MS-DOS, kdy velikost harddisku byla např 50MB, což v současné době jsou obrovské úložiště, v jazyce C# tento způsob pole není.

Pole v PHP

S poli se také pracuje v jazyce PHP a slouží především pro usnadnění práce s MySQL. Pole se vytváří příkazem `array` a do závorek se vkládají prvky. Pole se v PHP zavádí takto:

```
array("prvek1", "prvek2", "prvek3")
```

každý prvek má své číslo a začíná se vždy od nuly, stejně jako v C nebo C#.

Příklad zavedení pole v PHP

```
... *
    $pole = array ("mrkev", "celer", "brambory");
        echo($pole[0]); //vypíše 'mrkev';

$zahrada = array (3 => "jablon", "hrusen");
echo($zahrada[4]); //vypíše 'hrusen';
... *
```

tato ukázka programu je stažena z internetových stránek <http://www.tvorba-webu.cz/php>, kde je možné si nastudovat problematiku PHP.

Pole jako program bude vytvořen ve Visual Studio v kapitole ..

Kontrolní otázky:

1. Vysvětlete pojem pole
2. Rozdělte pole podle velikosti
3. Řekněte kde všude je možné zavést pole a jeho specifikace

Použitá literatura:

Miroslav Virius: C# pro zelenáče, 1.vydání, Neocortex, s.r.o Praha, 2002

Pavel Herout: Učebnice jazyka C, 3.vydání, KOOP České Budějovice, 1997

Miroslav Virius: od C++ k C#, 1.vydání, KOOP České Budějovice, 2002