

4. Konteinerinė klasė

Susipažinsite su:

- konteinerine klase, jos sąsajos metodais;
- vartotojo klasės metodu `ToString()`, paveldėtu iš klasės `Object` ir užklotu vartotojo klasėje;
- eilučių formavimo operatoriumi `string.Format()`;
- dinaminio masyvu `ArrayList`, kuris saugo tik adresus į objektus;
- nauju ciklo operatoriumi `foreach`;
- `string` tipo kintamųjų palyginimu, įvertinant lietuviškus simbolius;
- palyginimo ir loginių operacijų užklojimu;
- nauju įvedimo iš failo būdu;
- rikiavimo, pašalinimo algoritmais.

4.1. Konteinerinės klasės sąsajos metodai

- Dvi klasės: viena iš jų – konteinerinė klasė,
- sąsajos metodai `Dėti()` ir `Imti()`.

Užduotis. Sodas.

Sode yra n obelių. Kiekviena obelis apibūdinama tokiais charakteristikomis: pirmaisiais metais užderėjusių obuolių kiekiu `kiek`, obuolių prieaugiu `priaug` kiekvienais metais ir dėsniu, pagal kurį didėja obuolių kiekis. Dėsniui nurodomi 2 koeficientai `koef1` ir `koef2`. Dėsnis yra bendras visoms obelims, o koeficientai gali būti skirtingi. Reikia:

- rasti kiekvienais metais kiekvienos obels užderančių obuolių kiekį, kai žinomas metų kiekis. Metų kiekis nurodomas konstanta arba įvedamas klaviatūra;
- rasti nurodytais metais kiekvienos obels užderančių obuolių kiekį;
- suformuoti naują sodą iš obelių, kurios per nurodytą metų kiekį sunokino ne mažesnę, nei nurodyta bendrą obuolių kiekį. Duoti dydžiai nurodomi konstantomis arba įvedami klaviatūra.

Dėsnis:

$$y = z * t^2 - koef2 * t + koef1, \quad t = \sqrt{\sin(koef1 * z) - 0.1}$$

z kinta nuo `kiek` žingsniu `priaug`, `koef1`, `koef2` – prieaugio koeficientai.

Pradiniai duomenys ir rezultatai.

Pradiniai duomenys	Pirmo žingsnio rezultatai
4	Informacija apie obelis
2 8 8 10	Koef1 koef2 kiek prieaug
1 12 10 14	2 8 8 10
3 8 7 10	1 12 10 14
5 12 6 14	3 8 7 10
	5 12 6 14

Programos kūrimo eiga.

- Sukuriama klasė `Obelis` vienos obels duomenims saugoti. Parašomas konstruktorius ir spausdinimo metodas.
- Sukuriama konteinerinė klasė `Sodas` sodo duomenims saugoti. Parašomas konstruktorius ir sąsajos metodai. Už pagrindinio metodo `Main()` parašomi įvedimo ir spausdinimo metodai.
- Realizuojamas užduoties a) punktas.
- Realizuojamas užduoties b) punktas.
- Realizuojamas užduoties c) punktas.

U Pirmas žingsnis.

- Sukurkite klasę `Obelis` duomenims saugoti:

```
//-----
/** Obels klasė
@class Obelis */
class Obelis
{
    private int kiek,          // pirmaisiais metais užderėjusių obuolių kiekis:
                                // atitinka užduotyje z0
        priaug;               // obuolių priaugis kiekvienais metais:
                                // atitinka užduotyje zh
    private int koef1, koef2;  // dėsnio koeficientai: atitinka užduotyje a ir b

    //-----
    /** Pradiniai obels duomenys */
    //-----
    public Obelis()
    {
        kiek = 0;
        priaug = 16;
        koef1 = 1;
        koef2 = 2;
    }

    //-----
    /** Obels duomenys
    @param kiek    - pirmaisiais metais užderėjusių obuolių kiekis
    @param priaug  - obuolių priaugis kiekvienais metais
    @param koef1   - dėsnio koeficientas a
    @param koef2   - dėsnio koeficientas b */
    //-----
    public Obelis(int kiek, int priaug, int koef1, int koef2)
    {
        this.kiek = kiek;
        this.priaug = priaug;
        this.koef1 = koef1;
        this.koef2 = koef2;
    }

    //-----
    // Spausdinimo metodas
    //-----
    public override string ToString()
    {
        string eilute;
        eilute = string.Format("{0, 5:d} {1, 6:d} {2, 5:d} {3, 7:d}",
            koef1, koef2, kiek, priaug);
        return eilute;
    }
}
//-----
```

- Sukurkite klasę Sodos sodo duomenims saugoti:

```
//-----
/** Sodo klasė
@class Soda */
class Soda
{
    const int CMaxi = 100;
    private Obelis [] Obelys;
    private int n;

    public Soda()
    {
        n = 0;
        Obelys = new Obelis[CMaxi];
    }

    /** Gražina nurodyto indekso obels objektą.
    @param i - obels indeksas */
}
```

```

    public Obelis Imti(int i) { return Obelys[i]; }

    /** Grąžina obelių kiekį */
    public int Imti() { return n; } // metodų užklojimas

    /** Padeda į obelių objektų masyvą naują obelį ir
    // masyvo dydį padidina vienetu.
    @param ob - obelis objektas */
    public void Dėti(Obelis ob) { Obelys[n++] = ob; }
}
//-----

```

- Parašykite pagrindinį metodą Main(), įvedimo ir spausdinimo metodus už jo:

```

//-----
class Program
{
    const string CFd = "...\\...\\U1.txt";

    static void Main(string[] args)
    {
        Sodas sodas = new Sodas();
        Skaityti(ref sodas, CFd);
        Spausdinti(sodas);

        Console.WriteLine("Programa baigė darbą!");
    }

    //-----
    /** Failo duomenis surašo į konteinerį
    @param sodas - obelių konteineris
    @param fv - duomenų failo vardas */
    //-----
    static void Skaityti(ref Sodas sodas, string fv)
    {
        int koef1, koef2, kiek, priaug, n;
        string line;
        using (StreamReader reader = new StreamReader(fv))
        {
            n = int.Parse(reader.ReadLine());
            for (int i = 0; i < n; i++)
            {
                line = reader.ReadLine();
                string[] parts = line.Split(' ');
                koef1 = int.Parse(parts[0]);
                koef2 = int.Parse(parts[1]);
                kiek = int.Parse(parts[2]);
                priaug = int.Parse(parts[3]);
                Obelis ob = new Obelis(kiek, priaug, koef1, koef2);
                sodas.Dėti(ob);
            }
        }
    }

    //-----
    /** Spausdina konteinerio duomenis ekrane lentelė
    @param sodas - obelių konteineris */
    //-----
    static void Spausdinti(Sodas sodas)
    {
        string virsus = "    Informacija apie obelis  \r\n"
            + " ----- \r\n"
            + " Nr. koef1 koef2 kiek priaug \r\n"
            + " ----- ";
        Console.WriteLine(virsus);
        for (int i = 0; i < sodas.Imti(); i++)
            Console.WriteLine("{0, 4:d} {1}", i + 1, sodas.Imti(i).ToString());
        Console.WriteLine(" ----- \n\n");
    }
}

```

```

    }
//-----

```

- Sukurkite duomenų failą vardu U1.txt. Kompiliuokite programą. Patikrinkite atsakymą:

Informacija apie obelis

Nr.	koef1	koef2	kiek	prieaug
1	2	8	8	10
2	1	12	10	14
3	3	8	7	10
4	5	12	6	14

Programa baigė darbą!

Press any key to continue . . .

➊ Antras žingsnis.

- Papildykite klasę Obelis metodais Dėsnis() ir Obuoliai(). Atkreipkite dėmesį į metodo Dėsnis() realizaciją – obuoliai dalimis neauga, dėl to buvo panaudotas apvalinimo metodas į mažesnę pusę Math.Floor() ir tipų konversija:

```

//-----
/** Pagal nurodytą dėsnį - formulę apskaičiuoja ir
// grąžina užderėjusių obuolių kiekį
@param a - 1-asis dėsnio koeficientas
@param b - 2-asis dėsnio koeficientas
@param z - dėsnio parametras */
//-----
public int Dėsnis(int a, int b, double z)
{
    if (Math.Sin(a * z) > 0.1)
    {
        double t = Math.Pow(Math.Sin(a * z) - 0.1, 0.5);
        int y = (int)Math.Floor(a - b * t + z * t * t);
        return y;
    }
    else
        return 0;
}

//-----
/** Apskaičiuoja ir ekrane lentelę spausdina kiekvienais metais iki
// nurodytų metų obels užderėjusių obuolių kiekį
@param metai - metų kiekis */
//-----
public void Obuoliai(int metai)
{
    int z = kiek;
    int y;
    Console.WriteLine(" -----");
    Console.WriteLine("  Metai  Obuolių kiekis ");
    Console.WriteLine(" -----");
    for (int i = 0; i < metai; i++)
    {
        y = Dėsnis(koef1, koef2, z);
        if (y > 0)
            Console.WriteLine("{0,5:d}  {1,8:d}", i + 1, y);
        else
            Console.WriteLine("{0,5:d}      nėra obuolių", i + 1);
        z = z + prieaug;
    }
    Console.WriteLine(" -----\\r\\n");
}
//-----

```

- Pagrindinę klasę Program papildykite metodu Skaičiuoti():

```

//-----
/** Spausdina kiekvienos konteinerio obels kiekvienų metų

```

```
// derlių ekrane lentelė
@param sodas - obelių konteineris
@param metai - metų kiekis */
//-----
static void Skaiciuoti(Sodas sodas, int metai)
{
    Console.WriteLine(" Informacija apie derlių");
    for (int i = 0; i < sodas.Imti(); i++)
    {
        Console.WriteLine("{0,3:d} obelis", i + 1);
        sodas.Imti(i).Obuoliai(metai);
    }
}
//-----
```

- Papildykite pagrindinį metodą Main() kreipiniu į šį metodą. Pasirinkite, koku būdu nurodysite metus: konstanta ar klaviatūra:

```
int metai;
Console.Write("Įveskite metų reikšmę: ");
metai = int.Parse(Console.ReadLine());
Skaiciuoti(sodas, metai);
```

- Jei metų kiekis 4, tai atsakymas turi būti toks:

```
Įveskite metų reikšmę: 4
Informacija apie derlių
1 obelis
```

Metai	Obuolių kiekis
1	nėra obuolių
2	nėra obuolių
3	nėra obuolių
4	14

2 obelis

Metai	Obuolių kiekis
1	nėra obuolių
2	nėra obuolių
3	3
4	35

3 obelis

Metai	Obuolių kiekis
1	1
2	6
3	nėra obuolių
4	nėra obuolių

4 obelis

Metai	Obuolių kiekis
1	nėra obuolių
2	nėra obuolių
3	7
4	34

U Trečias žingsnis.

- Suraskite nurodytų metų kiekvienos obels išaugintų obuolių kiekį. Jei metai 4, tai atsakymas turi būti toks:

Informacija apie derlių**Obels Nr. obuolių kiekis**

1	14
2	35
2	nėra obuolių
4	34

U Ketvirtas žingsnis.

- Papildykite klasę `Obelis` metodu `VisoObuolių()`, kuris suskaičiuotų obels derlių per prabėgusius metus.
- Pagrindinę klasę `Program` papildykite metodu `Formuoti()`, kuris atrinks į naują objektų sąrašą tas obelis, kurių derlius yra didesnis už nurodytą kiek (pirmaisiais metais užderėjusių obuolių kiekį) reikšmę. Pasirinkite, koku būdu nurodysite obels derliaus kiekį: konstanta ar įvedimu klaviatūra.
- Nepamirškite paskelbti naujo `Sodas` objekto.
- Įdėkite kreipinį į naują metodą, o taip pat ir į rezultatų spausdinimo metodą:

```
//-----
/** Iš pirmojo konteinerio atrenka į antrąjį konteinerį obelis, kurios per
// nurodytą metų kiekį sunokina daugiau, negu nurodytas kiekis obuolių
@param sodas - pirmasis obelių konteineris
@param metai - metų kiekis
@param sodasN - antrasis obelių konteineris
@param kiekis - obuolių kiekis */
//-----
static void Formuoti(Sodas sodas, int metai, ref Sodas sodasN, int kiek)
{
    for (int i = 0; i < sodas.Imti(); i++)
    {
        if (sodas.Imti(i).VisoObuolių(metai) > kiek)
            sodasN.Dėti(sodas.Imti(i));
    }
}
//-----
```

- Išbandykite programą. Jei nurodytas derliaus dydis yra 37, tai atsakymas turi būti toks:

Sunokintų obuolių kiekis: 37
Informacija apie obelis

Nr.	koef1	koef2	kiek	prieaug
1	1	12	10	14
2	5	12	6	14

Savarankiško darbo užduotis.

Pateikiamas 9 aukštų namo parduodamų butų sąrašas. Kiekviename laiptinės aukšte yra po 3 butus. Žinomas buto numeris, bendras plotas, kambarių skaičius, pardavimo kaina, telefono Nr. Suraskite butus, kurie turi nurodytą kambarių skaičių ir kurių kaina neviršija nurodytos kainos, ir juos surašykite į tinkamų butų masyvą.

4.2.Operatorių užklojimas konteineriye

- Dinaminis masyvas `ArrayList`, kuris saugo tik adresus į objektus;
- naujas ciklo operatorius `foreach`;
- naujas įvedimo iš failo būdas;
- `string` tipo kintamųjų palyginimu, įvertinant lietuviškus simbolius;
- operatorių `<=`, `!` užklojimų realizavimas;
- rikiavimas;
- šalinimas.

Užduotis. Fakultetas.

Tekstiniame faile saugoma informacija apie studentus: pavardė, vardas, grupė, pažymiai. Sudaryti konteinerinę klasę, saugančią informaciją apie studentus. Reikia:

- sudaryti naują sąrašą, kuriame būtų tik pirmūnai (gavo tik 10 arba 9);
- surikiuoti pradinį sąrašą pagal pavardę ir vardą abėcėliškai;
- pašalinti iš pradinio sąrašo studentus, kurie nėra pirmūnai.

Pradiniai duomenys ir rezultatai.

Pradiniai duomenys						
Petraitis; Jonas; IF-1/8; 10 8 9 9						
Algaitis; Algis; IF-1/8; 10 9 9 9						
Petraitis; Kazys; IF-1/8; 9 9 10 10						
Algaitis; Rimas; IF-1/8; 9 9 9 9						
Petraitis; Vytas; IF-1/9; 10 8 9 7						
Petraitis; Anupras; IF-1/9; 9 8 7 6						
Petraitis; Vidas; IF-1/9; 8 9 8 9						
Petraitis; Anzelmas; IF-1/9; 8 9 9 9						
Petraitis; Šarūnas; IF-1/9; 10 9 10 9						
Rezultatai						
Pradinis studentų sąrašas						
Pavardė	Vardas	Grupė	Pažymiai			
Petraitis	Jonas	IF-1/8	10	8	9	9
Algaitis	Algis	IF-1/8	10	9	9	9
Petraitis	Kazys	IF-1/8	9	9	10	10
Algaitis	Rimas	IF-1/8	9	9	9	9
Petraitis	Vytas	IF-1/9	10	8	9	7
Petraitis	Anupras	IF-1/9	9	8	7	6
Petraitis	Vidas	IF-1/9	8	9	8	9
Petraitis	Anzelmas	IF-1/9	8	9	9	9
Petraitis	Šarūnas	IF-1/9	10	9	10	9
Naujas studentų sąrašas						
Pavardė	Vardas	Grupė	Pažymiai			
Algaitis	Algis	IF-1/8	10	9	9	9
Petraitis	Kazys	IF-1/8	9	9	10	10
Algaitis	Rimas	IF-1/8	9	9	9	9
Petraitis	Šarūnas	IF-1/9	10	9	10	9
Rikiuotas naujas studentų sąrašas						
Pavardė	Vardas	Grupė	Pažymiai			
Algaitis	Algis	IF-1/8	10	9	9	9
Algaitis	Rimas	IF-1/8	9	9	9	9
Petraitis	Kazys	IF-1/8	9	9	10	10
Petraitis	Šarūnas	IF-1/9	10	9	10	9
Pradinis studentų sąrašas po šalinimo						
Pavardė	Vardas	Grupė	Pažymiai			
Algaitis	Algis	IF-1/8	10	9	9	9
Petraitis	Kazys	IF-1/8	9	9	10	10
Algaitis	Rimas	IF-1/8	9	9	9	9
Petraitis	Šarūnas	IF-1/9	10	9	10	9

Programos kūrimo eiga.

- Sudaroma klasė `Studentas` studento duomenims saugoti. Parašomas konstruktorius be parametų, metodas `Dėti()` ir metodas spausdinimui į eilutę.
- Sudaroma konteinerinė klasė `Fakultetas` fakulteto duomenims saugoti. Parašomas konstruktorius be parametų, sąsajos metodas skaitliuko paėmimui, sąsajos metodai elemento paėmimui ir įrašymui.
- Pagrindinėje klasėje parašomas įvedimas ir spausdinimas.
- Realizuojamas užduoties a) punktas: papildoma klasė `Studentas` ir pagrindinė klasė.
- Realizuojamas užduoties b) punktas: papildoma klasė `Studentas` ir `Fakultetas` bei pagrindinė klasė.
- Realizuojamas užduoties c) punktas.

U Pirmas žingsnis.

- Sukurkite klasę studento duomenims saugoti. Panaudokite dinaminio masyvo tipą `ArrayList`. Jam reikia naujos direktyvos:

```
using System.Collections;

//-----
/** Klasė studento duomenims saugoti
@class Studentas */
class Studentas
{
    private string pavardė,    // studento pavardė
        vardas,              // studento vardas
        grupė;               // mokymosi grupė
    private ArrayList paž;    // pažymių masyvas

    //-----
    /** Pradiniai studento duomenys, išskyrus pažymius */
    //-----
    public Studentas()
    {
        pavardė = "";
        vardas = "";
        grupė = "";
        paž = new ArrayList();
    }

    //-----
    /** Studento duomenų įrašymas
    @param pav - nauja pavardės reikšmė
    @param vard - nauja vardo reikšmė
    @param grup - nauja grupės reikšmė
    @param pž - naujos pažymių reikšmės */
    //-----
    public void Dėti(string pav, string vard, string grup, ArrayList pž)
    {
        pavardė = pav;
        vardas = vard;
        grupė = grup;
        foreach (int sk in pž)
            paž.Add(sk);
    }

    //-----
    /** Spausdinimo metodas
    //-----
    public override string ToString()
    {
        string eilute;
        eilute = string.Format("{0, -12} {1, -9} {2, -7}",
            pavardė, vardas, grupė);
        foreach (int sk in paž)
            eilute = eilute + string.Format("{0, 3:d}", sk);
        return eilute;
    }
}
```



```
//-----
• Sukurkite klasę fakulteto duomenims saugoti:
//-----
/** Klasė studentų grupės duomenims saugoti
@class Fakultetas */
class Fakultetas
{
    const int CMax = 100;    // maksimalus studentų skaičius
    private Studentas[] St; // studentų duomenys
    private int n;           // studentų skaičius

    public Fakultetas()
    {
        n = 0;
        St = new Studentas[CMax];
    }

    /** Grąžina studentų skaičių */
    public int Imti() { return n; }

    /** Grąžina nurodyto indekso studento objektą
    @param i - studento indeksas */
    public Studentas Imti(int i) { return St[i]; }

    /** Padeda į studentų objektų masyvą naują studentą ir
    // masyvo dydį padidina vienetu
    @param ob - studento objektas */
    public void Dėti(Studentas ob) { St[n++] = ob; }
}
//-----
```

- Parašykite pagrindinį metodą Main(), įvedimo bei išvedimo metodus: duomenys įvedami iš failo, išvedami – į failą. Pavardės ir vardai gali būti sudaryti iš kelių žodžių, todėl įvedimo metodas bus pakankamai sudėtingas. Eilučių skaičius duomenų faile nurodytas. Taip pat reikia tikrinti, kad neviršytume masyvo dydžio. Ruošdami duomenų failą, po paskutinės eilutės „Enter“ klavišo nespauskite, nes įvesite papildomą nereikalingą eilutę.

```
//-----
class Program
{
    const string CFd = "..\\..\\U1.txt";
    const string CFr = "..\\..\\Rezultatai.txt";

    static void Main(string[] args)
    {
        Fakultetas grupes = new Fakultetas();
        if (File.Exists(CFr))
            File.Delete(CFr);
        Skaityti(ref grupes, CFd);
        Spausdinti(grupes, CFr, " Pradinis studentų sąrašas");

        Console.WriteLine("Programa baigė darbą!");
    }

    //-----
    /** Failo duomenis surašo į konteinerį
    @param grupe - studentų konteineris
    @param fv     - duomenų failo vardas */
    //-----
    static void Skaityti(ref Fakultetas grupe, string fv)
    {
        string pv, vrd, grp;
        ArrayList pz = new ArrayList();
        string[] lines = File.ReadAllLines(fv, Encoding.GetEncoding(1257));
        foreach (string line in lines)
        {
            string[] parts = line.Split(';');
            pv = parts[0].Trim();

```

```

        vrd = parts[1].Trim();
        grp = parts[2].Trim();
        // Toliau pažymiai
        string[] eil = parts[3].Trim().Split(new[] { ' ' },
                                           StringSplitOptions.RemoveEmptyEntries);
        pz.Clear();
        foreach (string eilute in eil)
        {
            int aa = int.Parse(eilute);
            pz.Add(aa);
        }
        Studentas stud = new Studentas();
        stud.Dėti(pv, vrd, grp, pz);
        grupe.Dėti(stud);
    }
}

//-----
/** Sprendina konteinerio duomenis faile lentele
@param grupe    - studentų konteineris
@param fv       - rezultatų failo vardas
@param antraste - užrašas virš lentelės */
//-----
static void Spsausdinti(Fakultetas grupe, string fv, string antraštė)
{
    string virsus =
        "-----\r\n"
    + " Pavardė    Vardas    Grupė    Pažymiai  \r\n"
    + "-----";
    using (var fr = File.AppendText(fv))
    {
        fr.WriteLine(antraštė);
        fr.WriteLine(virsus);
        for (int i = 0; i < grupe.Imti(); i++)
            fr.WriteLine("{0}", grupe.Imti(i).ToString());
        fr.WriteLine("-----\r\n");
    }
}
}
}
//-----

```

- Sukurkite duomenų failą U1.txt. Kompiliuokite programą. Patikrinkite atsakymą.

🔗Antras žingsnis.

- Sukurkite naują Fakultetas klasės objektą.
- Papildykite klasę Studentas operatoriaus ! užklojimo metodu:

```

//-----
/** Operatorius ! grąžina
// true, jeigu bent vienas pažymys yra mažesnis už 9;
// false - kitais atvejais */
//-----
public static bool operator !(Studentas c1)
{
    foreach (int sk in c1.paž)
    {
        if (sk < 9)
            return true;
    }
    return false;
}
}
//-----

• Pagrindinę klasę papildykite metodu Formuoti():
//-----
/** Iš pirmojo konteinerio atrenka į antrąjį konteinerį studentus,
// kurių įvertinimai yra 9 arba 10

```

```

@param D - pirmasis studentų konteineris
@param R - antrasis studentų konteineris */
//-----
static void Formuoti(Fakultetas D, ref Fakultetas R)
{
    for (int i = 0; i < D.Imti(); i++)
        if (!D.Imti(i))
            ;
        else
            R.Dėti(D.Imti(i));
    }
}
//-----

```

- Papildykite pagrindinį metodą `Main()` kreipiniu į metodą `Formuoti()` ir spausdinimu:

```

Formuoti(grupes, ref grupes1);
if (grupes1.Imti() > 0)
    Spausdinti(grupes1, CFr, " Naujas studentų sąrašas");
else
    using (var fr = File.AppendText(CFr))
    {
        fr.WriteLine("Tokių studentų nėra");
    }

```

☛ Trečias žingsnis.

- Realizuokite rikiavimo metodą klasėje `Fakultetas`. Tuo tikslu papildykite klasę `Studentas` operatorių `<=` ir `>=` užklojimo metodais, nes kompiliatorius reikalauja dviejų metodų, vieno palyginimo metodo užklojimo neužtenka:

```

//-----
/** Operatorius grąžina
// true, jeigu pavardė yra mažesnė už kitą pavardę, arba pavardės yra lygios,
// o vardas yra mažesnis už kitą vardą;
// false - kitais atvejais. */
//-----
public static bool operator <=(Studentas st1, Studentas st2)
{
    int p = String.Compare(st1.pavardė, st2.pavardė,
                           StringComparison.CurrentCulture);
    int v = String.Compare(st1 vardas, st2 vardas,
                           StringComparison.CurrentCulture);
    return (p < 0 || (p == 0 && v < 0));
}

//-----
/** Operatorius grąžina
// true, jeigu pavardė yra didesnė už kitą pavardę, arba pavardės yra lygios,
// o vardas yra didesnis už kitą vardą;
// false - kitais atvejais. */
//-----
public static bool operator >=(Studentas st1, Studentas st2)
{
    int p = String.Compare(st1.pavardė, st2.pavardė,
                           StringComparison.CurrentCulture);
    int v = String.Compare(st1 vardas, st2 vardas,
                           StringComparison.CurrentCulture);
    return (p > 0 || (p == 0 && v > 0));
}

//-----

```

- Papildykite klasę `Fakultetas` rikiavimo metodu:

```
//-----
void Fakultetas::Rikiuoti()
/** Surikiuoja studentų masyvą išrinkimo metodu pagal pavardes ir vardus */
public void Rikiuoti()
{
    for (int i = 0; i < n - 1; i++)
    {
        Studentas min = St[i];
        int im = i;
        for (int j = i + 1; j < n; j++)
            if (St[j] <= min)
            { // naudojamas užklotas operatorius <=
                min = St[j];
                im = j;
            }
        St[im] = St[i];
        St[i] = min;
    }
}

//-----
```

- Išbandykite programą. Patikrinkite, ar tvarkingai surikiuoja suformuoto konteinerio studentų duomenis.
- Savarankiškai pakeiskite operatoriaus <= užklojimo metodą, kad rikiuotų pagal grupę, pavardę ir vardą abėcėliškai.

Ketvirtas žingsnis.

Savarankiškai realizuokite studentų, kurie nėra pirmūnai, pašalinimo iš pradinio sąrašo metodą, nekeisdami rikiavimo tvarkos.

Savarankiško darbo užduotis.

Pirmoje failo eilutėje nurodytas fakulteto pavadinimas. Tekstiniame faile yra fakulteto žiemos sesijos pažymių sąrašas. Eilutėje apie studentą yra tokie duomenys: pavardė, vardas, grupė, pažymių kiekis, pažymiai. Nustatykite kiekvienos grupės bendrą mokymosi vidurkį. Rezultatus išspausdinkite surikiuotus mažėjančiai pagal vidurkį ir pagal grupes abėcėliškai.

4.3. Kontroliniai klausimai

1. Kokią reikšmę grąžina užklotas operatorius, pateiktas klasės apraše?

```
class Medis {
private string pav;
private int aukstis;

    . . .

public static bool operator * (Medis kitas)
{
    return aukstis > kitas.aukstis ||
        aukstis == kitas.aukstis &&
        pav > kitas.pav;
}
}
```

- a. operatorius grąžina true, jeigu abu medžiai yra vienodo aukščio, tačiau pirmo medžio pavadinimas yra uosis, o kitas – obelis.
- b. operatorius grąžina true, jeigu kitas medis aukštesnis, o pavadinimai vienodi.
- c. operatorius grąžina true, jeigu abu medžiai yra uosiai ir jie vienodo aukščio.
- d. operatorius klaidingas.

2. Ar yra klaida tolimesniame programos fragmente? Jei taip, tai kokia klaida?:

```
int A[8];
A[0] = 1;
A[1] = 2;
for (int i = 2; i < 9; i++)
    A[i] = A[i-1] + A[i-2];
```

3. Metodo parametrai yra 2 vienodo dydžio sveikųjų skaičių masyvai ir sveikojo tipo kintamasis, nurodantis masyvų ilgį. Parašykite metodą, kuris kopijuoja pirmojo masyvo reikšmės atvirkštine tvarka į antrąjį masyvą.
4. Kokią klasę vadiname konteinerine klase?
5. Išvardinkite konteinerinės klasės sąsajos metodus ir nurodykite jų funkcijas.
6. Kokio tipo parametras turi būti klasėje užklojamame dviviečiame operatoriuje?
7. Kada iškviečiamas klasėje užklotas operatorius?
8. Papasakokite kokiomis ypatybėmis pasižymi išrinkimo rikiavimo metodas.
9. Kokio raktinio žodžio papildomai reikia, norint skelbti klasėje metodą `ToString()`? Kokia šio raktinio žodžio prasmė?
10. Kokį duomenų tipą aprašo raktinis žodis `ArrayList`?
11. Kuo yra patogus ciklo operatorius `foreach`?
12. Kokią direktyvą reikia įtraukti, norint naudoti `ArrayList`?

4.4. Užduotys

U4-1. Bukletai

Spaustuvėje spausdinamų bukletų duomenys saugomi faile: buketo formatas, 500 lapų tokio formato popieriaus pakuotės kaina, buketo lapų skaičius ir egzempliorių kiekis. Parašykite programą, kuri suskaičiuotų kiekvieno buketo gamybai reikalingo popieriaus kainą ir surastų pigiausią užsakymą. Papildykite programą veiksmiais, kurie leistų atrinkti bukletus, kurių gamybai užtenka vienos 500 lapų pakuotės, ir šį sąrašą surikiuoti pagal buketo lapų skaičių ir egzempliorių kiekį mažėjimo tvarka.

U4-2. Mobiliojo ryšio kortelės

Norėdamas palyginti mobiliojo ryšio operatorių siūlomas išankstinio mokėjimo korteles Sirvydas surinko šią informaciją į tekstinį failą. Faile eilutėmis yra kortelių duomenys: kortelės (tinklo) pavadinimas, pradinė suma kortelėje, tarifas savame tinkle, tarifas į kitus tinklus, SMS žinučių tarifas savame tinkle ir į kitus tinklus. Parašykite programą, kuri spausdintų kortelių duomenis lentelę, surastų kortelę, kurios SMS žinučių tarifai į kitus tinklus mažiausi. Papildykite programą veiksmiais, kurie leistų atrinkti korteles, kurios leidžia skambinti ir siųsti SMS žinutes savame tinkle nemokamai, ir šį sąrašą surikiuoti pagal pradinę sumą mažėjimo tvarka ir kortelės pavadinimą abėcėliškai.

U4-3. Dėžės

Ūkininkas, ieškodamas taros savo auginamai produkcijai sandėliuoti, analizuoja dėžes. Tekstiniame faile yra surašyti ūkininko analizuojamos taros duomenys. Eilutėmis surašytos dėžių charakteristikos: medžiaga, iš kurios pagaminta dėžė, ilgis, plotis, aukštis, maksimalus produkcijos svoris, kurį gali atlaikyti dėžė, bei kiek dėžių galima sukrauti viena ant kitos. Parašykite programą, kuri spausdintų dėžių duomenis lentelę, surastų dėžę, į kurią telpa daugiausiai produkcijos ir suskaičiuotų kokio aukščio lentynas turi pastatyti sandėlyje (trūkstantis sandėlio duomenis programa paprašo įvesti klaviatūra), jeigu ūkininkas naudos šias dėžes. Papildykite programą veiksmiais, kurie leistų atrinkti nurodytos medžiagos dėžes ir šį sąrašą surikiuoti pagal maksimalų produkcijos svorį ir leistiną dėžių sukrovimą viena ant kitos mažėjimo tvarka.

U4-4. Indėliai

Banke yra laikomi terminuoti indėliai. Jų duomenys surašyti tekstiniame faile eilutėmis: indėlio dydis, sutarties terminas, metinė palūkanų norma, kaip skaičiuojamos palūkanos. Palūkanos gali būti skaičiuojamos: a) indėlio palūkanos apskaičiuojamos sutartinio laikotarpio pabaigoje; b) indėlio palūkanos apskaičiuojamos kas mėnesį ir pridedamos prie indėlio; c) indėlio palūkanos apskaičiuojamos kartą per metus ir pridedamos prie indėlio. Parašykite programą, kuri spausdintų indėlių duomenis lentelę, surastų didžiausią indėlį, suskaičiuotų bendrą sumą, kurią bankas turės išmokėti visiems indėlių savininkams. Papildykite programą veiksmiais, kurie leistų atrinkti indėlius, kurių palūkanos apskaičiuojamos nurodytu būdu, ir šį sąrašą surikiuoti pagal sutarties terminą ir indėlio dydį mažėjimo tvarka.

U4-5. Matavimo prietaisai

Tekstiniame faile surašyta įvairių matavimo prietaisų informacija: pavadinimas, matavimo ribos [nuo, iki], matavimo vienetai, skalės didžiųjų padalų skaičius, mažųjų padalų skaičius didžiojoje padaloje. Parašykite programą, kuri spausdintų matavimo prietaisų informaciją lentelę, surastų didžiausią mažiausios padalos vertę turintį prietaisą (absoliutine reikšme, matavimo vienetai nesvarbūs), suskaičiuotų vienos rūšies prietaisų kiekį (prietaisų rūšį nusako pavadinimas, jį reikia įvesti klaviatūra). Papildykite programą veiksmiais, kurie leistų atrinkti prietaisus, turinčius nurodytą

mažiausios padalos vertę (absoliutine reikšme) ir šį sąrašą surikiuoti pagal matavimo ribas didėjimo tvarka.

U4-6. Lazeriniai spausdintuvai

Tekstiniame faile turime įvairių gamintojų lazerinių spausdintuvų informaciją. Faile eilutėmis surašytos spausdintuvų charakteristikos: gamintojas, modelis, vienpusio spausdinimo sparta, dvipusio spausdinimo sparta (jeigu spausdintuvas neturi dvipusio spausdinimo galimybės, sparta - 0), pirmojo puslapio išspausdinimo laikas. Parašykite programą, kuri spausdintų spausdintuvų duomenis lentelę, suskaičiuotų, kiek modelių gali spausdinti dvipusius spaudinius, surastų sparčiausią vienpusį spausdintuvą. Papildykite programą veiksmiais, kurie leistų atrinkti tik dvipusio spausdinimo galimybę turinčius spausdintuvus ir šį sąrašą surikiuoti pagal gamintoją ir modelį abėcėliškai.

U4-7. Šulinių užterštumas

Seniūnas suregistravo kaimo sodybose esančius geriamo vandens šulinius ir atliko nitratų kiekio šių šulinių vandenyje tyrimus. Faile eilutėmis surašyti sodybų šulinių duomenys: sodybos adresas, šulinio gylis, šulinio skersmuo, nitratų kiekis. Parašykite programą, kuri spausdintų šulinių informaciją lentelę, surastų giliausią šulinį, suskaičiuotų, keliuose šuliniuose nitratų kiekis viršija leistiną normą (normą programa paprasčiausiai įvesti klaviatūra). Papildykite programą veiksmiais, kurie leistų atrinkti nurodytos gatvės sodybų šulinius ir šį sąrašą surikiuoti pagal nitratų kiekį didėjimo tvarka ir šulinio gylį mažėjimo tvarka.

U4-8. Moduliai

Tekstiniame faile duota informacija apie dėstytojų vedamus modulius: modulio pavadinimas, kreditų kiekis, atestacijos laikotarpis [nuo, iki], dėstytojo pavardė, vardas. Parašykite programą, kuri atrinktų nurodyto dėstytojo vedamus modulius į atskirą sąrašą, suskaičiuotų šio sąrašo kreditų sumą, surikiuotų pagal kreditų kiekį mažėjimo tvarka ir modulio pavadinimą abėcėlės tvarka. Papildykite programą veiksmiais, kurie leistų iš pradinio modulių sąrašo pašalinti modulius, kurių atestacija baigiasi šiais metais.

U4-9. Kopijuokliai

Tekstiniame faile surašytos skaitmeninių kopijavimo aparatų techninės charakteristikos: gamintojas, modelis, maksimalus popieriaus formatas, kopijavimo sparta, pirmojo puslapio išspausdinimo laikas, standartinės popieriaus kasetės talpa. Parašykite programą, kuri spausdintų kopijavimo aparatų duomenis lentelę, surastų didžiausią popieriaus talpą turintį kopijavimo aparatą, suskaičiuotų pastarojo aparato gamintojo gaminamų modelių kiekį. Papildykite programą veiksmiais, kurie leistų atrinkti nurodyta sparta (spartos intervalas įvedamas klaviatūra) kopijuojančius aparatus ir surikiuotų pagal spartą mažėjimo tvarka ir gamintoją abėcėlės tvarka.

U4-10. Kopijavimo darbai

Tekstiniame faile surašyti kopijavimo darbai: popieriaus formatas, lapų skaičius, kopijų skaičius, vienpusis ar dvipusis kopijavimas, kopijų grupavimas, papildoma kopijos apdorojimo funkcija. Papildoma kopijų apdorojimo funkcija, jeigu jos reikia, gali būti susegimas arba skylių išmušimas. Parašykite programą, kuri spausdintų darbų sąrašą lentelę, suskaičiuotų, kiek darbų turi papildomą susegimo funkciją ir kiek darbų turi skylių išmušimo funkciją, surastų daugiausiai kopijavimo aparato ciklą reikalaujantį darbą. Papildykite programą veiksmiais, kurie leistų atrinkti vienpusio kopijavimo darbus ir surikiuotų juos pagal formatą ir bendrą reikalingo popieriaus kiekį mažėjimo tvarka.

U4-11. Reklaminių klipai

Tekstiniame faile surašyti reklaminių video klipų duomenys: reklamuojamas produktas, produkto grupė, video klipo trukmė, klipo kraštinių santykis pikseliais (formatas). Parašykite programą, kuri spausdintų klipų duomenis lentelę, suskaičiuotų bendrą klipų trukmę, sudarytų reklaminių klipų sąrašą rodymui, kuriame būtų rodomi nurodyto vienodo formato klipai ir būtų rodomi po vieną klipą iš kiekvienos produktų grupės, jį surikiuotų pagal klipo trukmę ir produkto grupę ir suskaičiuotų atrinktų klipų rodymo trukmę.

U4-12. Lojalūs klientai

Tekstiniame faile yra lojalių parduotuvės klientų sąrašas: kortelės numeris, sukaupta virtualių pinigų suma, dažniausiai perkamos prekės pavadinimas, nupirktas kiekis, šiai prekei išleista pinigų suma. Parašykite programą, kuri spausdintų lojalių klientų duomenis lentelę, surastų didžiausią populiariausiai prekei išleistą sumą, suskaičiuotų vidutinę virtualių pinigų sumą. Papildykite programą veiksmiais, kurie leistų pasiūlyti klientams jų populiariausią prekę su nuolaida nuo vidutinės šios prekės kainos. Nuolaida klientui gali būti suteikiama, jeigu jo išleista pinigų suma ir nupirktas prekės kiekis viršija nurodytus dydžius. Pastarieji kartu su nuolaidos dydžiu įvedami klaviatūra. Sudarytą

nuolaidų sąrašą programa turi spausdinti surikiuotą pagal prekės kainą ir kortelės numerį didėjimo tvarka.

U4-13. Stipendijos

Tekstiniame faile pateikta informacija apie studentus. Pirmojoje failo eilutėje nurodytas stipendijų fondo dydis ir pažymių vidurkis stipendijai gauti. Tolimesnėse eilutėse tokia informacija: studento pavardė vardas, grupė, pažymių kiekis, pažymiai. Studentui skiriama stipendija, jei jo pažymių vidurkis viršija nurodytą dydį ir jis neturi skolų (visi pažymiai >4). Studentui skiriama 10% didesnė stipendija, jei jo visi pažymiai didesni už 8. Toks studentas vadinamas pirmūnu. Paskirstykite studentams stipendijas pagal duotą fondą. Fondą reikia maksimaliai išnaudoti, bet negalima viršyti fondo dydžio. Spausdinamas sąrašas turi būti surikiuotas pagal stipendijų dydį, pavardes ir vardus. Iš sąrašo pašalinkite studentus, kurie negauna stipendijos. Suformuokite ir atspausdinkite nurodytos grupės (įvedama klaviatūra) pirmūnų sąrašą.

U4-14. Įvertinimai

Tekstiniame faile pateikta informacija apie studentus. Pirmoje failo eilutėje nurodytas fakulteto pavadinimas ir teisingas atsakymas, pvz.: NTNNTNNTTT. Vienoje eilutėje yra tokia informacija: studento Nr., studento pavardė, vardas, atsakymas. Reikia surasti, kiek taškų T1 surinko geriausiai atsakęs studentas. Reikia įvertinti visus studentus. Įvertinimai turi būti tokie: jei studento surinktas taškų skaičius lygus T1 arba T1-1, tai įvertinimas „gerai“, jei taškų skaičius lygus T1-2 arba T1-3, tai įvertinimas „patenkinamai“, likusiais atvejais - „nepatenkinamai“. Atspausdinti rezultatus, surikiuotus pagal taškus ir pavardes. Sąraše palikti tik tuos, kurie gavo įvertinimą „gerai“, o kitus pašalinti. Atspausdinkite po pašalinimo.

U4-15. Žaidėjai

Tekstiniame faile pateikta informacija apie krepšinio komandos žaidėjus. Pirmoje failo eilutėje nurodytas komandos pavadinimas, žaistų rungtynių skaičius. Kitose eilutėse - informacija apie krepšininkus: pavardė, vardas, ūgis, žaista rungtynių, įmesta taškų, padaryta klaidų. Sudaryti sąrašą žaidėjų, kurie žaidė visas komandos rungtynes ir pelnė daugiau taškų, nei komandos žaidėjų taškų vidurkis. Surikiuokite sudarytą sąrašą pagal ūgį ir žaidėjo pavardę. Sąraše palikite tik tuos žaidėjus, kurių ūgis patenka į intervalą [n1, n2], o kitus – pašalinkite iš sąrašo.

U4-16. Prenumeratoriai

Tekstiniame faile pateikta informacija apie futbolininkus. Failo eilutėje nurodyta prenumeratoriaus pavardė, adresas, laikotarpio pradžia (sveikasis skaičius 1..12), laikotarpio ilgis (sveikasis skaičius 1..12), leidinio kodas, leidinių kiekis. Laikotarpio pradžia + laikotarpio ilgis <= 13. Sudaryti sąrašą prenumeratorių, kurie užsisakė leidinius daugiau kaip 1 mėnesiui. Atspausdinti pagal adresus ir pavardes, vaizduojant užsakytus mėnesius žvaigždutėmis, o neužsakytus - taškais. Išrinkti tarp jų prenumeratorių, kuris užsisakė daugiausiai leidinių.

U4-17. Futbolininkai

Tekstiniame faile pateikta informacija apie prenumeratorių: komanda, pavardė, vardas, žaistų rungtynių skaičius, įmuštų įvarčių skaičius. Surasti kiekvienos komandos futbolininką, kurio naudingumo koeficientas didžiausias (žaista ne mažiau kaip vidutinis rungtynių skaičius ir įmušta daugiausiai įvarčių) ir suformuoti iš jų sąrašą. Surikiuoti futbolininkus pagal komandas ir pavardes.

U4-18. Fakultetai

Tekstiniame faile pateikta informacija apie studentus. Eilutėje apie studentą yra tokie duomenys: fakultetas, pavardė, vardas, grupė, lytis, dešimtųjų, devynių, aštuonetų kiekiai ir bendras studento laikytų egzaminų kiekis. Nustatyti, kurio fakulteto studentai yra geriausi. Juos atrinkti į atskirą sąrašą. Surikiuoti šio fakulteto studentus pagal grupes ir pavardes, pirmūnus (visi pažymiai > 8) pažymint žvaigždute.

U4-19. Dėstytojai

Tekstiniame faile pateikta informacija apie studentų pasirenkamus modulius. Eilutėje yra tokie duomenys: modulio pavadinimas, atsakingo dėstytojo pavardė, vardas, kreditų kiekis, studento pavardė, vardas, grupė. Dėstytojas gali būti atsakingas už keletą modulių. Suformuokite dėstytojų sąrašą. Sąrašas turi būti surikiuotas pagal pavardes abėcėlės tvarka. Pašalinkite iš šio sąrašo dėstytojus, kurių modulius pasirinko tik po vieną studentą. Suraskite, kuris dėstytojas turi daugiausiai modulių. Sudarykite nurodyto dėstytojo (įvedamas klaviatūra) modulių sąrašą.

U4-20. Leidiniai

Tekstiniame faile pateikta informacija apie leidinių prenumeratą. Eilutėje yra tokie duomenys: prenumeratoriaus pavardė, adresas, laikotarpio pradžia (sveikasis skaičius 1..12), laikotarpio ilgis (sveikasis skaičius 1..12), leidinio pavadinimas, vieno mėnesio leidinio kaina. Nustatykite kiekvienam mėnesiui, kurio leidinio pajamos yra didžiausios. Nustatykite bendrąsias leidinių pajamas. Sudarykite sąrašą leidinių, kurių pajamos mažesnės už vidutines. Spausdinamas sąrašas turi būti surikiuotas pagal leidinius abėcėlės tvarka. Sudarykite nurodyto leidinio (įvedamas klaviatūra) nurodyto mėnesio (įvedamas klaviatūra) prenumeratorių sąrašą.

U4-21. Premijos

Tekstiniame faile pateikta informacija apie mokslinių darbuotojų darbo rezultatus.. Moksliniai darbuotojai atliko darbus 4 skirtingose temose. Pirmoje eilutėje pateikti premijų dydžiai, tolesnėse eilutėse – darbuotojų pavardės, vardai, banko pavadinimas, sąskaitos numeris, darbuotojų indėliai, kurie išreikšti naudingumo koeficientu, į eilinę temą. Remiantis darbuotojų indėliais, suskaičiuokite kiekvienam darbuotojui priklausančios premijos dydį pagal kiekvieną temą atskirai ir bendrą premijų sumą. Darbuotojo indėlis rodo, kokia premijos dalis jam priklauso. Indėlis gali būti išreikštas bet kokių skaičiumi. Bet tas pats matas naudojamas visiems darbuotojams. Neišdalinkite pinigų daugiau, negu turite, ir turite išdalinti visus pinigus. Suformuokite sąrašą darbuotojų, kurie uždirbo mažiau už vidurkį. Suformuokite kiekvienam bankui atskirai pavedimų sąrašą.

U4-22. Grupės

Tekstiniame faile pateikta informacija apie studentų pasirenkamus modulius. Eilutėje yra tokie duomenys: modulio pavadinimas, atsakingo dėstytojo pavardė, vardas, kreditų kiekis, studento pavardė, vardas, grupė. Dėstytojas gali būti atsakingas už keletą modulių. Nustatykite, kurio dėstytojo modulius pasirinko daugiausiai studentų. Atspausdinkite gautą rezultatą pagal modulio kreditų kiekį ir pavadinimą. Nustatykite, ar visų grupių studentai pasirinko šio dėstytojo modulius. Atspausdinkite grupes, kurių studentai nepasirinko šio dėstytojo modulių.

U4-23. Krepšininkai

Tekstiniame faile pateikta informacija apie krepšinio komandos žaidėjus. Eilutėje yra tokia informacija apie krepšininkus: komanda, pavardė, vardas, ūgis, gimimo metai, žaidimo pozicija (puolėjas, gynėjas, centras), žaista rungtynių, įmesta taškų. Sudarykite kiekvienos pozicijos geriausiųjų žaidėjų sąrašus (taškai/rungtynių skaičius). Iš sąrašų pašalinkite žaidėjus, kurie žaidė ne daugiau kaip vienas rungtynes. Nustatykite, kelių skirtingų komandų žaidėjai užima pirmąsias tris vietas geriausiųjų žaidėjų sąrašuose. Atspausdinkite tas komandas.

U4-24. Darbininkai

Tekstiniame faile pateikta informacija apie darbininkų darbo rezultatus. Eilutėje yra tokia informacija: įmonė, data (metai, mėnuo, diena), darbininko pavardė, detalės pavadinimas, įkainis, pagamintų vienetų skaičius. Suraskite daugiausiai uždirbusio darbininko pavardę, kiek dienų jis dirbo, kiek iš viso detalių pagamino ir už kokią sumą. Sudarykite tik vieno pavadinimo detales gaminusių darbininkų sąrašą, pagamintų detalių skaičių ir sumą. Realizuokite duomenų atrinkimą į kitą rinkinį pagal nurodytą požymį (pagamintų vienetų skaičius > S, įkainis < K, įvedami klaviatūra). Spausdinamas sąrašas turi būti surikiuotas pagal pavardes abėcėlės tvarka.