

P175B501

T9. Struktūros duomenų tipas

(4 val.)

Literatūra:

Programavimas C++: 267-285 psl.

C++ Builder: 191-200 psl.

Programavimo kalba C++: 117-124 psl.

Temos klausimai

- Alfabetinė-skaitmeninė informacija
- Struktūros prasmė, taikymas
- Struktūros aprašymas, kintamieji
- Statiniai ir dinaminiai struktūrų masyvai
- Įvedimas, išvedimas
- Veiksmai: įterpimas, šalinimas, naujo masyvo formavimas, paieška netvarkingame masyve

Struktūros duomenų tipas skirtas logiškai tarpusavyje susietiems duomenims apie **realaus pasaulio objektą** apjungti.

Objektų pavyzdžiai (objekto duomenys):

- Studentas (pavardė, vardas, asmens kodas, grupė)
- Modulis (kodas, pavadinimas, kreditų skaičius)
- Knyga (autorius, pavadinimas, leidimo metai, tiražas)
- dėstytojas (pavardė, vardas, laipsnis, katedra)

Iki dabar naudoti duomenų tipai:

- **Baziniai tipai**

`int`

`double`

`char`

`bool`

- **Masyvai**

**sveikųjų arba realiųjų skaičių
simbolių (eilutės)**

Struktūros duomenų tipo aprašas:

```
struct TipoVardas {  
    Kintamųjų (laukų) sąrašas  
};
```

Struktūros tipo aprašo pvz.:

```
struct TPirkinys {  
    char    pav[30];    // Pavadinimas  
    double  kai;       // Kaina  
    int     kie;       // Kiekis  
    double  sum;       // Pinigų suma  
};
```

Struktūros tipo kintamieji

TPrirkiny **A, B;** // Palyginkite: **int a, b;**

A				B			
pav	kai	kie	sum	pav	kai	kie	sum

Priskyrimo veiksmas: $B = A;$

Visi kintamojo **A** laukai priskiriami atitinkamiems kintamojo **B** laukams (net masyvo ir eilutės tipo laukai).

Struktūros **A** kintamieji (laukai)

A.pav, A.kai, A.kie, A.sum

Pradinių reikšmių suteikimas
struktūros **A** kintamiesiems (laukams)

```
strcpy(A.pav, "Šasiuvinis");  
A.kai = 1.50;  
A.kie = 10;  
A.sum = A.kai * A.kie;
```


Jeigu kintamojo **A** laukų
reikšmes saugo kiti kintamieji
(vardas, kaina, kiekis)

```
strcpy(A.pav, vardas);
```

```
A.kai = kaina;
```

```
A.kie = kiekis;
```

Struktūros tipo kintamojo **A** laukų reikšmių spausdinimas faile

```
ofstream fr ("Rez.txt");
```

```
...
```

```
fr << A.pav << "  "  
      << A.kai << "  "  
      << A.kie << "  "  
      << A.sum << endl;
```

```
...
```

Duomenų paruošimas faile struktūros tipo kintamiesiems

- **Duomenys surašyti atskirose eilutėse**
- **Pozicinis duomenų surašymas**

Pozicinio duomenų surašymo faile pvz.:

```
Sąsiuvinis langeliais storas    1.50    10
Rašiklis mėlynas                3.85     3
```

←----- 29 simboliai ----->_skaičius_skaičius

Pozicinis duomenų skaitymas iš failo

```
ifstream fd("Duom.txt");  
  
fd.get(A.pav, sizeof A.pav);  
fd >> A.kai >> A.kie;  
fd.ignore(); // Praleidžiamas naujos eilutės ('\n') požymis  
fd.get(B.pav, sizeof B.pav);  
fd >> B.kai >> B.kie;  
fd.ignore(); // Praleidžiamas naujos eilutės ('\n') požymis  
...
```

Struktūros tipo rodyklės

```
TPirkinys *P;
```

```
TPirkinys A;
```

```
Galimi veiksmai: P = &A;
```

```
A = *P;
```

Kreipiniai į struktūros laukus:

```
(*P).pav, (*P).kai, (*P).kie, (*P).sum
```

Patogesnė užrašymo forma:

```
P->pav, P->kai, P->kie, P->sum
```

Atminties skyrimas ir gražinimas struktūros tipo rodyklėms

```
TPirkinys *P;  
P = new TPirkinys;  
...  
// Veiksmai su dinaminio struktūros tipo kintamuoju *P  
...  
delete P;
```

Statinis struktūrų masyvas

Masyvo aprašas:

```
// const int CMax = 100;
```

```
TPirkinys A[CMax];
```

```
int n;
```

Kreipiniai į masyvo elementus:

```
A[0], A[99], A[i], A[k+2]
```

Masyvo laukų vardai:

```
A[0].pav, A[i].kie, A[k+2].pav
```

Dinaminis struktūrų masyvas

Masyvo aprašas:

```
TPirkinys *A;  
int n;
```

Kreipiniai į masyvo elementus:

```
* (A+0) , * (A+49) , * (A+i) , * (A+k+2)  
A[0] , A[49] , A[i] , A[k+2]
```

Masyvo laukų vardai:

```
(A+5) ->pav , (A+i) ->kie , (A+k+r) ->pav  
A[5] .pav , A[i] .kie , A[k+r] .pav
```

Žalia spalva parodyta patogesnė užrašymo forma

Atminties skyrimas ir grąžinimas dinaminiam struktūrų masyvui

```
TPirkinys *A;           // Dinaminis struktūrų masyvas ir
int n;                  // jo ilgis
...
// Masyvo ilgio n skaičiavimas
...
A = new TPirkinys [n]; // Atminties skyrimas
...
// Veiksmai su dinaminio struktūrų masyvu A
...
delete [] A;           // Atminties grąžinimas
```

Algoritmai

- Įvedimas
- Išvedimas (spausdinimas)
- Papildymas
- Įterpimas
- Šalinimas
- Paieška
- Rikiavimas
- Duomenų sujungimas
- Naujų sąrašų formavimas

Pavyzdys 1

Užduotis

Duomenų faile yra surašyti studento įsigyti pirkiniai:

Pirkinio pavadinimas, kaina, kiekis

Skaityti duomenis iš tekstinio failo į struktūrų masyvą ir juos spausdinti lentele rezultatų faile.
Rasti, kiek litų kainavo studento pirkiniai ?

Pavyzdys 1

Duomenų failo pavyzdys

Žemaitiška duona	3.5	15
Suvalkietiškas sūris	15.2	12
Aukštaitiškas medus	5.2	125

29 pozicijos



Pavyzdys 1

Struktūros tipas

```
struct TPirkinys {  
    char pav[30];    // Pavadinimas  
    double kai;     // Kaina  
    int kie;        // Kiekis  
    double sum;     // Pinigų suma  
};
```

Pavyzdys 1

(4)

TForm1 klasė

```
const int Cn = 100;      // Maksimalus pirkinių skaičius

class TForm1 {
  __published:
    TButton *Button1;
    void __fastcall Button1Click(TObject *Sender);
private:
    TPirkinyys A[Cn];    // Struktūrų masyvas ir
    int        n;       // jo ilgis
    void IvestiDuomenis(char fv[]);
    void Spausdinti(char fv[]);
    double BendraSuma();
};
```


Duomenų įvedimo iš failo metodas

```
void TForm1::IvestiDuomenis(char fv[])
{
    ifstream fd(fv);
    n = 0;
    while (!fd.eof() && n < Cn) {
        fd.get(A[n].pav, sizeof A[n].pav);
        fd >> A[n].kai >> A[n].kie;
        A[n].sum = A[n].kai * A[n].kie;
        fd.ignore();
        n++;
    }
    fd.close();
}
```

Pavyzdys 1

Sumos skaičiavimo metodas

```
double TForm1::BendraSuma ()
{
    double suma = 0;
    for (int i = 0; i < n; i++)
        suma += A[i].sum;
    return suma;
}
```


Masyvo spausdinimo faile metodas

```
void TForm1:: Spausdinti(char fv[])
{
    ofstream fr(fv);
    fr << "Pavadinimas           Kaina  Kiekis   Suma  \n";
    fr << "-----\n";
    for (int i = 0; i < n; i++)
        fr << A[i].pav
            << setw(5) << A[i].kai << " "
            << setw(5) << A[i].kie << " "
            << setw(7) << A[i].sum << endl;
    fr << "-----\n";
    fr << "Viso:           " << setw(7) << BendraSuma() << endl;
    fr.close();
}
```

Pavyzdys 1

Rezultatų failo pavyzdys

Pavadinimas	Kaina	Kiekis	Suma
Žemaitiška duona	3.5	15	52.5
Suvalkietiškas sūris	15.2	12	182.4
Aukštaitiškas medus	5.2	125	650
Viso:	884.9		

Pavyzdys 1

Mygtuko paspaudimo metodas

```
void __fastcall TForm1::Button1Click(TObject
    *Sender)
{
    strcpy(DFailas, DuomenuFailas().c_str());
    IvestiDuomenis(DFailas);
    strcpy(RFailas, RezultatuFailas().c_str());
    Spausdinti(RFailas);
    Mem1->Lines->LoadFromFile(RFailas);
}
```

Pastaba: kintamieji DFailas ir RFailas aprašyti klasėje TForm1

```
char DFailas[100]; // Duomenų failo vardas
char RFailas[100]; // Rezultatų failo vardas
```

Pavyzdys 1

Duomenų failo išrinkimo metodas

```
AnsiString  TForm1::DuomenuFailas ()
{
    OpenFileDialog->Filter="Tekstiniai failai (*.txt)|*.txt";
    if (OpenDialog1->Execute() &&
        FileExists (OpenDialog1->FileName))
        return OpenFileDialog->FileName;
    else return "";           // Gražina tuščią eilutę
}
```

Pavyzdys 1

Rezultatų failo išrinkimo metodas

```
Ansistring   TForm1::RezultatuFailas()  
{  
    SaveDialog1->Filter="Tekstiniai failai (*.txt)|*.txt";  
    if (SaveDialog1->Execute())  
        return SaveDialog1->FileName;  
    else return "";           // Gražina tuščią eilutę  
}
```

Masyvo papildymas nauju elementu

- **Masyvo pabaigoje**

A[n] = naujas;

n = n+1;

kur kintamasis **naujas** yra to paties tipo, kaip ir struktūrų masyvas **A**

Naujo elemento įterpimas masyve vietoje **k**

- Masyvo viduje (nerikiuotame masyve)

```
A[n] = A[k];
```

```
A[k] = naujas;
```

```
n = n + 1;
```

- Masyvo viduje (rikiuotame masyve)

```
for (int i = n; i > k; i--)
```

```
    A[i] = A[i-1];
```

```
A[k] = naujas;
```

```
n = n + 1;
```

Elemento šalinimas iš masyvo

- Paskutinio elemento

```
n = n - 1;
```

- Viduje esančio elemento (nerikiuotame masyve)

```
A[k] = A[n-1];
```

```
n = n - 1;
```

- Viduje esančio elemento (rikiuotame masyve)

```
for (int i = k; i < n-1; i++)
```

```
    A[i] = A[i+1];
```

```
n = n - 1;
```


Naujo masyvo formavimas

A[m] = naujas;

m = m + 1;

kur kintamasis **naujas** yra tokio paties tipo kaip ir struktūrų masyvas **A**

naujas gali būti:

- kito masyvo elementas, pvz. **B[j]**
- kintamasis suformuotas įvedimo iš failo metu

Pastaba: prieš pradedant formuoti naują masyvą reikia

m = 0;

Elemento paieška masyve

A(n), C.

Kuris masyvo **A** elementas pagal duotą požymį sutampa su **C**?

Galimi paieškos variantai:

- **Netvarkingame (nerikiuotame) masyve**
- **Tvarkingame (surikiuotame) masyve** (Kitoje paskaitoje)

Paieška netvarkingame masyve (1)

```
int SMasyvas::IndeksasNetvarkingame (TPirkinys C)
{
    for (int i = 0; i < n; i++)
        if (ArTinka(A[i], C))
            // Gražina surasto pirkinio C indeksą masyve A
            return i;
    // Gražina -1, jei masyve A tokio pirkinio nebuvo
    return -1;
}
```

Paieška netvarkingame masyve (2)

```
bool SMasyvas ::ArTinka (TPirkinys A, TPirkinys B)
{
    if (strcmp(A.pav, B.pav) == 0)
        // Gražina true, jei pirkinių pavadinimai sutampa
        return true;
    else
        // Gražina false, jei pirkinių pavadinimai nesutampa
        return false;
}
```

Sudėtingesni struktūrų tipai

Struktūros (kintamųjų) laukų tipai gali būti ir tokie

- **Struktūra (Pavyzdys 2)**
- **Statinis masyvas (Pavyzdys 3)**

Pavyzdys 2

Užduotis

Duomenų faile yra surašyti duomenys apie fakulteto studentus:

Pavardė vardas, gimimo data, grupė

Skaityti duomenis iš tekstinio failo į struktūrų masyvą ir juos spausdinti lentele rezultatų faile. Rasti, kiek fakultete yra nurodytos grupės studentų (**savarankiškai**)?

Pavyzdys 2

Duomenų failo pavyzdys

```

<----- 25 ----->_<- GimD ->_< Grupė >
Jonaitis      Jonas      1988 10 12 IF-7/1
Petraitis     Petras     1989 01 01 IF-7/2
Antanaitis    Antanas    1988 11 23 IF-7/1
Gerulaitis    Blogas     1989 05 05 IF-7/2
  
```

Pavyzdys 2

Struktūros tipas

```
struct Data {
    int mt,           // Metai
        mn,           // Mėnuo
        dn;           // Diena
};

struct Stud {
    char pv[26]; // Pavardė ir vardas
    Data gd;     // Gimimo data
    char grp[8]; // Grupė
};
```


TForm1 klasė

```
// Maksimalus studentų skaičius
const int CMax = 2000;
class TForm1 {
    ...
    private:
        Stud A[CMax];           // Studentų sąrašas
        int n;                   // Studentų skaičius
        void Ivesti(const char fv[]);
        void Spausdinti(const char fv[]);
        int KiekStudentu(char grupe[8]);
};
```

Duomenų įvedimo iš failo metodas

```
void TForm1::Ivesti(const char fv[])
{
    char eil[55]; // Pagalbinė eilutė
    ifstream fd(fv);
    // Perskaitoma informacinė eilutė
    fd.getline(eil, 55, '\n');
    n = 0;
    while (!fd.eof() && (n < CMax)) {
        fd.get(A[n].pv, sizeof A[n].pv);
        fd >> A[n].gd.mt >> A[n].gd.mn >> A[n].gd.dn;
        fd.ignore(); // Praleidžiamas tarpo simbolis
        fd.get(A[n].grp, sizeof A[n].grp);
        fd.ignore(); // Pereinama į naują eilutę
        n = n+1;
    }
    fd.close();
}
```

Masyvo spausdinimo faile metodas

```
void TForm1::Spausdinti(const char fv[])
{
    ofstream fr(fv, ios::app);
    fr << "          Fakulteto studentų sąrašas          \n";
    fr << "-----\n";
    fr << "Pavardė          Vardas          Gim data          Grupė \n";
    fr << "-----\n";
    for (int i = 0; i < n; i++) {
        fr << A[i].pv << " "
            << A[i].gd.mt << " "
            << setw(2) << A[i].gd.mn << " "
            << setw(2) << A[i].gd.dn << " "
            << " " << A[i].grp << endl;
    }
    fr << "-----\n";
    fr.close();
}
```

Pavyzdys 2

Rezultatų failo pavyzdys

Fakulteto studentų sąrašas

Pavardė	Vardas	Gim data	Grupė
Jonaitis	Jonas	1988 10 12	IF-7/1
Petraitis	Petras	1989 1 1	IF-7/2
Antanaitis	Antanas	1988 11 23	IF-7/1
Gerulaitis	Blogas	1989 5 5	IF-7/2

Pavyzdys 3

Užduotis

Duomenų faile yra surašyti duomenys apie grupės studentus:

Pavardė, vardas, sesijos pažymiai (5)

Skaityti duomenis iš tekstinio failo į struktūrų masyvą ir juos spausdinti lentele rezultatų faile. Rasti, kiek grupėje yra skolininkų (bent vienas įvertinimas < 5) (**savarankiškai**)?

Pavyzdys 3

Duomenų failo pavyzdys

IF-7/1

```

<----- 15 ----->_<-- 10 -->_< Pažymiai >
Jonaitis          Jonas          10   8   9  10   9
Petraitis         Petras           5   4   6   7   8
Antanaitis        Antanas          10  10  10  10  10
Gerulaitis        Blogas           8   9   9   4   7
  
```

Pavyzdys 3

Struktūros tipas

```
const int CPaz = 5; // Pažymių skaičius

struct Stud {
    char pav[16]; // Pavardė
    char vrd[11]; // Vardas
    int paz[CPaz]; // Pažymiai
};
```

Pavyzdys 3

(4)

TForm1 klasė

```
// Max studentų skaičius grupėje
const int CMax = 30;
class TForm1 {
    ...
private:
    char grupe[8]; // Grupės pavadinimas
    Stud G[CMax]; // Studentų sąrašas
    int m; // Studentų skaičius
    void Ivesti(const char fv[]);
    void Spausdinti(const char fv[]);
    int KiekSkolininku();
};
```


Duomenų įvedimo iš failo metodas

```
void TForm1::Ivesti(const char fv[])
{
    char eil[50];                // Pagalbinė eilutė
    ifstream fd(fv);
    fd >> grupe;
    fd.ignore();                // Pereinama į naują eilutę
    fd.getline(eil, 50, '\n');   // Perskaitoma informacinė eilutė
    m = 0;
    while (!fd.eof() && (m < CMax)) {
        fd.get(G[m].pav, sizeof G[m].pav);
        fd.ignore();           // Praleidžiamas tarpo simbolis
        fd.get(G[m].vrd, sizeof G[m].vrd);
        for (int j = 0; j < CPaz; j++) // Skaitomi požymiai į masyvą
            fd >> G[m].paz[j];
        fd.ignore();           // Pereinama į naują eilutę
        m = m + 1;
    }
    fd.close();
}
```

Masyvo spausdinimo faile metodas

```
void TForm1::Spausdinti(const char fv[])
{
    ofstream fr(fv, ios::app);
    fr << "    Grupės " << grupe << " studentų sąrašas" << endl;
    fr << "-----" << endl;
    fr << "Pavardė          Vardas          Pažymiai          " << endl;
    fr << "-----" << endl;
    for (int i = 0; i < m; i++) {
        fr << G[i].pav << ' ' << G[i].vrd << ' ';
        for (int j = 0; j < CPaz; j++)
            fr << setw(3) << G[i].paz[j];
        fr << endl;
    }
    fr << "-----" << endl;
    fr.close();
}
```

Pavyzdys 3

Rezultatų failo pavyzdys

Grupės IF-7/1 studentų sąrašas

Pavardė	Vardas	Pažymiai				
Jonaitis	Jonas	10	8	9	10	9
Petraitis	Petras	5	4	6	7	8
Antanaitis	Antanas	10	10	10	10	10
Gerulaitis	Blogas	8	9	9	4	7

Pavyzdys 4

S (1)

Skaitymas iki eilutės pabaigos

Faile eilutėmis surašyti studentų vieno dalyko įvertinimai. Viena eilutė skirta vienam studentui. Įvertinimų skaičius skirtingas. Rasti kiekvieno studento dalyko įvertinimų vidurkį.

9 9 8 7

6

10 10 9 8 7 8

5 6 4 4 5 8

7 9

Pavyzdys 4

S (2)

Skaitymas iki eilutės pabaigos

```
...
char sim;
double suma;
int kiek, sk;
while (!fd.eof()) {
    suma = 0;
    kiek = 0;
    sim = ' ';
    while ((sim != '\n') && !fd.eof()) {
        fd >> sk;
        sim = fd.get();
        suma += sk;
        kiek++;
    }
    fr << setw(5) << setprecision(2) << suma/kiek << endl;
}
...
```

Skaitymas iki eilutės pabaigos

```
...
double suma;
int kiek, sk;
while (!fd.eof()) {
    suma = 0;
    kiek = 0;
    while (fd.peek() != '\n' && !fd.eof()) {
        fd >> sk;
        suma += sk;
        kiek++;
    }
    fd.ignore();
    fr << setw(5) << setprecision(2) << suma/kiek << endl;
}
...
```

Sužinojote:

- kaip aprašoma struktūra;
- kaip aprašomi struktūros tipo kintamieji ir masyvai;
- kaip paruošiami duomenys struktūroms;
- kaip atliekamas struktūrų įvedimas ir išvedimas;
- kaip praktiškai naudojamos struktūros.

Struktūros duomenų tipas

Sėkmės