

T06. Objektų rinkinio charakteristikų radimas, paieška

2 ak. val.

Temos klausimai

1. Sumos, sandaugos, kiekio skaičiavimo algoritmai.
2. Vidurkio skaičiavimas.
3. Didžiausios reikšmės paieška.
4. Naujo masyvo formavimas.
5. Objektų ryšiai indeksu.
6. Objektų ryšiai reikšme.
7. Nuosekli paieška nerikiuotame masyve.
8. Dviejų objektų rinkinių sujungimas.



Sumos, sandaugos, kiekio skaičiavimo algoritmai

Objekto charakteristika

Objekto atributas (būsena, kintamasis).

Pavyzdžiui,

Klasės **Studentas** objektų charakteristikos:

- pavardė ir vardas

- pažymys

Klasės **Pazymys** objektų charakteristikos:

- pažymys

- pažymio žodinė reikšmė

- pažymio pasikartojimo skaičius

Objektų rinkinio charakteristikos

Apskaičiuojama objektų rinkinio charakteristika

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

Išrenkama (pagal tam tikrą kriterijų) objektų rinkinio charakteristika

- didžiausia (mažiausia) reikšmė
- dviejų didžiausių (mažiausių) reikšmė
- ...

Bus pateikta vėliau

Skaičiuojamos charakteristikos

Suma / vidurkis / sandauga

- visų reikšmių;
- reikšmių su apribojimais:
 - teigiamų (neigiamų) reikšmių
 - tam tikrame intervale esančių reikšmių

Kiekis

- teigiamų (neigiamų) reikšmių
- tam tikrame intervale esančių reikšmių

Sumos, sandaugos, kiekio, vidurkio skaičiavimo algoritmų esmė

Sumos, sandaugos, kiekio skaičiavimui taikomas kaupimo algoritmas:

prieš ciklą suteikiama pradinė reikšmė 0 (suma, kiekis) arba 1 (sandauga) kintamajam, kuriame kaupiamas rezultatas

ciklo viduje tikrinama kiekvieno elemento reikšmė, tinkama reikšmė pridedama (iš jos padauginama ar pridedamas 1);

Vidurkis skaičiuojamas:

$$\text{Vidurkis} = \text{suma} / \text{kiekis}.$$

Kiekio skaičiavimo masyve metodas (1)

```
static int KiekMazesniu(int[] Mas, int kiek)
{
    int k = 0;
    for (int i = 0; i < kiek; i++)
        if (Mas[i] < 5)
            k = k + 1;    // arba k++
    return k;
}
```

		Masyvo dydis: kiek = 7;						
Mas	----->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6

Metodą pritaikius
masyvui **Mas(kiek)**
rezultatas: **1**

Kiekio skaičiavimo masyve metodas (2)

```
static int KiekIntervale(int[] Mas, int kiek, int a, int b)
{
    int k = 0;
    for (int i = 0; i < kiek; i++)
        if ((a <= Mas[i]) && (Mas[i] <= b)) // a <= Mas[i] <= b
            k = k + 1; // arba k++
    return k;
}
```

		Masyvo dydis: kiek = 7;						
Mas	----->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6

Metodą pritaikius masyvui **Mas(kiek)** ir nurodžius a=8, o b=10 rezultatas: **3**

Sumos skaičiavimo masyve metodas

```
static int Suma(int[] Mas, int kiek)
{
    int suma = 0;
    for (int i = 0; i < kiek; i++)
        suma = suma + Mas[i];    // suma += Mas[i]
    return suma;
}
```

		Masyvo dydis: kiek = 7;						
Mas	----->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6

Metodą pritaikius
masyvui **Mas(kiek)**
rezultatas: **47**

```
static double vidurkis(int[] Mas, int kiek)
{
    double sum = 0.0;
    for (int i = 0; i < kiek; i++)
        sum += Mas[i];
    if (kiek != 0)
        return sum / kiek;
    else
        return 0.0;
}
```

Vietoje ciklo galima rašyti kreipinį į metodą **Suma(Mas, kiek)**, o po to gautą rezultatą padalinti iš kiek.

		Masyvo dydis: kiek = 7;						
Mas	----->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6

Metodą pritaikius masyvui **Mas(kiek)** rezultatas:
6,714285714

```
static double vidurkis1(int[] Mas, int kiek)
{
    double sum = 0.0;
    int k = 0;
    for (int i = 0; i < kiek; i++)
        if (Mas[i] > (Mas[0] + Mas[kiek-1])/2.0)
        {
            sum += Mas[i];
            k++;
        }
    if (k != 0)
        return sum / k;
    else
        return 0.0;
}
```

		Masyvo dydis: kiek = 7;						
Mas	----->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6

Metodą pritaikius
masyvui **Mas(kiek)**
rezultatas: **8,25**

Didžiausios reikšmės paieškos masyve metodas (1)

```
static int Didžiausias(int[] Mas, int kiek)
{
    int maxInd = 0;
    int maxSkaic = Mas[0];
    for (int i = 1; i < kiek; i++)
        if (Mas[i] > maxSkaic)
        {
            maxInd = i;
            maxSkaic = Mas[i];
        }
    return maxInd;    // arba maxSkaic
}
```

Ar gera pradinė ciklo
parametro **i**
reikšmė?

Kokius pakeitimus
reikėtų atlikti,
norint rasti
mažiausią reikšmę?

		Masyvo dydis: kiek = 7;						
Mas	----->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6

Metodą pritaikius
masyvui **Mas(kiek)**
rezultatas: **2 (10)**

Didžiausios reikšmės paieškos masyve metodas (2)

```
static int Didžiausias(int[] Mas, int kiek)
{
    int maxInd = 0;
    for (int i = 1; i < kiek; i++)
        if (Mas[i] > Mas[maxInd])
            maxInd = i;
    return maxInd;    // arba Mas[maxInd]
}
```

Kurio didžiausio
elemento
indeksą suras
metodas, jeigu
masyve bus
kelios vienodos
didžiausios
reikšmės?

		Masyvo dydis: kiek = 7;						
Mas	----->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6

Metodą pritaikius
masyvui **Mas(kiek)**
rezultatas: **2 (10)**

Naujo masyvo formavimo metodas

```
static void Formuoti(int[] Mas, int kiek, int riba,
                    int[] MasA, out int kiekA)
{
    kiekA = 0;
    for (int i = 0; i < kiek; i++)
    {
        if (Mas[i] > riba)
        {
            MasA[kiekA++] = Mas[i];
        }
    }
}
```

Čia užrašoma naujo
masyvo formavimo
sąlyga

		Masyvo dydis: kiek = 7;						
Mas	----->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6
		Masyvo dydis: kiekA = 3;						
MasA	--->	8	10	8				
		0	1	2				

Metodą pritaikius n ir nurodžius ribą lyg MasA(kiekA) bus įrašyti visi elementai nuo n iki kiekA-1.

Metodą pritaikius masyvui **Mas(kiek)**
ir nurodžius ribą lygią 7 į masyvą
MasA(kiekA) bus įrašytos 3 reikšmės.

Demo

Demonstracinė programa MVS: kiekio, sumos, sandaugos ir vidurkio skaičiavimas skaičių masyve.

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 Program

```
class Program
{
    const int Cn = 100;    // maksimalus studentų skaičius
    const string CFd = "..\\..\\Studentai.txt"; // duomenų failas
    const string CFr = "..\\..\\Rezultatai.txt"; // rezultatų failas

    static void Main(string[] args)
    {
        int kiek;          // studentų skaičius
        Studentas[] Studentai = new Studentas[Cn]; // studentų masyvas
        // Studentų duomenų įvedimas ir spausdinimas
        ...
        // Charakteristikų (kiekio, vidurkio ir t.t.) skaičiavimas
        ...
    }
}
```

Duomenų ir rezultatų pavyzdys

7

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

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

Žinių vertinimo sistema

Pažymys Pažymio reikšmė Kiekis

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

Objektų masyvo charakteristika: kiekis

```
static int KiekNeigiamu(Studentas[] studentai, int kiek)
{
    int kiekis = 0;
    for (int i = 0; i < kiek; i++)
        if (Studentai[i].ImtiPazym() < 5)
            kiekis = kiekis + 1;    // kiekis++
    return kiekis;
}
...
Console.WriteLine("Neigiamų pažymių skaičius: {0} \n",
                  KiekNeigiamu(studentai, kiek));
...
Ekране matysite: Neigiamų pažymių skaičius: 1
```

Objektų masyvo charakteristika: kiekis

```
static int kiekIntervale(Studentas[] Studentai, int kiek,
                        int a, int b)
{
    int kiekis = 0;
    for (int i = 0; i < kiek; i++)
        if ((a <= Studentai[i].ImtiPazym()) &&
            (Studentai[i].ImtiPazym() <= b))
            kiekis = kiekis + 1;
    return kiekis;
}
...
Console.WriteLine("Pažymių intervale [8;10] mokosi: {0}",
                  kiekIntervale(Studentai, kiek, 8, 10));
...
Ekrane matysite: Pažymių intervale [8;10] mokosi: 3
```

Objektų masyvo charakteristika: kiekis

```
static int KiekStudentuPr(Studentas[] Studentai,
                          int kiek, char simb)
{
    int kiekis = 0;
    for (int i = 0; i < kiek; i++)
        if (Studentai[i].ImtiPavv()[0] == simb)
            kiekis = kiekis + 1;
    return kiekis;
}
...
Console.WriteLine("Raide (J) prasideda {0} pavardės.\n",
                  KiekStudentuPr(Studentai, kiek, 'J'));
...
Ekrane matysite: Raide (J) prasideda 2 pavardės.
```

Pirma pavardės raidė

Objektų masyvo charakteristika: kiekis

```
static int KiekStudentuPab(Studentas[] Studentai, int kiek,
                           char simb)
{
    int kiekis = 0;
    int paskSimbInd; // vardo paskutinio simbolio indeksas
    for (int i = 0; i < kiek; i++)
        paskSimbInd = Studentai[i].ImtiPavv().Length - 1;
        if (Studentai[i].ImtiPavv()[paskSimbInd] == simb)
            kiekis = kiekis + 1;
    return kiekis;
}
...
Console.WriteLine("Raide (s) pasibaigia {0} vardai.\n",
                  KiekStudentuPab(Studentai, kiek, 's'));
...
Ekrane matysite: Raide (s) pasibaigia 5 vardai.
```

Vaikinų vardai

Objektų masyvo charakteristika: suma

```
static int Suma(Studentas[] Studentai, int kiek)
{
    int suma = 0;
    for (int i = 0; i < kiek; i++)
    {
        suma = suma + Studentai[i].ImtiPazym();
        // suma += Studentai[i].ImtiPazym();
    }
    return suma;
}

...
Console.WriteLine("Pažymių suma: {0, 3} \n",
                  Suma(Studentai, kiek));

...
Ekrane matysite: Pažymių suma: 47
```


Objektų masyvo charakteristika: vidurkis

```
static double vidurkis(Studentas[] Studentai, int kiek)
{
    double sum = 0.0;
    for (int i = 0; i < kiek; i++)
        sum += Studentai[i].ImtiPazym();
    if (kiek != 0)
        return sum / kiek;
    else
        return 0.0;
}
...
Console.WriteLine("Pažymių vidurkis: {0, 5:f2} \n",
                    vidurkis(Studentai, kiek));
...
Ekrane matysite: Pažymių vidurkis: 6,71
```

Vietoje ciklo galima rašyti kreipinį į metodą `Suma(Studentai, kiek)`

Objektų masyvo charakteristika: pažangiausio studento indeksas

```
static int MaxIndex(Studentas[] Studentai, int kiek)
{
    int maxInd = 0;
    for (int i = 1; i < kiek; i++)
        if (Studentai[i].ImtiPazym() > Studentai[maxInd].ImtiPazym())
            maxInd = i;
    return maxInd;
}
...
Console.WriteLine("Geriausio indeksas: {0, 3:d}",
                  MaxIndex(Studentai, kiek));
...
Ekrane matysite: Geriausio indeksas: 2
```

Naujo objektų masyvo formavimas

```
static void Formuoti(Studentas[] Studentai, int kiek, int riba,
                    Studentas[] StudentaiA, out int kiekA)
{
    kiekA = 0;
    for (int i = 0; i < kiek; i++)
    {
        if (Studentai[i].ImtiPazym() > riba)
        {
            StudentaiA[kiekA++] = Studentai[i];
        }
    }
}

...
Formuoti(Studentai, kiek, ribaStipendijai, StudentaiA, out kiekA);
...
Suformuoto masyvo elementų reikšmės galima pamtyti pritaikius masyvo spausdinimo metodą:
spausdintiStud(CFr, StudentaiA, kiekA, „Pirmūnų sąrašas”);
```

Demo

Demonstracinė programa MVS: kiekio, sumos, sandaugos ir vidurkio skaičiavimas objektų masyve.

Objektų rinkinių ryšiai

Skirtinguose objektų rinkiniuose objektai dažnai būna tarpusavyje susiję tam tikru ryšiu.

Objektai gali būti susieti:

- pagal indeksą;
- pagal reikšmę;
- kt.



Susiejimas indeksu

Objektų susiejimas indeksu

Galimi **du susiejimo indeksu** variantai:

- 1) Informacijos yra daug ir ji išskaidyta į keletą objektų rinkinių. Pavyzdžiui, viename rinkinyje informacija apie studentų programavimo žinių įvertinimą (testas), o kitame – apie studentų asmens duomenis. Norėdami informacijos apie i-tojo studento mokyklą, kurią baigė, kreipiamės tuo pačiu indeksu **i** į objektų rinkinį apie asmens duomenis. Šiuo atveju objektų **rinkiniai yra vienodo ilgio**, o atitinkamų **objektų indeksai sutampa**.
- 2) Rinkinio objektas informacijos išrinkimui saugo kito rinkinio objekto indeksą.

Sutampa objektų indeksai (1/7)

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 (2/7)

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 masyvuose) surašyti ta pačia tvarka, duomenų kiekis abiejuose failuose yra vienodas, duomenys teisingi.

Sutampa objektų indeksai (3/7)

```
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; }
```

```
}
```

Sutampa objektų indeksai (4/7)

```
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 (5/7)

```
const int Cn = 500; // maksimalus studentų skaičius
Studentas[] Studentai = new Studentas[Cn]; // stud. testo rezultatai
StudentasF[] StudentaiF = new StudentasF[Cn]; // studentų asmens duom.
int kiek; // studentų skaičius
```

```
SkaitytiStud(CFd, Studentai, out kiek);
SpausdintiStud(CFr, Studentai, kiek, "Studentų sąrašas");
SkaitytiStudF(CFdF, StudentaiF, out kiek);
SpausdintiStudF(CFr, StudentaiF, kiek, "Studentų sąrašas
(fakulteto)");
```

```
Console.WriteLine("Užrašykite mokyklos (gimnazijos) pavadinimą");
string mokykla = Console.ReadLine();
SpausdintiStud2(CFr, Studentai, StudentaiF, kiek,
               mokykla, "Studentų sąrašas: " + mokykla);
```

Pastaba: skaitymo metodų universalumui užtikrinti abu metodai grąžina studentų skaičių **kiek**.

Sutampa objektų indeksai (6/7)

```
static void spausdintiStud2(string fv, Studentas[] Studentai, StudentasF[] StudentaiF,
                           int kiek, string mokykla, string antraste)
{
    const string virsus =
        "-----\n"
        + " Nr.  Pavardė ir vardas      Pažymys\n"
        + "-----";
    using (var fr = File.AppendText(fv))
    {
        fr.WriteLine("\n      " + antraste);
        fr.WriteLine(virsus);
        for (int i = 0; i < kiek; i++)
        {
            int poz = String.Compare(StudentaiF[i].ImtiMokykla(), mokykla,
                                     StringComparison.CurrentCulture);

            if (poz == 0)
                fr.WriteLine("{0, 3}  {1, -20}  {2, 2}",
                             i + 1, StudentaiF[i].ImtiPavv(), Studentai[i].ImtiPazym());
        }
        fr.WriteLine("-----\n");
    }
}
```

Tai blogo metodo
pavyzdys !

Kodėl?
Kaip reikėtų
pakoreguoti šį metodą?

Sutampa objektų indeksai (7/7)

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
4	Giedraitis Giedrius	5

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;
```

```
// Surašo masyvo Pazymiai(kiek) element. atvirkščia tvarka
static void Apsukti(Pazymys[] Pazymiai, int kiek)
{
    for (int i = 0; i < kiek/2; i++)
    {
        Pazymys p = Pazymiai[i];
        Pazymiai[i] = Pazymiai[kiek - i - 1];
        Pazymiai[kiek - i - 1] = p;
    }
}
```


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(CnP)

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

```
const int Cn = 500;    // maksimalus studentų skaičius
const int CnP = 10;    // pažymių skaičius vertinimo sist.
Studentas[] Studentai = new Studentas[Cn];
int kiek;               // studentų skaičius
Pazymys[] Pazymiai = ... ;    // pažymių masyvas

SkaitytiStud(CFd, Studentai, out kiek);
Apsukti(Pazymiai, CnP);
RodytiPazVertes(Studentai, kiek, Pazymiai, CnP);
PapildytiPazMasyva(Studentai, kiek, Pazymiai);
Apsukti(Pazymiai, CnP); // Galima masyvo ir neapsukti atgal
SpausdintiPazym2(CFr, Pazymiai, CnP, "Žinių vertinimo sistema");
```

```
static void RodytiPazVertes(Studentas[] Studentai,  
                             int kiek, Pazymys[] Pazymiai)  
{  
    for (int i = 0; i < kiek; i++)  
    {  
        int pazStud = Studentai[i].ImtiPazym();  
        string pazPaz= Pazymiai[pazStud-1].ImtiPazReiksme();  
        Console.WriteLine("{0, 3}    {1, -20} {2, -14}",  
                            i + 1, Studentai[i].ImtiPavv(), pazPaz);  
    }  
}
```

1. Ar galima užrašyti be papildomų kintamųjų **pazStud** ir **pazPaz**?
2. Ar nereikia perduoti metodui masyvo **Pazymiai** elementų kiekio reikšmės?

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

```
static void PapildytiPazMasyva(Studentas[] Studentai,  
                                int kiek, Pazymys[] Pazymiai)  
{  
    for (int i = 0; i < kiek; i++)  
    {  
        int pazStud = Studentai[i].ImtiPazym();  
        int pazPaz = Pazymiai[pazStud - 1].ImtiPazKieki();  
        Pazymiai[pazStud - 1].DetiPazKieki(pazPaz+1);  
    }  
}
```

Faile matysite:

Žinių vertinimo sistema

 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



Susiejimas reikšme

Realiuose uždaviniuose dažnai skirtingų tipų objektai būna susieti pagal tam tikros savybės (-ių) reikšmę (-es).

Savybės, kurios gali sieti skirtinguose rinkiniuose esančius objektus:

- pavardė,
- pavardė ir vardas,
- asmens kodas,
- . . .

Prisiminkime ankstesnį pavyzdį.

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.

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

StudTestas.txt

StudIF.txt

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

```
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;49606066666;Gimnazija 2;
```

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

Objektų susiejimas reikšme atliekamas per **raktinį lauką** (*pavardė ir vardas, asmens kodas ir pan.*), kurio reikšmė skirtinguose rinkiniuose esančiuose objektuose sutampa.

Dirbant su keletu tokių objektų rinkinių, reikia rinkinyje surasti objektą su žinoma tam tikros savybės reikšme. Rinkiniai gali būti skirtingo ilgio.

Atsiranda (iškyla) **paieškos problema**.

Ankstesniame pavyzdyje būtų reikalinga studento paieška pagal pavardę ir vardą (raktinis laukas) antrame rinkinyje.

Paieškos masyve variantai

Netvarkingame (*nerikiuotame*)
masyve



Tvarkingame (*surikiuotame*)
masyve





Paieška nerikiuotame masyve

Elemento paieška masyve (1/3)

Duomenys:

masyvas **Mas(kiek)**;
ieškoma reikšmė **x**.

Rasti:

kurio elemento masyve **Mas(kiek)** reikšmė
sutampa su reikšme **x**?

Algoritmas:

1. Nuosekliai peržiūriname masyvo elementus.
2. Radę tinkamą reikšmę, paiešką nutraukiame.
3. Tvirtinti, kad reikiamos reikšmės nėra, galima tik peržiūrėjus visą masyvą.

Elemento paieška masyve (2/3)

// Grąžina pirmą nurodyto skaičiaus x indeksą masyve Mas(kiek),
// jeigu toks skaičius masyve yra, priešingu atveju grąžina -1

```
static int skaičiausVieta(int[] Mas, int kiek, int x)
{
    for (int i = 0; i < kiek; i++)
    {
        if (Mas[i] == x)
            return i;        // skaičius surastas
    }
    return -1;               // skaičius nerastas
}
```

Elemento paieška masyve (3/3)

```
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
}
```

Paieškoje lyginami ne patys objektai, o tam tikrų jų savybių reikšmės: pavardė ir vardas, asmens kodas, mokykla.

Kai mokėsime operatorių užklojimą, galėsime lyginti ir objektus, iki tol reikia lyginti objektų savybių reikšmes.

Paieška nerikiuotame masyve (1/3)

```
// Grąžina nurodyto (pavardė ir vardas) studento pavV indeksą,  
// priešingu atveju grąžina -1  
// StudentaiF(kiek) - studentų masyvas  
static int StudentoVieta(StudentasF[] StudentaiF, int kiek, string pavV)  
{  
    for (int i = 0; i < kiek; i++)  
    {  
        int poz = String.Compare(StudentaiF[i].ImtiPavv(), pavV,  
                                   StringComparison.CurrentCulture);  
        if (poz == 0) // ieškoma pavardė ir vardas masyve sutapo  
            return i;  
    }  
    return -1; // ieškomos pavardės ir vardo masyve nėra  
}
```

Paieška nerikiuotame masyve (2/3)

Kreipinys:

```
Console.WriteLine("Užrašykite studento pavardę ir vardą: ");  
string pavVrd = Console.ReadLine();  
int indeksas = StudentoVieta(StudentaiF, kiek, pavVrd);  
if (indeksas > -1)  
    Console.WriteLine("Pageidaujamo studento numeris: {0}\n",  
        indeksas + 1);  
else  
    Console.WriteLine("Tokio studento nėra!\n");
```

Bendruoju atveju reikia tikrinti, ar paieška buvo sėkminga.

Tai nėra būtina, jei žinoma, kad ieškomas objektas tikrai yra rinkinyje.

Paieška nerikiuotame masyve (3/3)

Pavyzdžiui ekrane užrašius: **Giedraitis Giedrius**

matyme:

Pageidaujamo studento numeris: 4

Ankstesnio pavyzdžio duomenys paieškai masyve pasitikrinti:

studentų sąrašas (fakulteto)

Nr.	Pavardė ir vardas	Asmens kodas	Mokykla

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

Paieškos reikšme susijusiuose objektų rinkiniuose pavyzdys (1/5)

Mokėdami atlikti paiešką nerikiuotame masyve galime išspręsti aukščiau suformuluotą užduotį:

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.

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

Pastaba: Įvertinant ankstesnį (blogą) metodo pavyzdį, klaidos nekartosime: pirmiausia suformuosime nurodytos mokyklos (gimnazijos) studentų rinkinį, o po to jį išspausdinsime.

Paieškos reikšme susijusiuose objektų rinkiniuose pavyzdys (2/5)

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;	49606066666;	Gimnazija 2;

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

Paieškos reikšme susijusiuose objektų rinkiniuose pavyzdys (3/5)

```
static void Formuoti(Studentas[] Studentai, StudentasF[] StudentaiF, int kiek, string mokykla,
    Studentas[] StudentaiA, out int kiekA)
{
    kiekA = 0;
    for (int i = 0; i < kiek; i++) // einama per masyvą Studentai(kiek)
    {
        string pavVrd = Studentai[i].ImtiPavv();
        int index = StudentoVieta(StudentaiF, kiek, pavVrd); // ieškoma masyve StudentaiF(kiek)
        int poz = String.Compare(StudentaiF[index].ImtiMokykla(), mokykla,
                                StringComparison.CurrentCulture);

        if ((index >= 0) && (poz == 0))
            StudentaiA[kiekA++] = Studentai[i];
    }
}
```

Paieškos reikšme susijusiuose objektų rinkiniuose pavyzdys (4/5)

```
const int Cn = 500; // maksimalus studentų skaičius
Studentas[] Studentai = new Studentas[Cn]; // stud. testo rezultatai
StudentasF[] StudentaiF = new StudentasF[Cn]; // studentų asmens duom.
int kiek; // studentų skaičius
```

```
SkaitytiStud(CFd, Studentai, out kiek);
SpausdintiStud(CFr, Studentai, kiek, "Studentų sąrašas");
SkaitytiStudF(CFdF, StudentaiF, out kiek);
SpausdintiStudF(CFr, StudentaiF, kiek, "Studentų sąrašas (fakulteto)");
```

```
Console.WriteLine("Užrašykite mokyklos (gimnazijos) pavadinimą");
string mokykla = Console.ReadLine();
Formuoti(Studentai, StudentaiF, kiek, mokykla, StudentaiA, out kiekA);
if (kiekA > 0)
    SpausdintiStud(CFr, StudentaiA, kiekA, "Studentų sąrašas: " + mokykla);
else
    Console.WriteLine("Studentų iš nurodytos mokyklos {0} nėra!", mokykla);
```

Pastaba: skaitymo metodų universalumui užtikrinti abu metodai grąžina studentų skaičių **kiek**.

Paieškos reikšme susijusiuose objektų rinkiniuose pavyzdys (5/5)

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
4	Giedraitis Giedrius	5



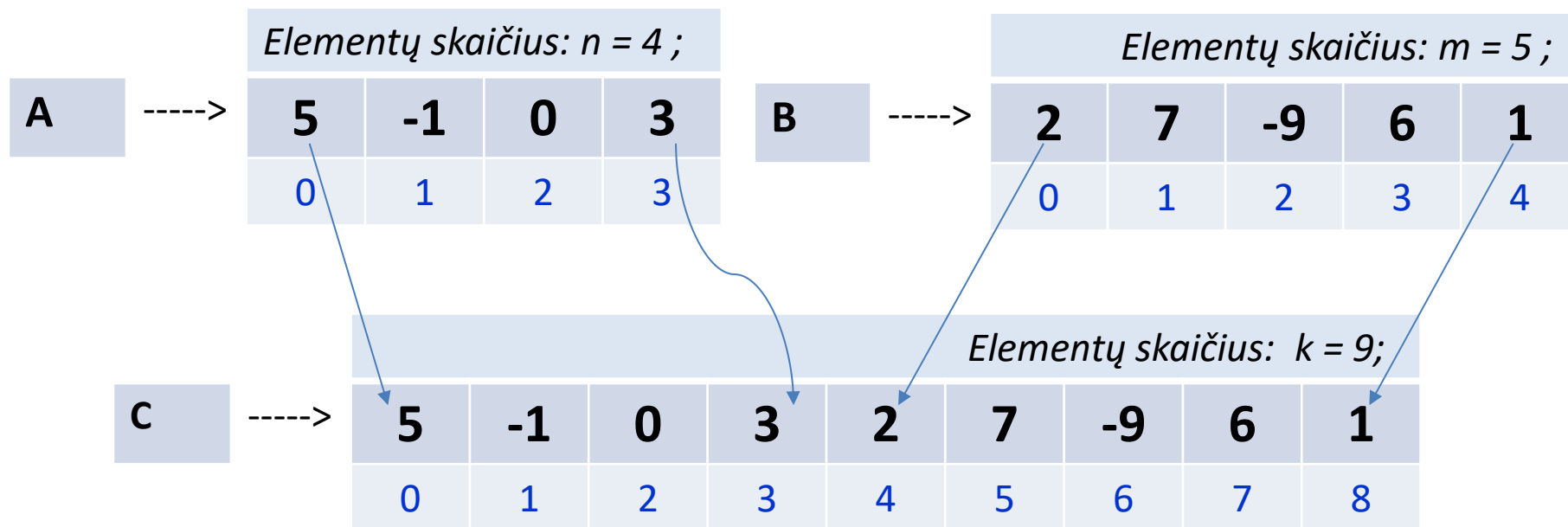
Dviejų objektų rinkinių sujungimas

Dviejų masyvų sujungimas

Pavyzdžiui, masyvų **A(n)** ir **B(m)** reikšmes reikia surašyti į masyvą **C(k)**.

Galimi atvejai:

- kai duomenys netvarkingi,
- kai duomenys tvarkingi, t.y. surikiuoti ([bus nagrinėjama vėliau](#)).



Dviejų netvarkingų masyvų sujungimas (programos fragmentas)

```
...  
k = 0;  
for (int i = 0; i < n; i++)  
{  
    C[k] = A[i];  
    k++;  
}  
for (int i = 0; i < m; i++)  
{  
    C[k] = B[i];  
    k++;  
}  
...
```

Keleto netvarkingų masyvų sujungimas

```
static void Papildyti(int[] A, int n,  
                     int[] C, ref int k)  
{  
    for (int i = 0; i < n; i++)  
    {  
        C[k] = A[i];  
        k++;  
    }  
}
```

Metodo pritaikymo pavyzdys: trys masyvai **A(n)**, **B(m)** ir **D(p)** sujungiami į vieną masyvą **C(k)**

```
k = 0;  
Papildyti(A, n, C, ref k);  
Papildyti(B, m, C, ref k);  
Papildyti(D, p, C, ref k);
```

Pastaba: Sujungiant masyvus svarbu, kad pakaktų vietos masyve, į kurį surašomi kitų masyvų informacija. Tuo reikia pasirūpinti iš anksto, išskiriant masyvui atmintį.

Keleto netvarkingų masyvų sujungimas

Skaičių masyvų sujungimo algoritmas tinka ir objektų masyvų sujungimui.

Gali būti reikalavimai objektų masyvų perrašymo į vieną masyvą tvarkai. Pvz.:

- kurio masyvo objektus perrašyti pirmiau?
- kokia seka perrašyti objektus (pradžioje su viena savybės reikšme, po to – su kita ir pan.)?

Šioje temoje susipažinote:

1. Sumos, sandaugos, kiekio skaičiavimo algoritmais.
2. Vidurkio skaičiavimo algoritmu.
3. Didžiausios reikšmės paieškos algoritmu.
4. Naujo masyvo formavimu.
5. Objektų ryšiais indeksu.
6. Objektų ryšiais reikšme.
7. Nuoseklia paieška nerikiuotame masyve.
8. Dviejų objektų rinkinių sujungimu.



Klausimai?