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()`;
- dinaminiu 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` obelų. Kiekviena obelis apibūdinama tokiomis 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:

a) rasti kiekvienais metais kiekvienos obels užderančių obuolių kiekį, kai žinomas metų kiekis. Metų kiekis nurodomas konstanta arba įvedamas klaviatūra;

b) rasti nurodytais metais kiekvienos obels užderančių obuolių kiekį;

c) suformuoti naują sodą iš obelų, kurios per nurodytą metų kiekį sunokino ne mažesnį, 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 2 8 8 10 1 12 10 14 3 8 7 10 5 12 6 14	Informacija apie obelis Koef1 koef2 kiek prieaug 2 8 8 10 1 12 10 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.

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 prieaugis 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 prieaugis 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 `Sodas` sodo duomenims saugoti:

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

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

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

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

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

        /** Padeda į obelų objektų masyvą naują obelį ir
        // masyvo dydį padidina vienetu.
        @param ob - obels 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 - obelų 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 lentele
        @param sodas - obelų konteineris */
        //---------------------------------------------------------
        static void Spausdinti(Sodas sodas)
        {
            string virsus = "    Informacija apie obelis   \r\n"
                          + " -------------------------------- \r\n"
                          + "  Nr.  koef1  koef2  kiek  prieaug \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 . . .
```

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 lentele 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 + priaug;
            }
            Console.WriteLine(" ----------------------\r\n");
        }
//------------------------------------------------------------
```

- Pagrindinę klasę `Program` papildykite metodu `Skaičiuoti()`:

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

```
        //  derlių ekrane lentele
        @param sodas - obelų konteineris
        @param metai - metų kiekis */
        //-------------------------------------------------------------
        static void Skaičiuoti(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, kokiu būdu nurodysite metus: konstanta ar klaviatūra:

```
int metai;
Console.Write("Įveskite metų reikšmę: ");
metai = int.Parse(Console.ReadLine());
Skaičiuoti(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
 ----------------------
```

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
```

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, kokiu 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 sunokino daugiau, negu nurodytas kiekis obuolių
        @param sodas  - pirmasis obelų konteineris
        @param metai  - metų kiekis
        @param sodasN - antrasis obelų 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 konteineryje

- 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:

a) sudaryti naują sąrašą, kuriame būtų tik pirmūnai (gavo tik 10 arba 9);
b) surikiuoti pradinį sąrašą pagal pavardę ir vardą abėcėliškai;
c) 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 parametrų, metodas `Dėti()` ir metodas spausdinimui į eilutę.
- Sudaroma konteinerinė klasė `Fakultetas` fakulteto duomenims saugoti. Parašomas konstruktorius be parametrų, 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.

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 nenurodytas. 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);
            }
        }

        //-------------------------------------------------------------
        /** Spausdina konteinerio duomenis faile lentele
        @param grupe    - studentų konteineris
        @param fv       - rezultatų failo vardas
        @param antraste - užrašas virš lentelės */
        //-------------------------------------------------------------
        static void Spausdinti(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:

```
//------------------------------------------------------------
        /** 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.