

T08. Objektų konteineriai

2 ak. val.

Temos klausimai

1. Masyvas klasėje – konteineris
2. Objektų ryšiai indeksu ir reikšme (sąsajos metodai)
3. Metodų užklojimas
4. Skaičiavimų rezultatų surašymas į naują masyvą (konteinerį)



Konteinerinė klasė

Konteinerio sąvoka

Ne visada objektų savybės klasėje apibrėžiamos paprastais kintamaisiais.

Kai kada jos nusakomos **objektų** ar **reikšmių rinkiniu**.

Pavyzdžiui, tėvams gali tekti nurodyti savo vaikus (rinkinys), studentams – išklausytus ir atsiskaitytus dalykus, sesijos rezultatus ir kt. rinkinius. Tam sukuriamos ir naudojamos **konteinerinės klasės**.

Konteinerinė klasė

Klasė vadinama **konteinerine**,
jei savyje turi (aprašo)
vienatipių objektų rinkinį.

Konteinerinė klasė MasyvasInt

```
class MasyvasInt
{
    const int Cn = 100;    // masyvo dydis
    private int[] Mas;     // sveikųjų skaičių masyvas
    private int kiek;      // masyvo elementų skaičius

    public MasyvasInt()
    {
        kiek = 0;
        Mas = new int[Cn];
    }

    // grąžina i-ąjį masyvo skaičių
    public int ImtiSkaiciu(int i) { return Mas[i]; }
    // grąžina masyvo elementų skaičių
    public int ImtiKiek() { return kiek; }
}
```

Konteineriniai objektai (konteineriai)

Konteineriniai objektai kuriami taip pat, kaip ir paprasti objektai:

```
KlasėsVardas objektas = new KlasėsVardas();
```

Pavyzdžiui,

```
MasyvasInt gr1Paz = new MasyvasInt(); // 1-os gr. testas  
MasyvasInt gr2Paz = new MasyvasInt(); // 2-os gr. testas  
...
```

Objektų konteineriai

Realiose programose konteineriuose paprastai patalpinami **objektai**.

Pavyzdžiui, studijų centrui nepakanka nurodyti tik gautus testo pažymius (įvertinimus), reikia nurodyti ir studentų pavardes ir vardus.

Konteinerio elementu tampa **objektas**, kuris aprašomas kita klase (pvz., **Studentas**).

Klasė Studentas

```
class Studentas
```

```
{
```

```
    private string pavVrd;    // studento pavardė ir vardas
```

```
    private int pazym;       // pažymys (įvertinimas)
```

```
    public Studentas(string pavv, int pazym)
```

```
    {
```

```
        pavVrd = pavv;
```

```
        this.pazym = pazym;
```

```
    }
```

```
    public string ImtiPavv() { return pavVrd; }
```

```
    public int ImtiPazym()   { return pazym; }
```

```
}
```

Klasė MasyvasStudentai

```
class MasyvasStudentai
{
    const int Cn = 50;           // studentų masyvo dydis
    private Studentas [] Studentai; // studentų objektų masyvas
    private int kiek;           // studentų skaičius

    public MasyvasStudentai()
    {
        kiek = 0;
        Studentai = new Studentas[Cn];
    }

    public Studentas ImtiStudenta(int i) { return Studentai[i]; }
    public int ImtiKiek() { return kiek; }
    public void DetiStudenta(Studentas st) { Studentai[kiek++] = st;}
}
```

Klasės MasyvasStudentai konteineriai

```
MasyvasStudentai Grup1Mas = new MasyvasStudentai();  
MasyvasStudentai Grup2Mas = new MasyvasStudentai();
```

Kiekviename iš šių dviejų konteinerių gali būti saugoma iki 50 studentų įvertinimai.
Pavyzdžiui, tai gali būti dviejų studentų grupių testo įvertinimai.



Objekto išrinkimas pagal kriterijų

Objektų rinkinio charakteristikos

Apskaičiuojama objektų rinkinio charakteristika

- suma
- kiekis
- vidurkis
- sandauga (retai naudojama)

Išrenkama (pagal tam tikrą kriterijų) objektų rinkinio charakteristika

- didžiausia (mažiausia) reikšmė
- 2-a pagal didumą reikšmė
- . . .

Pavyzdys: užduotis (1/9)

Prisiminkime užduotį:

Faile "Studentai.txt" duota informacija apie vienos grupės studentų pažymius (pvz., kontrolinio darbo įvertinimai): *studento pavardė ir vardas, pažymys*.

Reikia rasti geriausią įvertinimą gavusį studentą.

Užduoties sprendimui naudosime anksčiau sudarytą klasę **Studentas** ir konteinerinę klasę **MasyvasStudentai**.

Reikia parašyti 3 **metodus** darbui su **objektų konteineriu**:

1. Duomenų skaitymui iš failo.
2. Duomenų spausdinimui rezultatų faile.
3. Geriausio studento paieškai.

Pavyzdys: klasės MasyvasStudentai konteineris (3/9)

```
const int Cn = 100;    // maksimalus studentų skaičius
static void Main(string[] args)
{
    // studentų objektų konteineris
    MasyvasStudentai TestasMas = new MasyvasStudentai();
    ...
}
```

Jonaitis Jonas;	8;
Petraitis Petras;	7;
Antanaitis Antanas;	10;
Giedraitis Giedrius;	5;
Onaitytė Ona;	8;
Juozaitis Juozas;	4;
Ramunaitė Ramunė;	5;

Pavyzdys: klasės **MasyvasStudentai** konteineris (4/9)

Konteineris MasyvasStudentai		
0	Jonaitis Jonas	8
1	Petraitis Petras	7
2	Antanaitis Antanas	10
3	Giedraitis Giedrius	5
4	Onaitytė Ona	8
5	Juozaitis Juozas	4
6	Ramunaitė Ramunė	5

Pavyzdys: metodas duomenų skaitymui (5/9)

```
static void SkaitytiStudKont(string fv, MasyvasStudentai StudentaiKont)
{
    using (StreamReader srautas = new StreamReader(fv))
    {
        string eilute; // visa duomenų failo eilutė
        while ((eilute = srautas.ReadLine()) != null)
        {
            string[] eilDalis = eilute.Split(';'); // eilutės dalims
            string pavVrd = eilDalis[0];
            int pazym = int.Parse(eilDalis[1]);
            Studentas studentas = new Studentas(pavVrd, pazym);
            StudentaiKont.DetiStudenta(studentas);
        }
    }
}
```

Pavyzdys: metodus duomenų spausdinimui (6/9)

```
static void SpausdintiStudKont(string fv, MasyvasStudentai StudentaiKont,
                                string antraste)
{
    const string virsus =
        "-----\n"
        + " Nr.   Pavardė ir vardas      Pažymys \n"
        + "-----";
    using (var fr = File.AppendText(fv)) // arba File.CreateText(fv)
    {
        fr.WriteLine("\n" + antraste);
        fr.WriteLine(virsus);
        for (int i = 0; i < StudentaiKont.ImtiKiek(); i++)
        {
            Studentas stud = StudentaiKont.ImtiStudenta(i);
            fr.WriteLine("{0, 3}   {1, -20}   {2, 2}",
                          i + 1, stud.ImtiPavv(), stud.ImtiPazym());
        }
        fr.WriteLine("-----\n");
    }
}
```

Pavyzdys: metodas geriausio studento paieškai (7/9)

```
static Studentas MaxStudentasKont(MasyvasStudentai StudentaiKont)
{
    Studentas max = StudentaiKont.ImtiStudenta(0);
    for (int i = 1; i < StudentaiKont.ImtiKiek(); i++)
    {
        Studentas stud = StudentaiKont.ImtiStudenta(i);
        if (stud.ImtiPazym() > max.ImtiPazym())
            max = stud;
    }
    return max;
}
```

Pavyzdys: kreipiniai į metodus (8/9)

```
MasyvasStudentai TestasMas = new MasyvasStudentai();  
...  
SkaitytiStudKont(CFd1, TestasMas);  
SpausdintiStudKont(CFr, TestasMas, "Studentų sąrašas");  
Console.WriteLine("Studentų skaičius: {0}",  
                    TestasMas.ImtiKiek());  
Studentas studMax = MaxStudentasKont(TestasMas);  
Console.WriteLine("Geriausias studentas: {0} {1}\n",  
                    studMax.ImtiPavv(), studMax.ImtiPazym());
```

Pavyzdys: rezultatai (9/9)

Duomenys faile:

```
Jonaitis Jonas;      8;
Petraitis Petras;    7;
Antanaitis Antanas; 10;
Giedraitis Giedrius; 5;
Onaitytė Ona;        8;
Juozaitis Juozas;    4;
Ramunaitė Ramunė;    5;
```

Rezultatai faile:

Studentų sąrašas		
Nr.	Pavardė ir vardas	Pažymys
1	Jonaitis Jonas	8
2	Petraitis Petras	7
3	Antanaitis Antanas	10
4	Giedraitis Giedrius	5
5	Onaitytė Ona	8
6	Juozaitis Juozas	4
7	Ramunaitė Ramunė	5

Rezultatai ekrane:

Studentų skaičius: 7

Geriausias studentas: Antanaitis Antanas 10

Demo

Demonstracinė programa MVS: Konteinerinė klasė
ir jos konteineris, duomenų skaitymas ir
spausdinimas, skaičiavimai



Susiję objektų rinkiniai konteineriuose

Sutampa objektų indeksai (1/10)

Faile "StudTestas.txt" duota informacija apie informatikos fakulteto I kurso studentų pažymius (**pvz., testo, programavimo žinių patikrinimo, įvertinimai**): studento pavardė ir vardas, pažymys.

Faile "StudIF.txt" duota studijų centro informacija apie **įstojusius į fakultetą studentus**: studento pavardė ir vardas, asmens kodas, kokią mokyklą (gimnaziją) baigė. Spausdinti studentų, kurie baigė nurodytą (klaviatūra) mokyklą (gimnaziją) testo įvertinimus.

Sutampa objektų indeksai (2/10)

StudTestas.txt

```
Jonaitis Jonas;      8;
Petraitis Petras;    7;
Antanaitis Antanas; 10;
Giedraitis Giedrius; 5;
Onaitytė Ona;        8;
Juozaitis Juozas;    4;
Ramunaitė Ramunė;    5;
```

StudIF.txt

```
Jonaitis Jonas;39601011111;Gimnazija 1;
Petraitis Petras;39602022222;Mokykla 1;
Antanaitis Antanas;39503033333;Gimnazija 2;
Giedraitis Giedrius;39704044444;Gimnazija 1;
Onaitytė Ona;49605055555;Mokykla 1;
Juozaitis Juozas;39506066666;Mokykla 3;
Ramunaitė Ramunė;49606066666;Gimnazija 2;
```

Pastaba: duomenys abiejuose failuose (gali būti ir **konteineriuose**) surašyti ta pačia tvarka, duomenų kiekis abiejuose failuose yra vienodas, duomenys teisingi.

Sutampa objektų indeksai (3/10)

Konteineris MasyvasStudentai

0	Jonaitis Jonas	8
1	Petraitis Petras	7
2	Antanaitis Antanas	10
3	Giedraitis Giedrius	5
4	Onaitytė Ona	8
5	Juozaitis Juozas	4
6	Ramunaitė Ramunė	5



Konteineris MasyvasStudentaiF

0	Jonaitis Jonas	39601011111	Gimnazija 1
1	Petraitis Petras	39602022222	Mokykla 1
2	Antanaitis Antanas	39503033333	Gimnazija 2
3	Giedraitis Giedrius	39704044444	Gimnazija 1
4	Onaitytė Ona	49605055555	Mokykla 1
5	Juozaitis Juozas	39506066666	Mokykla 3
6	Ramunaitė Ramunė	49606067777	Gimnazija 2

Sutampa objektų indeksai (4/10)

```
class Studentas
```

```
{  
    private string pavVrd;    // studento pavardė ir vardas  
    private int pazym;       // pažymys (įvertinimas)  
    ...                      // konstruktorius ir sąsajos metodai  
}
```

```
//-----  
// Konteinerinė klasė
```

```
class MasyvasStudentai
```

```
{  
    const int Cn = 500;      // studentų masyvo dydis  
    private Studentas [] studentai; // studentų masyvas  
    private int kiek;        // studentų skaičius  
    ...                      // konstruktorius ir sąsajos metodai  
}
```

Sutampa objektų indeksai (5/10)

```
class StudentasF
```

```
{
```

```
    private string pavVrd;    // studento pavardė ir vardas
```

```
    private long asmKodas;    // studento asmens kodas
```

```
    private string mokykla;    // kurią mokyklą baigė
```

```
    public StudentasF(string pavv, long ak, string mok)
```

```
    {
```

```
        pavVrd = pavv; asmKodas = ak;    mokykla = mok;
```

```
    }
```

```
    public string ImtiPavv()    { return pavVrd;    }
```

```
    public long ImtiAsmKoda()    { return asmKodas; }
```

```
    public string ImtiMokykla() { return mokykla; }
```

```
}
```

Pasirinktas tipas **long**,
nes **int** tipo nepakanka.

Sutampa objektų indeksai (6/10)

```
// Konteinerinė klasė
```

```
class MasyvasStudentaiF
```

```
{
```

```
    const int Cn = 500;           // studentų masyvo dydis
```

```
    private StudentasF[] Studentai; // studentų objektų masyvas
```

```
    private int kiek;             // studentų skaičius
```

```
public MasyvasStudentaiF()
```

```
{
```

```
    kiek = 0;
```

```
    Studentai = new StudentasF[Cn];
```

```
}
```

```
public StudentasF ImtiStudenta(int i) { return Studentai[i]; }
```

```
public int ImtiKiek() { return kiek; }
```

```
public void DetiStudenta(StudentasF st)
```

```
    { Studentai[kiek++] = st; }
```

```
}
```

Sutampa objektų indeksai (7/10)

```
static void Formuoti(MasyvasStudentai StudentaiKont,
                    MasyvasStudentaiF StudentaiFKont,
                    string mokykla, MasyvasStudentai StudentaiKontA)
{
    for (int i = 0; i < StudentaiKont.ImtiKiek(); i++)
    {
        StudentasF studF = StudentaiFKont.ImtiStudenta(i);
        Studentas stud = StudentaiKont.ImtiStudenta(i);
        int poz = String.Compare(studF.ImtiMokykla(), mokykla,
                                StringComparison.CurrentCulture);
        if (poz == 0)
            StudentaiKontA.DetiStudenta(stud);
    }
}
```

Sutampa objektų indeksai (8/10)

```
static void Main(string[] args)
{
    MasyvasStudentai TestasMas = new MasyvasStudentai();
    SkaitytiStudKont(CFd1, TestasMas);
    SpausdintiStudKont(CFr, TestasMas, "Studentų sąrašas (Testas)");

    MasyvasStudentaiF FakultMas = new MasyvasStudentaiF();
    SkaitytiStudFKont(CFdF, FakultMas);
    SpausdintiStudFKont(CFr, FakultMas, "Studentų sąrašas (fakulteto)");

    Console.WriteLine("Užrašykite mokyklos (gimnazijos) pavadinimą");
    string mokykla = Console.ReadLine();
    . . .
}
```

Pastaba: metodai **SkaitytiStudFKont(...)** ir **SpausdintiStudFKont(...)** nėra pateikti, jie yra yra analogiški metodams **SkaitytiStudKont(...)** ir **SpausdintiStudKont(...)**. Juos sukurkite savarankiškai.

Sutampa objektų indeksai (9/10)

```
static void Main(string[] args)
{
    . . .
    MasyvasStudentai TestasMas1 = new MasyvasStudentai();
    Formuoti(TestasMas, FakultMas, mokykla, TestasMas1);
    if (TestasMas1.ImtiKiek() > 0)
        SpausdintiStudKont(CFr, TestasMas1, "Studentų sąrašas: "
                                + mokykla);
    else
        Console.WriteLine("Studentų iš nurodytos mokyklos {0} nėra!",
                            mokykla);
    . . .
}
```

Sutampa objektų indeksai (10/10)

Įvykdžius programą ir klaviatūra įvedus: **Gimnazija 1** rezultatų faile rasite atspausdintus ne tik pradinius duomenis, bet ir sudarytą nurodytos mokyklos studentų sąrašą:

...

Studentų sąrašas: Gimnazija 1

Nr.	Pavardė ir vardas	Pažymys

1	Jonaitis Jonas	8
2	Giedraitis Giedrius	5

Demo

Demonstracinė programa MVS: Dvi konteinerinės klasės ir jų konteineriai, duomenų skaitymas ir spausdinimas, konteinerių susiejimas indeksu (sutampa objektų indeksai).

Objektas saugo kito objekto indeksą (1/11)

Prisiminkime ankstesnio uždavinio sąlygą.

Faile "Studentai.txt" duota informacija apie vienos grupės studentų pažymius (pvz., kontrolinio darbo įvertinimai): studento pavardė ir vardas, pažymys.

Faile "Pazymiai.txt" duota informacija apie žinių vertinimo sistemą: pažymys, pažymio reikšmė.

Reikia kiekvieno studento gautą įvertinimą užrašyti žodžiais.

Reikia paskaičiuoti, kiek kokių pažymių gavo studentai ? (taip buvo anksčiau)

Tam tikslui žinių vertinimo sistemos duomenis surašykime atvirkčia tvarka. Tada pradžioje bus blogiausias, pabaigoje geriausias pažymys. Parašykime masyve esančių duomenų apgręžimo metodą (to nedarant galima pakeisti failo duomenis).

Studentai.txt

```
Jonaitis Jonas; 8;  
Petraitis Petras; 7;  
...  
Ramunaitė Ramunė; 5;
```

Pazymiai.txt

```
10;Puikiai;  
9;Labai gerai;  
...  
2;Nepatenkinamai;  
1;Nepatenkinamai;
```

Objektas saugo kito objekto indeksą (2/11)

```
class Studentas
```

```
{  
    private string pavVrd;    // studento pavardė ir vardas  
    private int pazym;       // pažymys (įvertinimas)  
    ...                      // konstruktorius ir sąsajos metodai  
}
```

```
//-----
```

```
class MasyvasStudentas
```

```
{  
    const int Cn = 500;       // studentų masyvo dydis  
    private Studentas [] Studentai; // studentų masyvas  
    private int kiek;         // studentų skaičius  
    ...                      // konstruktorius ir sąsajos metodai  
}
```

Objektas saugo kito objekto indeksą (3/11)

```
class MasyvasPazymiai
{
    const int Cn = 10;           // pažymių masyvo dydis
    private Pazymys[] Pazymiai; // pažymių objektų nuoroda
    private int kiek;           // pažymių skaičius

    public MasyvasPazymiai()
    {
        kiek = Cn;
        Pazymiai = new Pazymys[Cn] // pažymių objektų masyvas
        {
            new Pazymys(10, "Puikiai"),
            new Pazymys(9, "Labai gerai"),
            ...
            new Pazymys(1, "Nepatenkinamai")
        };
    }
    // sąsajos metodai ir metodas Apsukti()
}
```

Objektas saugo kito objekto indeksą (4/11)

// Sąsajos metodai

```
public Pazymys ImtiPazymi(int i) { return Pazymiai[i]; }
```

```
public int ImtiKiek() { return kiek; }
```

```
public void DetiPazymi(Pazymys obj, int i){ Pazymiai[i]= obj;}
```

// Apsuka masyvą Pazymiai(kiek) atvirkščia tvarka

```
public void Apsukti()
```

```
{
```

```
    for (int i = 0; i < kiek / 2; i++)
```

```
    {
```

```
        Pazymys p = Pazymiai[i];
```

```
        Pazymiai[i] = Pazymiai[kiek - i - 1];
```

```
        Pazymiai[kiek - i - 1] = p;
```

```
    }
```

```
}
```

Objektas saugo kito objekto indeksą (5/11)

```
static void Main(string[] args)
{
    MasyvasStudentai TestasMas = new MasyvasStudentai();
    SkaitytiStudKont(CFd1, TestasMas);
    SpausdintiStudKont(CFr, TestasMas, "Studentų sąrašas (kont.)");

    MasyvasPazymiai PazymMas = new MasyvasPazymiai();
    PazymMas.Apsukti(); // apskama, kad galima būtų kreiptis pagal indeksą
    RodytiPazVertesKont(TestasMas, PazymMas);
    PapildytiPazMasyvaKont(TestasMas, PazymMas);
    PazymMas.Apsukti(); // galima apsukti masyvą atgal
    SpausdintiPazymKont(CFr, PazymMas, "Žinių vertinimo sistema");
    ...
}
```


Objektas saugo kito objekto indeksą (6/11)

Duomenys surašyti atvirkščia tvarka.

Studentai.txt

Jonaitis Jonas;	8;
Petraitis Petras;	7;
Antanaitis Antanas;	10;
Giedraitis Giedrius;	5;
Onaitytė Ona;	8;
Juozaitis Juozas;	4;
Ramunaitė Ramunė;	5;

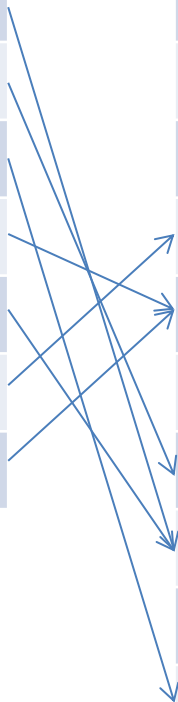
Pazymiai(Cn)

1	Nepatenkinamai
2	Nepatenkinamai
3	Nepatenkinamai
4	Nepatenkinamai
5	Silpnai
6	Patenkinamai
7	Vidutiniškai
8	Gera
9	Labai gerai
10	Puikiai

Objektas saugo kito objekto indeksą (7/11)

Konteineris MasyvasStudentai		
0	Jonaitis Jonas	8
1	Petraitis Petras	7
2	Antanaitis Antanas	10
3	Giedraitis Giedrius	5
4	Onaitytė Ona	8
5	Juozaitis Juozas	4
6	Ramunaitė Ramunė	5

Konteineris MasyvasPazymiai			
0	1	Nepatenkinamai	0
1	2	Nepatenkinamai	0
2	3	Nepatenkinamai	0
3	4	Nepatenkinamai	0
4	5	Silpnai	0
5	6	Patenkinamai	0
6	7	vidutiniškai	0
7	8	Gera	0
8	9	Labai gerai	0
9	10	Puikiai	0



Objektas saugo kito objekto indeksą (8/11)

```
static void RodytiPazVertesKont(MasyvasStudentai StudentaiKont,  
                                MasyvasPazymiai PazymKont)  
{  
    for (int i = 0; i < StudentaiKont.ImtiKiek(); i++)  
    {  
        Studentas stud = StudentaiKont.ImtiStudenta(i);  
        int pazStud = stud.ImtiPazym();  
        string pazPaZ = PazymKont.ImtiPazymi(pazStud - 1).ImtiPazReiksme();  
        Console.WriteLine("{0, 3}    {1, -20} {2, -14}",  
                            i + 1, stud.ImtiPavv(), pazPaZ);  
    }  
}
```

Objektas saugo kito objekto indeksą (9/11)

Ekrane matysite:

1	Jonaitis Jonas	Geraĩ
2	Petraitis Petras	Vidutiniškai
3	Antanaitis Antanas	Puikiai
4	Giedraitis Giedrius	Silpnai
5	Onaitytė Ona	Geraĩ
6	Juozaitis Juozas	Nepatenkinamai
7	Ramunaitė Ramunė	Silpnai

Objektas saugo kito objekto indeksą (10/11)

```
// suskaičiuoja pažymių pasikartojimų skaičius kont. StudentaiKont  
// ir juos surašo į konteinerį PazymKont
```

```
static void PapildytiPazMasyvaKont(MasyvasStudentai StudentaiKont,  
                                   MasyvasPazymiai PazymKont)
```

```
{
```

```
    for (int i = 0; i < StudentaiKont.ImtiKiek(); i++)
```

```
    {
```

```
        Studentas stud = StudentaiKont.ImtiStudenta(i);
```

```
        int pazStud = stud.ImtiPazym();
```

```
        Pazymys pazym = PazymKont.ImtiPazymi(pazStud - 1);
```

```
        int pazKiek = pazym.ImtiPazKieki();
```

```
        pazym.DetiPazKieki(pazKiek + 1);
```

```
        PazymKont.DetiPazymi(pazym, pazStud-1);
```

```
    }
```

```
}
```

Objektas saugo kito objekto indeksą (11/11)

Rezultatų faile matysite:

...

Žinių vertinimo sistema

Studentų sąrašas

Pažymys	Pažymio reikšmė	Kiekis

10	Puikiai	1
9	Labai gerai	0
8	Gera	2
7	Vidutiniškai	1
6	Patenkinamai	0
5	Silpnai	2
4	Nepatenkinamai	1
3	Nepatenkinamai	0
2	Nepatenkinamai	0
1	Nepatenkinamai	0

Demo

Demonstracinė programa MVS: Dvi konteinerinės klasės ir jų konteineriai, duomenų skaitymas ir spausdinimas, konteinerių susiejimas indeksu (objektas saugo kito objekto indeksą).

Faile "StudTestas.txt" duota informacija apie informatikos fakulteto I kurso studentų pažymius (pvz., testo, programavimo žinių patikrinimo, įvertinimai): studento pavardė ir vardas, pažymys.

Faile "StudIF.txt" duota informacija apie įstojusius į fakultetą studentus: studento pavardė ir vardas, asmens kodas, kokią mokyklą (gimnaziją) baigė.

Skaityti failų informaciją į **objektų konteinerius**.

Spausdinti studentų, kurie baigė nurodytą (klaviatūra) mokyklą (gimnaziją) testo įvertinimus.

StudTestas.txt

```
Jonaitis Jonas;      8;
Petraitis Petras;    7;
Antanaitis Antanas; 10;
Giedraitis Giedrius; 5;
Onaitytė Ona;        8;
Juozaitis Juozas;    4;
Ramunaitė Ramunė;    5;
```

StudIF.txt

```
Antanaitis Antanas;39503033333;Gimnazija 2;
Giedraitis Giedrius;39704044444;Gimnazija 1;
Jonaitis Jonas;39601011111;Gimnazija 1;
Juozaitis Juozas;39506066666;Mokykla 3;
Onaitytė Ona;49605055555;Mokykla 1;
Petraitis Petras;39602022222;Mokykla 1;
Ramunaitė Ramunė;49606067777;Gimnazija 2;
```

Pastaba: duomenų surašymo tvarka failuose (gali būti ir konteineriuose) skiriasi. Čia daroma prielaida, kad rinkinyje nėra sutampančių (vienodų) studentų pavardžių ir vardų.

```
class Studentas
{
    private string pavVrd;    // studento pavardė ir vardas
    private int pazym;        // pažymys (įvertinimas)
    ...    // konstruktorius ir sąsajos metodai
}
//-----
class MasyvasStudentai
{
    const int Cn = 500;        // studentų masyvo dydis
    private Studentas [] Studentai;    // studentų masyvas
    private int kiek;          // studentų skaičius
    ...    // konstruktorius ir sąsajos metodai
}
```

```
class StudentasF
{
    private string pavVrd;    // studento pavardė ir vardas
    private long  asmKodas;   // studento asmens kodas
    private string mokykla;   // kurią mokyklą baigė
    ...                      // konstruktorius ir sąsajos metodai
}
//-----
class MasyvasStudentaiF
{
    const int Cn = 500;      // studentų masyvo dydis
    private StudentasF[] Studentai; // studentų objektų masyvas
    private int kiek;        // studentų skaičius
    ...                      // konstruktorius ir sąsajos metodai
}
```

Konteineris MasyvasStudentai			Konteineris MasyvasStudentai F			
0	Jonaitis Jonas	8	0	Antanaitis Antanas	39503033333	Gimnazija 2
1	Petraitis Petras	7	1	Giedraitis Giedrius	39704044444	Gimnazija 1
2	Antanaitis Antanas	10	2	Jonaitis Jonas	39601011111	Gimnazija 1
3	Giedraitis Giedrius	5	3	Juozaitis Juozas	39506066666	Mokykla 3
4	Onaitytė Ona	8	4	Onaitytė Ona	49605055555	Mokykla 1
5	Juozaitis Juozas	4	5	Petraitis Petras	39602022222	Mokykla 1
6	Ramunaitė Ramunė	5	6	Ramunaitė Ramunė	49606066666	Gimnazija 2

```
class MasyvasStudentaiF
```

```
{
```

```
...
```

```
public int StudentoVieta(string pavVrd)
```

```
{
```

```
    for (int i = 0; i < kiek; i++)
```

```
    {
```

```
        int poz = String.Compare(Studentai[i].ImtiPavv(), pavVrd,  
                                StringComparison.CurrentCulture);
```

```
        if (poz == 0) // ieškoma pavardė ir vardas masyve sutapo  
            return i;
```

```
    }
```

```
    return -1; // ieškomos pavardė ir vardo masyve nėra
```

```
}
```

```
}
```

Klasė **MasyvasStudentaiF**
papildyta studento paieškos
metodu

```
static void Formuoti1(MasyvasStudentai StudentaiKont,  
                      MasyvasStudentaiF StudentaiFKont,  
                      string mokykla,  
                      MasyvasStudentai StudentaiKontA)  
{  
    for (int i = 0; i < StudentaiKont.ImtiKiek(); i++)  
    {  
        Studentas stud = StudentaiKont.ImtiStudenta(i);  
        string pavVrd = stud.ImtiPavv();  
        int index = StudentaiFKont.StudentoVieta(pavVrd);  
        StudentasF studF = StudentaiFKont.ImtiStudenta(index);  
        int poz = String.Compare(studF.ImtiMokykla(), mokykla,  
                                StringComparison.CurrentCulture);  
        if ((index >= 0) && (poz == 0))  
            StudentaiKontA.DetiStudenta(stud);  
    }  
}
```

```
static void Main(string[] args)
{
    MasyvasStudentai TestasMas = new MasyvasStudentai();
    SkaitytiStudKont(CFd1, TestasMas);
    SpausdintiStudKont(CFr, TestasMas, "Studentų sąrašas kont.");
    MasyvasStudentaiF FakultMas = new MasyvasStudentaiF();
    SkaitytiStudFKont(CFdF, FakultMas);
    SpausdintiStudFKont(CFr, FakultMas, "Studentų sąrašas(fakult.)");
    Console.WriteLine("Užrašykite mokyklos (gimn.) pavadinimą");
    string mokykla = Console.ReadLine();
    . . .
}
```

```
static void Main(string[] args)
{
    . . .
    MasyvasStudentai TestasMas1 = new MasyvasStudentai();
    Formuoti1(TestasMas, FakultMas, mokykla, TestasMas1);
    if (TestasMas1.ImtiKiek() > 0)
        SpausdintiStudKont(CFr, TestasMas1, "Studentų sąrašas: " +
                                mokykla);
    else
        Console.WriteLine("Studentų iš mokyklos {0} nėra!",
                                mokykla);
}
```


Įvykdžius programą ir klaviatūra įvedus: **Gimnazija 1** rezultatų faile rasite atspausdintus ne tik pradinis duomenis, bet ir sudarytą nurodytos mokyklos studentų sąrašą:

...

Studentų sąrašas: Gimnazija 1

Nr.	Pavardė ir vardas	Pažymys

1	Jonaitis Jonas	8
2	Giedraitis Giedrius	5

Demo

Demonstracinė programa MVS: Dvi konteinerinės klasės ir jų konteineriai, duomenų skaitymas ir spausdinimas, konteinerių susiejimas reikšmėmis (objektas saugo kito objekto reikšmę).



Metodų užklojimas

Užklojimo apibrėžimas

- Užklojimas leidžia turėti dvi ar daugiau skirtingų to paties metodo realizacijų toje pačioje vardų galiojimo srityje (**klasėje**).
- Paprasčiau – tai galimybė panaudoti tą patį vardą dviem ar daugiau metodų.
- Kurį metodą naudoti į jį kreipiantis, kompiliatorius sprendžia pagal argumentų ir parametrų atitikimą.

Užklojimo pavyzdys (1/3)

```
class MasyvasDouble
```

```
{
```

```
    const int Cn = 100;    // masyvo dydis
```

```
    private double[] Mas;  // realiųjų skaičių masyvas
```

```
    private int kiek;      // masyvo elementų skaičius
```

```
    // Konstruktorius ir sąsajos metodai
```

```
    ...
```

```
    // Du užkloti metodai Max()
```

```
    ...
```

```
}
```

Užklojimo pavyzdys (2/3)

```
public double Max(int i1, int i2)
{
    double max = -999.9; // skaičius, mažesnis už masyvo reikšmes
    for (int i = i1; i <= i2; i++)
        if (Mas[i] > max)
            max = Mas[i];
    return max;
}

//-----
public double Max(double d1, double d2)
{
    double max = -999.9; // skaičius, mažesnis už masyvo reikšmes
    for (int i = 0; i < kiek; i++)
        if (Mas[i] >= d1 && Mas[i] <= d2 && Mas[i] > max)
            max = Mas[i];
    return max;
}
```

Užklojimo pavyzdys (3/3)

```

static void Main(string[] args)
{
    MasyvasDouble paz = new MasyvasDouble(); // pažangumas (vidur)
    Skaityti(CFd, paz);
    Console.WriteLine("Vidurkis indeksų intervale: {0}",
        paz.Max(paz.ImtiKiek() / 2, paz.ImtiKiek()-1));
    Console.WriteLine("Vidurkis reikšmių intervale: {0}",
        paz.Max(7.5, 10.0));
    ...
}
  
```

Koks bus rezultatas?

Konteineris **paz** sudarytas iš 7 elementų

5.45	9.59	8.87	4.65	9.25	7.18	8.35
0	1	2	3	4	5	6

Parametrų tipų skirtingumas

Užklojami metodai atskiriami pagal skirtingus parametrus.

Du ar daugiau užklojamų metodų:

- negali skirtis vien tik grąžinama reikšme,
- turi skirtis parametrais (jų skaičiumi, jų tipais).

Užkloto metodo išrinkimas

Kviečiant metodą imamas tas užklotas metodas, kurio *parametrai* eilės tvarka pagal tipą sutampa su *argumentų tipais*, esančiais kreipinyje į metodą.

Kada metodai neturėtų būti užklojami?

Kai metodai, kuriuos ruošiatės užkloti, atlieka skirtingus veiksmus. Toks užklojimas paslėptų šiuos skirtumus ir teisingai naudoti metodus bus sunkiau.

Konstruktoriaus užklojamas

Kaip bet kuris kitas metodas, konstruktorius taip pat gali būti užklotas. Jų užklojimui galioja tos pačios taisyklės.

```
class Studentas
{
    private string pavVrd;    // studento pavardė ir vardas
    private int pazym;       // pažymys (įvertinimas)

    // užkloti konstruktoriai
    public Studentas(string pavv, int pazym)
    {
        pavVrd = pavv;
        this.pazym = pazym;
    }
    public Studentas(string pavv)
    {
        pavVrd = pavv;
        pazym = 0;
    }
    public Studentas()
    {
        pavVrd = "Pavardenis Vardenis";
        pazym = -1;
    }
    ... // sąsajos metodai
}
```

```
Studentas stud1 = new Studentas("Pavardaitis Vardaitis", 10);  
Console.WriteLine("{0} {1}", stud1.ImtiPavv(), stud1.ImtiPazym());  
  
Studentas stud2 = new Studentas("Pavardienis Vardienis");  
Console.WriteLine("{0} {1}", stud2.ImtiPavv(), stud2.ImtiPazym());  
  
Studentas stud3 = new Studentas();  
Console.WriteLine("{0} {1}", stud3.ImtiPavv(), stud3.ImtiPazym());
```

Ivykdžius šį programos fragmentą ekrane matysite:

```
Pavardaitis Vardaitis 10  
Pavardienis Vardienis 0  
Pavardenis Vardenis -1
```



Konteinerio užpildymas skaičiavimų rezultatais

Užduotis

Funkcijos $y = (a + x) / (b - x)$ reikšmes surašykite į konteinerį, kai argumentas x kinta žingsniu hx intervale $x_p \leq x \leq x_g$.

Pavyzdys: konteinerio užpildymas skaičiais rezultatais (2/7)

```
class Taskas
{
    private double x;
    private double y;
    public Taskas(double xx, double yy)
    {
        x = xx;
        y = yy;
    }
    public double ImtiX()
    {
        return x;
    }
    public double ImtiY()
    {
        return y;
    }
}
```


Pavyzdys: konteinerio užpildymas skaičiais rezultatais (3/7)

```
class MasyvasTaskai
{
    const int Cn = 1000;           // masyvo dydis
    private Taskas[] Taskai;       // taškų masyvas
    int kiek;                       // taškų skaičius
    public MasyvasTaskai()
    {
        kiek = 0;
        Taskai = new Taskas[Cn];
    }
    public int ImtiKiek()
    {
        return kiek;
    }
    public void DėtiTaska(Taskas tsk) { Taskai[kiek++] = tsk; }
    public Taskas ImtiTaska(int i)
    {
        return Taskai[i];
    }
}
```

Pavyzdys: konteinerio užpildymas skaičiais rezultatais (4/7)

```
static void Formuoti(MasyvasTaskai TskKont, double a, double b,  
                    double xp, double xg, double hx)  
{  
    for (double x = xp; x <= xg; x += hx)  
        if (Math.Abs(b - x) > 0.001)  
        {  
            Taskas taskas = new Taskas(x, (a + x) / (b - x));  
            TskKont.DėtiTaska(taskas);  
        }  
}
```

Pavyzdys: konteinerio užpildymas skaičiais rezultatais (5/7)

```
static void spausdinti(MasyvasTaskai fr, string koment)
{
    const string virusus =
        "-----\n"
        + "      X      Y      \n"
        + "-----";
    Console.WriteLine("\n " + koment);
    Console.WriteLine(virusus);
    for (int i = 0; i < fr.ImtiKiek(); i++)
    {
        Taskas taskas = fr.ImtiTaska(i);
        double x = taskas.ImtiX();
        double y = taskas.ImtiY();
        Console.WriteLine("{0, 7:F2}    {1,10:F5}", x, y);
    }
    Console.WriteLine("-----\n");
}
```

Pavyzdys: konteinerio užpildymas skaičiais rezultatais (6/7)

```
static void Main(string[] args)
{
    MasyvasTaskai FunkcReiksmes = new MasyvasTaskai();
    double a = 1.5;
    double b = 4.7;
    double xp = 4.1;
    double xg = 5.3;
    double hx = 0.1;
    Formuoti(FuncReiksmes, a, b, xp, xg, hx);
    Spausdinti(FuncReiksmes, "Funkcijos reikšmių lentelė");
}
```

Pavyzdys: konteinerio užpildymas skaičiavimų rezultatais (7/7)

Funkcijos reikšmių lentelė

X	Y
4,10	9,33333
4,20	11,40000
4,30	14,50000
4,40	19,66667
4,50	30,00000
4,60	61,00000
4,80	-63,00000
4,90	-32,00000
5,00	-21,66667
5,10	-16,50000
5,20	-13,40000
5,30	-11,33333

Demo

Demonstracinė programa MVS: konteinerio užpildymas skaičiavimų rezultatais ir konteineryje esančių reikšmių parodymas lentele ekrane.

Šioje temoje susipažinome su:

1. Konteinerinėmis klasėmis.
2. Susijusiais objektų rinkiniais konteineriuose.
3. Metodų užklojimu.
4. Skaičiavimo rezultatų surašymu į konteinerį.



Klausimai?