

## Struktūros. Struktūrų masyvai

**Pirmoji pavyzdinė programa.** Pateiktas n mokinių sąrašas: vardas, pažymys, klasė.

- Sukurkite struktūrą Mokinys, aprašančią vieną mokinį.
- Sukurkite skaitymo funkciją, kuri suformuotų 3 mokinių masyvus: visų, 11-tokų ir 12-tokų.
- Sukurkite vidurkio skaičiavimo funkciją, į kurią kreipsitės skaičiuodami visų, 11-tokų ir 12-tokų vidurkius.
- Sukurkite kiekio skaičiavimo funkciją, į kurią kreipsitės skaičiuodami, kiek yra gerai besimokančių mokinių (pažymys >7): visų, 11-tokų ir 12-tokų.
- Sukurkite rikiavimo funkciją, kuri surikiuotų mokinius pažymių gerėjimo kryptimi. Jei pažymiai sutampa, tuomet pagal vardus abėcėliškai: visų, 11-tokų ir 12-tokų.
- Sukurkite rašymo funkciją, į kurią kreipsitės išvesdami apskaičiuotas kiekvienos mokinių grupės charakteristikas.

```
#include <iomanip>
#include <cmath>
#include <fstream>
#include <iostream>
using namespace std;
//-----
struct Mokinys {
    string vardas;
    int p;
    int k;
};
//-----
void Skaitymas (Mokinys M[], int & n, Mokinys M1[], int & n1, Mokinys M2[], int & n2);
double Vidurkis (Mokinys X[], int x);
int Kiekis (Mokinys X[], int x);
void Rikiavimas (Mokinys X[], int x);
void Rasymas (string kas, Mokinys X[], int x);
//-----
int main () {
    Mokinys M[100], M1[100], M2[100];
    int n, n1, n2;
    Skaitymas (M, n, M1, n1, M2, n2);
    ofstream fr("mokiniairez.txt");
    fr.close();
    Rasymas("VISI", M, n);
    Rasymas("VIENUOLIKTOKAI", M1, n1);
    Rasymas ("DVYLIKTOKAI", M2, n2);
    return 0;
}
//-----
void Skaitymas (Mokinys M[], int & n, Mokinys M1[], int & n1, Mokinys M2[], int & n2){
    n1 = 0; n2 = 0;
    ifstream fd ("mokiniai.txt");
    fd >> n;
    for (int i = 0; i < n; i++){
        fd >> M[i].vardas >> M[i].p >> M[i].k;
        if (M[i].k == 11){
            M1[n1] = M[i];
            n1++;
        }
        else {
            M2[n2] = M[i];
            n2++;
        }
    }
    fd.close();
}
```

```

//-----
double Vidurkis (Mokinys X[], int x){
    int s = 0;
    for (int i = 0; i < x; i++){
        s = s + X[i].p;
    }
    return (double) s / x;
}
//-----
int Kiekis (Mokinys X[], int x){
    int k = 0;
    for (int i = 0; i < x; i++){
        if (X[i].p > 7) k++;
    }
    return k;
}
//-----
void Rikiavimas (Mokinys X[], int x){
    int m;
    for (int i = 0; i < x - 1; i++){
        m = i;
        for (int j = i + 1; j < x; j++){
            if (X[j].p < X[m].p || (X[j].p == X[m].p && X[j].vardas < X[m].vardas))
                m = j;
        }
        swap(X[i], X[m]);
    }
}
//-----
void Rasymas (string kas, Mokinys X[], int x)
{
    ofstream fr ("mokiniairez.txt", ios::app);
    fr << kas << endl;
    fr << "Vidurkis: " << fixed << setprecision(2) << Vidurkis(X, x) << endl;
    fr << "Kiekis: " << Kiekis(X, x) << endl;
    Rikiavimas(X, x);
    fr << "Surikiuoti " << kas << endl;
    for (int i = 0; i < x; i++)
        fr << fixed << left << setw(15) << X[i].vardas << left << setw(5) << X[i].p
            << left << setw(5) << X[i].k << endl;
    fr << "-----" << endl;
    fr.close();
}

```

Pradiniai duomenys	Rezultatai
7	VISI
Romas 5 11	Vidurkis: 7.57
Rasa 8 12	Kiekis: 4
Rita 7 12	Surikiuoti VISI
Rima 7 12	Romas 5 11
Tomas 8 11	Rima 7 12
Rytis 8 12	Rita 7 12
Dominykas 10 11	Rasa 8 12
	Rytis 8 12
	Tomas 8 11
	Dominykas 10 11
	-----
	VIENUOLIKTOKAI
	Vidurkis: 7.67
	Kiekis: 2
	Surikiuoti VIENUOLIKTOKAI
	Romas 5 11
	Tomas 8 11
	Dominykas 10 11
	-----
	DVYLIKTOKAI
	Vidurkis: 7.50
	Kiekis: 2
	Surikiuoti DVYLIKTOKAI
	Rima 7 12
	Rita 7 12
	Rasa 8 12
	Rytis 8 12
	-----

**Antroji pavyzdinė programa.** Pateiktas n mokinių sąrašas: vardas ir pavardė (20 simbolių), pažymys, klasė.

- Sukurkite struktūrą Mokinys, aprašančią vieną mokinį.
- Sukurkite skaitymo funkciją, kuri suformuotų 3 mokinių masyvus: visų, 11-tokų ir 12-tokų.
- Sukurkite vidurkio skaičiavimo funkciją, į kurią kreipsitės skaičiuodami visų, 11-tokų ir 12-tokų vidurkius.
- Sukurkite kiekio skaičiavimo funkciją, į kurią kreipsitės skaičiuodami, kiek yra gerai besimokančių mokinių (pažymys >7): visų, 11-tokų ir 12-tokų.
- Sukurkite rikiavimo funkciją, kuri surikiuotų mokinius pažymių gerėjimo kryptimi. Jei pažymiai sutampa, tuomet pagal vardus/pavardes abėcėliškai: visų, 11-tokų ir 12-tokų.
- Sukurkite rašymo funkciją, į kurią kreipsitės išvesdami apskaičiuotas kiekvienos mokinių grupės charakteristikas.

```
#include <iomanip>
#include <cmath>
#include <fstream>
#include <iostream>
using namespace std;
//-----
struct Mokinys {
    string vardaspavarde;
    int p;
    int k;
};
//-----
void Skaitymas (Mokinys M[], int & n, Mokinys M1[], int & n1, Mokinys M2[], int & n2);
double Vidurkis (Mokinys X[], int x);
int Kiekis (Mokinys X[], int x);
void Rikiavimas (Mokinys X[], int x);
void Rasymas (string kas, Mokinys X[], int x);

//-----
int main () {
    Mokinys M[100], M1[100], M2[100];
    int n, n1, n2;
    Skaitymas (M, n, M1, n1, M2, n2);
    ofstream fr("mokiniaailrez.txt");
    fr.close();
    Rasymas("VISI", M, n);
    Rasymas("VIENUOLIKTOKAI", M1, n1);
    Rasymas ("DVYLIKTOKAI", M2, n2);
    return 0;
}
//-----
void Skaitymas (Mokinys M[], int & n, Mokinys M1[], int & n1, Mokinys M2[], int & n2){
    n1 = 0; n2 = 0;
    char eil[21];
    ifstream fd ("mokiniaail.txt");
    fd >> n;
    for (int i = 0; i < n; i++){
        fd.ignore();
        fd.get(eil, 20);
        M[i].vardaspavarde = eil;
        fd >> M[i].p >> M[i].k;
        if (M[i].k == 11){
            M1[n1] = M[i];
            n1++;
        }
        else {
            M2[n2] = M[i];
            n2++;
        }
    }
    fd.close();
}
```

```

//-----
double Vidurkis (Mokinys X[], int x){
    int s = 0;
    for (int i = 0; i < x; i++){
        s = s + X[i].p;
    }
    return (double) s / x;
}
//-----
int Kiekis (Mokinys X[], int x){
    int k = 0;
    for (int i = 0; i < x; i++){
        if (X[i].p > 7) k++;
    }
    return k;
}
//-----
void Rikiavimas (Mokinys X[], int x){
    int m;
    for (int i = 0; i < x - 1; i++){
        m = i;
        for (int j = i + 1; j < x; j++){
            if (X[j].p < X[m].p || (X[j].p == X[m].p &&
                X[j].vardaspavarde < X[m].vardaspavarde)) m = j;
        }
        swap(X[i], X[m]);
    }
}
//-----
void Rasymas (string kas, Mokinys X[], int x)
{
    ofstream fr ("mokiniaiirez.txt", ios::app);
    fr << kas << endl;
    fr << "Vidurkis: " << fixed << setprecision(2) << Vidurkis(X, x) << endl;
    fr << "Kiekis: " << Kiekis(X, x) << endl;
    Rikiavimas(X, x);
    fr << "Surikiuoti " << kas << endl;
    for (int i = 0; i < x; i++)
        fr << fixed << left << setw(22) << X[i].vardaspavarde
            << left << setw(5) << X[i].p << left << setw(5) << X[i].k << endl;
    fr << "-----" << endl;
    fr.close();
}

```

Pradiniai duomenys	Rezultatai
7	VISI
Romas Romaitis 5 11	Vidurkis: 7.57
Rasa Rasaite 8 12	Kiekis: 4
Rita Ritaite 7 12	Surikiuoti VISI
Rima Rimaite 7 12	Romas Romaitis 5 11
Tomas Tomaitis 8 11	Rima Rimaite 7 12
Rytis Rytaitis 8 12	Rita Ritaite 7 12
Dominykas Domaitis 10 11	Rasa Rasaite 8 12
	Rytis Rytaitis 8 12
	Tomas Tomaitis 8 11
	Dominykas Domaitis 10 11
	-----
	VIENUOLIKTOKAI
	Vidurkis: 7.67
	Kiekis: 2
	Surikiuoti VIENUOLIKTOKAI
	Romas Romaitis 5 11
	Tomas Tomaitis 8 11
	Dominykas Domaitis 10 11
	-----
	DVYLIKTOKAI
	Vidurkis: 7.50
	Kiekis: 2
	Surikiuoti DVYLIKTOKAI
	Rima Rimaite 7 12
	Rita Ritaite 7 12
	Rasa Rasaite 8 12
	Rytis Rytaitis 8 12
	-----