

Struktūrų masyvas

Struktūra

Struktūra – tai vienodo arba skirtingo tipo kintamųjų rinkinys.

Struktūros laikomos vartotojo sukurtu duomenų tipu.

Struktūrai aprašyti naudojama tokia sintaksė:

```
struct  struktūros_vardas  
  {  
      lauko_tipas1 lauko_vardas1;  
      lauko_tipas2 lauko_vardas2;  
      . . . . .;  
  };
```

Pavyzdžiui, turime pirkinių sąrašą. Prekės pavadinimui, kiekiui, vieneto kainai ir pinigų sumai galime sudaryti tokią struktūrą

```
struct preke
{
string pav;           //pavadinimas
int kiekis;          //prekių kiekis
double kaina;        //prekės vnt. kaina
double suma;         //pinigų suma
} ;
```

Programose struktūros tipo kintamieji aprašomi naudojant sukurto tipo vardą.

```
preke C, D;
```

Vidinio struktūros elemento vardas susideda iš struktūros vardo ir vidinio kintamojo vardo tarp kurių dedamas taškas. Pavyzdžiui:

```
C.pav C.kiekis C.kaina C.suma
```

Struktūros tipo kintamiesiems galima priskirti reikšmes:

```
C.Pav = "Duona"; C.kiekis = 2; C.kaina = 0,58;
```

Dažniausiai naudojamos ne pavienės struktūros, o jų masyvai, kurie sudaro duomenų bazių pagrindą. Struktūrų masyvai apibrėžiami analogiškai, kaip ir įprastiniai bazinių duomenų tipų masyvai.

Aprašome struktūrą:

```
struct preke  
{  
string pav;           //pavadinimas  
int kiekis;         //prekių kiekis  
double kaina;      //prekės vnt. kaina  
double suma;       //pinigų suma  
};
```

Sukuriname struktūrų masyvą:

```
preke C[100];
```

Kreipiniai į struktūros elementų laukus užrašomi:

```
C[i].pav, C[i].kiekis, C[i].kaina, C[i].suma
```

Programos pavyzdys

Klasėje yra n mokinių, iš kurių kiekvienas per pusmetį gavo po p pažymių. Parašykite programą, kuri apskaičiuotų kiekvieno mokinio pažymių aritmetinį vidurkį ir surastų, keli mokiniai yra pirmūnai. Pirmūnu laikomas mokinys, kurio pažymių vidurkis ne mažesnis už nustatytą pažymių vidurkio ribą v . Surikiuokite mokinių sąrašą vidurkio mažėjimo tvarka.

Programa turi skaityti duomenis iš tekstinio Duom2.txt failo. Pirmoje failo eilutėje yra trys skaičiai:

- mokinių skaičius n (n sveikasis skaičius, $1 \leq n \leq 100$),
- pažymių skaičius p (p sveikasis skaičius, $1 \leq p \leq 15$),
- pažymių vidurkio riba v (v realusis skaičius).

Tolesnėse n eilučių surašyti duomenys apie mokinius. Viena eilutė skirta vienam mokiniui. Pirmos 20 eilutės pozicijų skirtos mokinio vardui ir pavardei užrašyti, toliau pateikiami gauti pažymiai, atskirti vienas nuo kito tarpu.

Rezultatus programa turi įrašyti į Rez2.txt failą: kiekvienoje eilutėje turi būti užrašyti mokinio vardas ir pavardė, kuriems skiriamos pirmos 20 eilutės pozicijų, ir pažymių vidurkis vieno ženkle po kablelio tikslumu.

Paskutinėje $n + 1$ eilutėje spausdinamas pirmūnų skaičius.

Privalomi reikalavimai

- Duomenis ir rezultatus saugokite masyve (masyvuose) su struktūros tipo elementais. [?]
- Sukurkite ir panaudokite funkciją (void tipo) duomenims skaityti į masyvą su struktūros tipo elementais. [?]
- Sukurkite ir panaudokite funkciją mokinių pažymių aritmetiniam vidurkiui skaičiuoti. [?]
- Sukurkite ir panaudokite funkciją pirmūnų skaičiui rasti.
- Sukurkite ir panaudokite rikiavimo funkciją.
- Sukurkite ir panaudokite funkciją (void tipo) rezultatams įrašyti į failą.

```
Duom2.txt – užrašinė
Failas  Redagavimas  Formatavimas  Rodymas  Žinynas
4 5 8.5
Petras Petraitis      7 10 8 9 10
Morka Morkutė        8 8 9 9 6
Mikė Pukuotukas     10 9 8 7 6
Pilkas Žvirblis     9 9 9 8 8
```

```
Rez2.txt – užrašinė
Failas  Redagavimas  Formatavimas  Rodymas  Žinynas
Petras Petraitis      8.8
Pilkas Žvirblis      8.6
Mikė Pukuotukas     8.0
Morka Morkutė      8.0
2
```

```
int const Cmax = 100; //mokinių skaičius
int const Cpav = 20; //v. ir pavardės simbolių skaičius
int const Cp = 15; // mokinio pažymių skaičius
```

struct Mokinys

```
{
    string vardas; //mokinio vardas ir pavardė
    int B[Cp]; //pažymių masyvas
    int p_sk; // pažymių kiekis
    double vid ; //pažymių vidurkis
};
```

//Funkcijų prototipai

```
void Duomenys (Mokinys A[], int & n, double & v);
void Vidurkis (Mokinys A[], int n);
void Rikiavimas (Mokinys A[], int n);
int Pirmunai (Mokinys A[], int n, double v);
void Spausdinimas (Mokinys A[], int n, int pir);
```

Pagrindinė funkcija

```
int main ()
{
    setlocale(LC_ALL, "Lithuanian");
    Mokinys A[Cmax]; //Sukuriamas struktūrų masyvas
    int n,           //struktūros masyvo elementų kiekis
    pir;            // pirmūnų kiekis
    double v;       // vidurkio riba
    Duomenys (A, n, v) ;
    Vidurkis (A, n);
    Rikiavimas (A, n);
    pir = Pirmunai (A, n, v);
    Spausdinimas (A, n, pir);
    return 0;
}
```


Duomenų skaitymo funkcija

```
void Duomenys (Mokinys A[], int & n, double & v)
{
    ifstream fd ("Duom2.txt");
    char simb [Cpav + 1 ]; //sukuriamas simbolių masyvas
                                // mokinio vardui ir pavardei saugoti
    int p;                       // pažymių kiekis;
    fd >> n >> p >> v;
    fd.ignore(80, '\n');
    for (int i = 1; i <= n; i++)
    {
        fd.get (simb, sizeof simb); //skaitoma mokinio vardas
                                    // ir pavardė
        A[i].vardas = simb ;
        A[i].p_sk = p;
            for (int j = 1; j <= A[i].p_sk; j++)
                fd >> A[i].B[j]; //skaitomi pažymiai į
                                    //sveik. skaičių masyvą
        fd.ignore(80, '\n');
    }
    fd.close() ;
}
```

Vidurkio ir pirmūnų kiekio skaičiavimo funkcijos

```
void Vidurkis (Mokinys A[], int n)
{
    for ( int i = 1 ; i <= n ; i++)
    {
        int suma = 0 ;
        for (int j = 1; j <= A[i].p_sk; j++)
            suma += A[i].B[j];    //skaičiuojama pažymių suma
        A[i].vid = (double)suma / A[i].p_sk; //skaičiuojamas vid.
    }
}
```

```
int Pirmunai (Mokinys A[], int n, double v)
{
    int kiek = 0;
    for (int i = 1; i <= n ; i++)
        if (A[i].vid >= v) //kriterijus
            kiek++;    //skaičiuojamas kiekis pagal kriterijų
    return kiek;
}
```

Rikiavimo ir spausdinimo funkcijos

```
void Rikiavimas (Mokinys A[], int n)
```

```
{  
    for ( int i = 1 ; i <= n - 1 ; i++ )  
        for ( int j = i + 1 ; j <= n ; j++ )  
            if ( A[i].vid < A[j].vid )  
                swap (A[j],A[i]);  
}
```

```
void Spausdinimas (Mokinys A[], int n, int pir)
```

```
{  
    ofstream fr ("Rez2.txt");  
    for ( int i = 1; i <= n; i++)  
        fr << A[i].vardas << fixed << setprecision(1) <<  
A[i].vid << endl;  
        fr << pir;  
    fr.close () ;  
}
```