

1. Polia v jazyku C++

Niekedy potrebujeme použiť väčšie množstvo premenných, ku ktorým budeme pristupovať ako k prvému, druhému, ... prvku v množine týchto premenných. Vtedy použijeme dátový typ pole. Každá premenná z tejto množiny je daná len jedným menom, ale rozlišujeme ich podľa poradia (indexu) premennej v tejto množine. Tento dátový typ nazývame pole.

Dátové pole je nejaké spojité miesto vyhradené kdesi v pamäti pre niekoľko objektov rovnakého dátového typu.

Pred použitím poľa je nutné ho deklarovať (vyhradiť preňho miesto v pamäti):

Deklarácia poľa:

Uda jovyTyp NazovPremennej[PocetPrvkov];

Príklady deklarácie poľa:

```
int nCislo[10];
char cZnak[18];
```

Je potrebné si uvedomiť, index v hranatej zátvorke začína od nuly a teda deklarácia

```
int nCislo[10];
```

vyhradí v pamäti miesto pre 10 premenných od **nCislo[0]** po **nCislo[9]** a teda prvok poľa **nCislo[10]** neexistuje!!!!!!!

Ako všetky premenné aj tieto je vhodné hneď po deklarácii hneď aj inicializovať, teda vložiť do nich nejaké hodnoty:

```
int nCislo[10] = {2, 5, 8, 0, -5, 3, 14, -2, 32, 1};
```

Ak majú prvky poľa len jeden index, hovoríme o jednorozmernom poli. Polia však môžu byť aj viacrozmerné: dvojrozmerné, trojrozmerné i viacrozmerné:

```
int nZnamka[4][4][4];
```

Počet prvkov takéhoto poľa je $4 \times 4 \times 4 = 64$. I takéto pole sa dá ihneď inicializovať:

```
int nZnamka[2][2][2] = {2, 4, 1, 5, 1, 2, 1, 3};
```

Krajšie to ale vyzerá takto:

```
int nZnamka[2][2][2] = {{{2, 4}, {1, 5}}, {{1, 2}, {1, 3}}};
```

A ešte krajšie:

```
int nZnamka[2][2][2] = {{{2, 4},
                        {1, 5}},
                        {{1, 2},
                        {1, 3}}};
```

Príklad Známky žiakov:

Majme dve školy, v každej dve triedy a v každej dvoch žiakov, ktorým zadáme známky (zatiaľ z jedného predmetu). Našou úlohou bude:

- Zadať mená škôl
- Zadať mená tried
- Zadať mená žiakov
- Zadať ich známky
- Rôznym spôsobom tieto údaje spracovať a výsledky poslať na výstup (konzolu).

Postup:

Postup budem kombinovať s výpismi z kódu spolu s vloženými poznámkami.

Začneme vytvorením **CLR Empty Projectu** vo vlastnom adresári, pričom dodržiujeme pravidlá (žiadne interpunkčné znamienka, bodky, medzery ...). Do podadresára **Source Files** vložíme prázdny **cpp** súbor. Tento sa **nemusí** volať **main.cpp** (Len funkcia v ňom – prvá volaná funkcia programu sa tak volať musí). Nezabudnime po napísaní kostry program hneď preložiť, aby sme si boli istí, že neobsahuje chyby.

```
#include <iostream>
#include <string>
using namespace std;

int main(void)
{
    // pozastavenie programu pred jeho skončením (vždy na konci programu)
    cout << "Zadaním celeho čísla ukončí program " << endl;
    int nEnd = 0;
    cin >> nEnd;
    return 0;
}
```

Ako prvé vložíme do programu mená škôl:

```
int main(void)
{
    string sSkola[3]; //pre jednoduchost budeme skoly indexovat 1 a 2
                     // hadam si mozme dovolit "zahodit" sSkola[0]
                     //ktora ale existuje a teda mame tri prvky pola
    //Zadaj meno skoly
    for(int k = 1; k <= 2; k++)
    {
        cout << "Zadaj meno skoly c. " << k << endl;
        cin >> sSkola[k];
    }
}
```

Úloha:

Dajte hneď vypísať mená škôl.

Ak to máme, aby sme v budúcnosti šetrili čas, zapoznámkujeme ručné zadávanie a budeme názvy škôl inicializovať pri deklarácii: (meno **sSkola** sme zvolili aby každý videl že sa jedná o dátový typ string).

```
string sSkola[3] = {"NultaSkola", "Orange", "Javorka"};
//Vypis skoly
for(int k = 1; k <= 2; k++)
{
    cout << "Skola c. " << k << " je " << sSkola[k] << endl;
}
```

Otázka:

Prečo sme vložili až tri mená škôl?

Teraz vložíme mená tried pre jednotlivé školy (a hneď ich aj vypíšeme):

```
//Zadaj mena tried pre kazdu skolu (bude ich 2x2)
```

```
string sTrieda[3][3];
for(int k = 1; k <= 2; k++)
    for(int j = 1; j <= 2; j++)
    {
        cout << "Zadaj meno triedy c. " << j << " zo skoly " <<
        sSkola[k] << endl;
        cin >> sTrieda[j][k];
    }
//Vypis mena tried pre kazdu skolu (bude ich 2x2)
for(int k = 1; k <= 2; k++)
    for(int j = 1; j <= 2; j++)
    {
        cout << "Trieda c. " << j << " zo skoly " << sSkola[k] << "
        je " << sTrieda[j][k] << endl;
    }
```

Znovu pre šetrenie času zadáme mená tried pri inicializácii:

```
//Vypis mena tried pre kazdu skolu (bude ich 2x2)
string sTrieda[3][3] = {{ "skola0", "skola0", "skola0"},
                        { "NultaTrieda", "4.A", "2.E"},
                        { "NultaTrieda", "Sx.A", "Ok.B"} };

for(int k = 1; k <= 2; k++)
    for(int j = 1; j <= 2; j++)
    {
        cout << "Trieda c. " << j << " zo skoly " << sSkola[k] << "
        je " << sTrieda[j][k] << endl;
    }
```

A máme prvý problém:

```
Skola c. 1 je Orange
Skola c. 2 je Javorka
Trieda c. 1 zo skoly Orange je 4.A
Trieda c. 2 zo skoly Orange je Sx.A
Trieda c. 1 zo skoly Javorka je 2.E
Trieda c. 2 zo skoly Javorka je Ok.B
Zadanim celeho cisla ukonci program
```

Vidíme, že poradie indexov pri inicializácii musí byť dodržané, potom ale poradie indexov pri výpise musíme obrátiť:

```
//Vypis mena tried pre kazdu skolu (bude ich 2x2)
string sTrieda[3][3] = {{ "skola0", "skola0", "skola0"},
                        { "NultaTrieda", "4.A", "2.E"},
                        { "NultaTrieda", "Sx.A", "Ok.B"} };

for(int k = 1; k <= 2; k++)
    for(int j = 1; j <= 2; j++)
    {
        cout << "Trieda c. " << j << " zo skoly " << sSkola[k] << "
        je " << sTrieda[k][j] << endl;
    }
```

Ak sa chceme bližšie pozrieť na to aké je poradie a vnorenie prvkov poľa najlepšie je program „debugovať“, teda zastaviť ho pomocou **breakpointu** na riadok výpisu triedy, program spustiť. Po zastavení sa programu na danom riadku do okna **Watch** pridáme

premenné **j**, **k**, **sTrieda[k][j]** a pomocou klávesy **F10** pozorujeme ako sa jednotlivé premenné menia. **Doporučujem**.

Ale najlepšou pomocou pri takýchto problémoch je vhodná voľba mien premenných. Jednotlivé indexy budeme označovať **iSkola** a **iTrieda** namiesto nič nevraviacich **j** a **k**. (Každý vidí, že sa jedná o index – predpona i):

```
for(int iSkola = 1; iSkola <= 2; iSkola++)
    for(int iTrieda = 1; iTrieda <= 2; iTrieda++)
    {
        cout << "Trieda c. " << iTrieda << " zo skoly " <<
            sSkola[iSkola] << " je " << sTrieda[iSkola][iTrieda] <<
            endl;
    }
```



A kód je hneď prehľadnejší.

Teraz ostáva zadať mená žiakov: (Je ich 2x2x2).

```
//Zadaj mena ziaikov z tried pre kazdu skolu (bude ich 2x2x2)
string sZiak[3][3][3];
for(int iSkola = 1; iSkola <= 2; iSkola++)
    for(int iTrieda = 1; iTrieda <= 2; iTrieda++)
        for(int iZiak = 1; iZiak <= 2; iZiak++)
        {
            cout << "Zadaj meno ziaika c. " << iZiak << " zo skoly "
                << sSkola[iSkola] << " z triedy " <<
                sTrieda[iSkola][iTrieda] << endl;
            cin >> sZiak[iSkola][iTrieda][iZiak];
        }
//Vypis mena tried pre kazdu skolu (bude ich 2x2)
for(int iSkola = 1; iSkola <= 2; iSkola++)
    for(int iTrieda = 1; iTrieda <= 2; iTrieda++)
        for(int iZiak = 1; iZiak <= 2; iZiak++)
        {
            cout << "Ziak c. " << iZiak << " zo skoly " <<
                sSkola[iSkola] << " z triedy " <<
                sTrieda[iSkola][iTrieda] << " je " <<
                sZiak[iSkola][iTrieda][iZiak] << endl;
        }
```

Znovu pre šetrenie času zadáme mená žiakov pri inicializácii:

```
//Vypis mena ziaikov z tried pre kazdu skolu (bude ich 2x2x2)
string sZiak[3][3][3] = {{{"0", "0", "0"},
                            {"0", "0", "0"},
                            {"0", "0", "0"},
                            {"0", "Katka", "Brano"},
                            {"0", "Oliver", "Biba"}},
                           {"0", "0", "0"},
                           {"0", "0", "0"},
                           {"0", "Samo", "Denisa"},
                           {"0", "Alica", "Lubo"}}};
for(int iSkola = 1; iSkola <= 2; iSkola++)
    for(int iTrieda = 1; iTrieda <= 2; iTrieda++)
        for(int iZiak = 1; iZiak <= 2; iZiak++)
        {
```

```
        cout << "Ziak c. " << iZiak << " zo skoly " <<
        sSkola[iSkola] << " z triedy " <<
        sTrieda[iSkola][iTrieda] << " je " <<
        sZiak[iSkola][iTrieda][iZiak] << endl;
    }
```

Opäť sa zamyslite, prečo sme zadávali reťazce „0“ a kde.

Nakoniec nám ostáva zadať známky pre každého žiaka:

```
//Zadaj znamky pre kazdeho ziaka (bude ich 2x2x2)
int nZnamka[3][3][3];
for(int iSkola = 1; iSkola <= 2; iSkola++)
    for(int iTrieda = 1; iTrieda <= 2; iTrieda++)
        for(int iZiak = 1; iZiak <= 2; iZiak++)
        {
            cout << "Zadaj znamku pre " <<
            sZiak[iSkola][iTrieda][iZiak] << " zo skoly " <<
            sSkola[iSkola] << " z triedy " <<
            sTrieda[iSkola][iTrieda] << endl;
            cin >> nZnamka[iSkola][iTrieda][iZiak];
        }
//Vypis mena ziakov z tried pre kazdu skolu (bude ich 2x2x2)
for(int iSkola = 1; iSkola <= 2; iSkola++)
    for(int iTrieda = 1; iTrieda <= 2; iTrieda++)
        for(int iZiak = 1; iZiak <= 2; iZiak++)
        {
            cout << "Znamka ziaka " <<
            sZiak[iSkola][iTrieda][iZiak] << " zo skoly " <<
            sSkola[iSkola] << " z triedy " <<
            sTrieda[iSkola][iTrieda] << " je " <<
            nZnamka[iSkola][iTrieda][iZiak] << endl;
        }
```

Znovu pre šetrenie času zadáme známky pri inicializácii:

```
int nZnamka[3][3][3] = {{{0, 0, 0},
                           {0, 0, 0},
                           {0, 0, 0}},
                           {{0, 0, 0},
                           {0, 1, 2},
                           {0, 2, 3}},
                           {{0, 0, 0},
                           {0, 3, 4},
                           {0, 4, 5}}};
```

Teraz máme zadávanie dát kompletne hotové. Môžeme začať tieto údaje spracovávať. Napr. chceme priemernú známku žiakov zo školy Orange. Musíme:

1. Spočítať všetky známky žiakov školy s indexom **iSkola = 1**. Najprv musíme:
 - a. Deklarovať novú premennú, do ktorej budeme pričítovať jednotlivé známky a inicializovať ju na nulu.
 - b. Pomocou dvoch vnorených cyklov, kde sa budú meniť indexy iTrieda a iZiak, pričom index iSkola ostane konštantný.
 - c. Nakoniec túto sumu vydělíme počtom žiakov v škole – 4:

```
//priemer ziakov v skole Orange:
int nSuma = 0;
for(int iTrieda = 1; iTrieda <= 2; iTrieda++)
    for(int iZiak = 1; iZiak <= 2; iZiak++)
    {
        nSuma += nZnamka[1][iTrieda][iZiak];
    }
cout << "Priemer znamok ziakov v skole " << sSkola[1] << " je " <<
nSuma/4.0 << endl;
```

Toto je ale dosť primitívne. Čo takto zadať pre ktorú školu chcem priemer a spočítať ho pre ľubovoľný počet žiakov v škole?:

Najprv vyzveme na zadanie mena školy a potom nájdeme aký index tejto školy prináleží:

```
//elegantnejšie - najprv zistíme ktoremu indexu odpoveda dane meno školy:
//zadame meno:
string sHladaneMeno = " ";
cout << "Zadanj meno školy " << endl;
cin >> sHladaneMeno;
int iHladanyIndex = 0;
for(int iSkola = 1; iSkola <= 2; iSkola++)
    if(sHladaneMeno == sSkola[iSkola])
        iHladanyIndex = iSkola; //Nasiel som index, pre ktorý sa
        //zadane meno zhoduje s hodnotou v sSkola[iSkola]
        //teraz už poznám index zadanej školy a môžem počítať priemer.
        //zavediem si ešte jednu premennú, kde spočítam počet žiakov v
        //škole (program bude univerzálny počet tried, žiakov
        //priemer žiakov v škole zadanej škole:
double dSuma = 0;
int nPocetZiakov = 0;
for(int iTrieda = 1; iTrieda <= 2; iTrieda++)
    for(int iZiak = 1; iZiak <= 2; iZiak++)
    {
        dSuma += nZnamka[iHladanyIndex][iTrieda][iZiak]; //spocitavam
        //znamky v škole s indexom iHladanyIndex
        nPocetZiakov++; //spocitavam žiakov
    }
cout << "Priemer znamok žiakov v škole " << sSkola[iHladanyIndex] << " je
" << dSuma/nPocetZiakov << endl;
```

Teraz môžeme zadať niekoľko úloh:

Úlohy:

1. Vypočítajte priemer známok zadanej triedy v zadanej škole.
2. Vymenujte mená žiakov v zadanej škole.
3. Vymenujte mená žiakov zadanej triedy v zadanej škole.
4. Rozšírte úlohu tak, že navyše pre každého žiaka zadáte známky aspoň z dvoch predmetov s menami „Fyzika“, „Informatika“

Tu si môžete vyskúšať množstvo nových úloh (napr. Aký je priemer z Informatiky na Orange?)