

### 3. Objektų rinkinys

Susipažinsite su:

- masyvo struktūra ir masyvo tipo kintamuoju, kuriame bus saugomas objektų rinkinys;
- tekstiniais failais, kuriuose saugomi objektų duomenys ir į kuriuos surašomi skaičiavimų rezultatai;
- sumos, kiekio, aritmetinio vidurkio skaičiavimo algoritmais;
- didžiausios (mažiausios) reikšmės paieškos algoritmu;
- dviejų objektų rinkinių apjungimo į vieną algoritmu;

#### 3.1. Pažintis su objektų rinkiniu

- Masyvas objektų rinkiniui saugoti.
- Sumos, kiekio ir aritmetinio vidurkio skaičiavimo algoritmų taikymas.
- Duomenų skaitymas iš tekstinio failo.

**Užduotis.** Dviratis.

Dviračių nuomos punkte yra įvairių modelių dviračiai. Žinomi kiekvieno dviračio pagaminimo metai ir kaina.

Parašykite programą, kuri:

- suskaičiuotų, kiek yra tinkamų naudoti dviračių (ne senesni kaip  $m$  metų) ir kokia jų piniginė vertė;
- kiek yra netinkamų naudoti dviračių ir kokia jų piniginė vertė;
- koks vidutinis tinkamų naudoti ir jau netinkamų naudoti dviračių amžius.

**Pradiniai duomenys ir rezultatai.**

Pradiniai duomenys	Paaškinimai
15	Kritinis dviračio naudojimo laikas
5	Dviračių skaičius
1935; 365,10;	Dviračio pagaminimo metai ir kaina
1988; 87,09;	Dviračio pagaminimo metai ir kaina
1995; 58,10;	Dviračio pagaminimo metai ir kaina
2008; 285,58;	Dviračio pagaminimo metai ir kaina
2012; 1040,42;	Dviračio pagaminimo metai ir kaina
<b>Skaičiavimų rezultatai</b>	
Tinkami naudoti: 2 1326,00	
Netinkami naudoti: 3 510,29	
Tinkamų naudoti dviračių vidutinis amžius: 5,00	
Netinkamų naudoti dviračių vidutinis amžius: 42,33	

**Programos kūrimo eiga.**

- Sukuriama klasė dviračio duomenims saugoti.
- Parašomi klasės metodai, skirti duomenims dėti ir imti.
- Paruošiamas duomenų failas.
- Paskelbiamas objektų rinkinys.
- Parašomas duomenų skaitymo iš failo metodas, kuris suteikia rinkinio objektams reikšmes.
- Parašomas nurodyto amžiaus intervalo dviračių kiekio ir jų piniginės vertės skaičiavimo metodas.
- Parašomas nurodyto amžiaus intervalo dviračių amžiaus vidurkio skaičiavimo metodas.

**U Pirmas žingsnis.**

- Sukurkite klasę, skirtą dviračio duomenims saugoti. Jos aprašą rašykite virš pagrindinio metodo `Main()`.

```
//-----
/** Klasė dviračio duomenims saugoti
 * @class Dviratis */
class Dviratis
{
    private int metai;        // pagaminimo metai
```

```

private double kaina;    // piniginė vertė

//-----
/** Dviračio duomenys
@param metai - nauja metų reikšmė
@param kaina - nauja kainos reikšmė */
//-----
public Dviratis(int metai, double kaina)
{
    this.metai = metai;
    this.kaina = kaina;
}

/** grąžina metus */
public int ImtiMetus() { return metai; }

/** grąžina dviračio reikšmę */
public double ImtiKainą() { return kaina; }
}
//-----

```

- Patikrinkite, ar klasė aprašyta teisingai, ar teisingai dirba sukurti metodai. Tam pagrindiniame metode `Main()` parašykite darbo su klasės objektu imitaciją:

```

//-----
class Program
{
    static void Main(string[] args)
    {
        Dviratis d;
        d = new Dviratis(2012, 25.26);
        Console.WriteLine("{0} {1, 6:f2} \n", d.ImtiMetus(), d.ImtiKainą());
        Console.WriteLine("Programa baigė darbą!");
    }
}
//-----

```

- Įvykdysite programą. Ekrane turėtumėte matyti dviračio duomenis:

```

2012  25.26
Programa baigė darbą!
Press any key to continue . . .

```

## 🔗 Antras žingsnis.

- Paruoškite duomenų failą.
- Parašykite rinkinio (masyvo) maksimalų galimą dydį apibrėžiančią konstantą:

```
const int Cn = 100;
```

- Parašykite duomenų failo vardą, kuris nurodomas konstanta:

```
const string CFd = "...\\...\\Duom.txt";
```

- Pašalinkite darbo su objektu `d` veiksmus.
- Objektą `d` pakeiskite objektų rinkiniu `D[Cn]`, kur `Cn` – konstanta, reiškianti didžiausią galimą turėti dviračių skaičių.
- Parašykite du sveikojo tipo kintamuosius, skirtus rinkinio duomenų skaičiui ir dviračio naudojimo kritinio amžiaus reikšmėms saugoti.
- Parašykite duomenų skaitymo iš failo metodą.

```

//-----
/** Skaity duomenis iš failo
@param fv - failo vardas, kuris nurodomas konstanta CFd
@param D - objektų rinkinys dviračių duomenims saugoti
@param n - dviračių skaičius
@param m - kritinis dviračių naudojimo amžius */
//-----
static void Skaityti(string fv, Dviratis[] D, out int n, out int m)

```

```

    {
        int metai;
        double kaina;
        using (StreamReader reader = new StreamReader(fv))
        {
            string line;
            line = reader.ReadLine();
            string[] parts;
            m = int.Parse(line);
            line = reader.ReadLine();
            n = int.Parse(line);
            for (int i = 0; i < n; i++)
            {
                line = reader.ReadLine();
                parts = line.Split(';');
                metai = int.Parse(parts[0]);
                kaina = double.Parse(parts[1]);
                D[i] = new Dviratis(metai, kaina);
            }
        }
    }
}

```

- Papildykite pagrindinį metodą Main() kreipiniu į parašytą skaitymo metodą.

```

//-----
const int Cn = 100;
const string CFd = "...\\...\\Duum.txt";

static void Main(string[] args)
{
    Dviratis [] D = new Dviratis[Cn]; // dviračių duomenys - objektai
    int n; // dviračių skaičius
    int am; // dviračio tinkamumo naudoti kritinis amžius
    Skaityti(CFd, D, out n, out am);
    Console.WriteLine("Dviračių kiekis {0}\\n", n); // kontrolinis spausdinimas
    Console.WriteLine("Programa baigė darbą!");
}
//-----

```

- Įvykdykite programą. Jeigu ekrane matote skaičių 5, tai reiškia, kad galbūt duomenys teisingai perskaityti. Norėdami įsitikinti, ar taip tikrai yra, tikslinga kontrolinį spausdinimą papildyti sakiniais, kurie spausdintų ekrane visus dviračių rinkinio duomenis:

```

Console.WriteLine("Pagaminimo metai      Kaina");
for (int i = 0; i < n; i++)
    Console.WriteLine("      {0}      {1, 7:f2}",
        D[i].ImtiMetus(), D[i].ImtiKaina());
Console.WriteLine();

```

- Įvykdykite programą. Vėliau kontrolinio duomenų spausdinimo sakiniai nebus reikalingi, todėl juos bus galima pašalinti.

### U Trečias žingsnis.

- Parašykite metodą, skaičiuojantį nurodyto amžiaus intervalo dviračių [amPr, amPb] kiekį ir jų piniginę vertę. Toks metodas universalus, nes galima atlikti veiksmus skirtingiems amžių intervalams.

```

//-----
/** Skaičiuoja tinkamo amžiaus dviračių kainų sumą ir kiekį
 * @param D - dviračių duomenys
 * @param n - dviračių skaičius
 * @param amPr - dviračių paieškos amžiaus intervalo pradžia
 * @param amPb - dviračių paieškos amžiaus intervalo pabaiga
 * @param metai - metai, kurių atžvilgiu skaičiuojamas amžius
 * @param kiek - dviračių skaičius duotame amžiaus intervale
 * @param suma - duoto amžiaus intervalo dviračių piniginė vertė */
//-----
static void Pinigai(Dviratis[] D, int n, int amPr, int amPb, int metai,
    out int kiek, out double suma)

```

```

    {
        kiek = 0;
        suma = 0.0;
        int amžius;
        for (int i = 0; i < n; i++)
        {
            amžius = metai - D[i].ImtiMetus();
            if ((amPr <= amžius) && (amžius <= amPb))
            {
                suma = suma + D[i].ImtiKainą();
                kiek++;
            }
        }
    }
}
//-----

```

- Papildykite pagrindinį metodą Main() kreipiniu į metodą Pinigai() ir gautų rezultatų spausdinimo sakiniu:

```

int kiekTinka;
double sumaTinka;
Pinigai(D, n, 0, am, 2015, out kiekTinka, out sumaTinka);
Console.WriteLine("Tinkami naudoti: {0,3:d} {1,7:f2}", kiekTinka, sumaTinka);

```

- Įvykdykite programą ir patikrinkite gautus rezultatus.
- Papildykite programą veiksmams dėl netinkamų naudoti dviračių:

```

int kiekNetinka;
double sumaNetinka;
// Netinkami naudoti dviračiai, kurių amžius didelis,
// t.y. intervale nuo m iki begalybės (pvz., 1000 metų)
Pinigai(D, n, am + 1, 1000, 2015, out kiekNetinka, out sumaNetinka);
Console.WriteLine("Netinkami naudoti: {0} {1,7:f2}\n", kiekNetinka,
    sumaNetinka);

```

- Įvykdykite programą ir patikrinkite gautus rezultatus.

#### 🔗 Ketvirtas žingsnis.

- Parašykite metodą, skaičiuojantį nurodyto dviračių amžiaus intervalo vidurkį:

```

//-----
/** Skaiciuoja nurodyto dviračių amžiaus intervalo vidurkį
@param D - dviračių duomenys
@param n - dviračių skaičius
@param metai - metai, kurių atžvilgiu skaičiuojamas dviračio amžius
@param amPr - dviračių paieškos amžiaus intervalo pradžia
@param amPb - dviračių paieškos amžiaus intervalo pabaiga */
//-----
static double Vidurkis(Dviratis[] D, int n, int metai, int amPr, int amPb)
{
    int kiek = 0, suma = 0;
    int amžius;
    for (int i = 0; i < n; i++)
    {
        amžius = metai - D[i].ImtiMetus();
        if ((amPr <= amžius) && (amžius <= amPb))
        {
            suma = suma + amžius;
            kiek++;
        }
    }
    if (kiek > 0) return (double) suma / kiek;
    return 0.0;
}
//-----

```

- Papildykite pagrindinį metodą Main() likusiais skaičiavimais:

```

double vidurkisTinka = Vidurkis(D, n, 2015, 0, am);
if (vidurkisTinka > 0)
    Console.WriteLine("Tinkamų naudoti dviračių vidutinis amžius: {0,7:f2} ",

```

```

        vidurkisTinka );
    else Console.WriteLine("Tinkamų naudoti dviračių nėra");

    double vidurkisNetinka = Vidurkis(D, n, 2015, am + 1, 1000);
    if (vidurkisNetinka > 0)
        Console.WriteLine("Netinkamų naudoti dviračių vidutinis amžius: {0,7:f2}\n",
            vidurkisNetinka);

    else Console.WriteLine("Netinkamų naudoti dviračių nėra\n");

```

- Pasitikrinkite gautus atsakymus:

```

Dviračių kiekis 5
Pagaminimo metai      Kaina
1935                   365,10
1988                   87,09
1995                   58,10
2008                   285,58
2012                   1040,42

Tinkami naudoti: 2    1326,00
Netinkami naudoti: 3    510,29

Tinkamų naudoti dviračių vidutinis amžius:      5,00
Netinkamų naudoti dviračių vidutinis amžius:    42,33

Programa baigė darbą!
Press any key to continue . . .

```

### Programos papildymas.

- Parašykite pagrindiniame metode `Main()` sakinius, kurie suskaičiuotų ir išspausdintų visų dviračių piniginę vertę ir visų dviračių vidutinį amžių.
- Parašykite pagrindiniame metode `Main()` sakinius, kurie suskaičiuotų ir išspausdintų dviračių, kurie buvo pagaminti nurodytais (pvz., 2012) metais, piniginę vertę ir visų dviračių vidutinį amžių.
- Parašykite pagrindiniame metode `Main()` sakinius, kurie suskaičiuotų ir išspausdintų dviračių, kurie buvo pagaminti 1000 metais (tokių nėra), piniginę vertę ir visų dviračių vidutinį amžių. Ar programa pateikia teisingus rezultatus?

### Savarankiško darbo užduotis.

Žinoma, kiek kuris turistinės grupės narys turi pinigų (eurai, centai). Duomenys tekstiniame faile. Parašykite programą, kuri suskaičiuotų, kiek grupė turi pinigų iš viso ir kiek vidutiniškai jų tenka kiekvienam nariui. Kiekvienas narys bendroms grupės išlaidoms skiria ketvirtadalį turimų pinigų. Kiek pinigų bus iš viso surinkta bendroms grupės išlaidoms?

## 3.2. Du objektų rinkiniai

- Duomenys dviejuose tekstiniuose failuose.
- Veiksmai su dviem objektų rinkiniais.
- Objektų rinkinių charakteristikų radimas.

### Užduotis. Dviratis.

Dviejuose dviračių nuomos punktuose yra įvairių modelių dviračiai. Žinomi kiekvieno dviračio pagaminimo metai ir kaina.

Parašykite programą, kuri suskaičiuotų:

- kiek yra tinkamų naudoti dviračių (ne senesni kaip  $m$  metų) ir kokia jų piniginė vertė;
- kiek yra netinkamų naudoti dviračių ir kokia jų piniginė vertė;
- vidutinį tinkamų naudoti ir jau netinkamų naudoti dviračių amžių.

### Pradiniai duomenys.

Pirmojo nuomos punkto duomenys	Paaškinimai
Lėk su vėjeliu	Nuomos punkto pavadinimas
15	Kritinis dviračio naudojimo laikas
5	Dviračių skaičius
1935; 365,10;	Dviračio pagaminimo metai ir kaina

1988; 87,09;	Dviračio pagaminimo metai ir kaina
1995; 58,10;	Dviračio pagaminimo metai ir kaina
2008; 285,58;	Dviračio pagaminimo metai ir kaina
2012; 1040,42;	Dviračio pagaminimo metai ir kaina

Antrojo nuomos punkto duomenys	Paiškinimai
Neskubėk ir būsi pirmas 10 6 1955; 290,02; 1990; 100,13; 1995; 145,06; 2010; 260,94; 2012; 1040,42; 2012; 435,07;	Nuomos punkto pavadinimas Kritinis dviračio naudojimo laikas Dviračių skaičius Dviračio pagaminimo metai ir kaina Dviračio pagaminimo metai ir kaina Dviračio pagaminimo metai ir kaina Dviračio pagaminimo metai ir kaina Dviračio pagaminimo metai ir kaina Dviračio pagaminimo metai ir kaina

### Programos kūrimo eiga.

- Nukopijuojama ankstesnio pratimo programa.
- Paruošiamas antras duomenų failas.
- Programa papildoma dviem objektų rinkiniais ir veiksmams su jais.

### U Pirmas žingsnis.

- Nukopijuokite ankstesnio pratimo programą.
- Papildykite programą antru duomenų rinkiniu.
- Duomenų rinkinio pavadinimą D pakeiskite pavadinimu D1 bei kitus kintamųjų pavadinimus. Papildykite programą kintamaisiais, skirtais nuomos punkto pavadinimui saugoti ir antruoju objektų rinkiniu:

```
// Pirmojo dviračių nuomos punkto
Dviratis [] D1 = new Dviratis[Cn]; // dviračių duomenys
int n1; // dviračių skaičius
int am1; // dviračio tinkamumo naudoti kritinis amžius
string pav1; // nuomos punkto pavadinimas
int kiekTinka1, kiekNetinka1;
double sumaTinka1, sumaNetinka1;

// Antrojo dviračių nuomos punkto
Dviratis [] D2 = new Dviratis[Cn]; // dviračių duomenys
int n2; // dviračių skaičius
int am2; // dviračio tinkamumo naudoti kritinis amžius
string pav2; // nuomos punkto pavadinimas
int kiekTinka2, kiekNetinka2;
double sumaTinka2, sumaNetinka2;
```

- Papildykite duomenų skaitymo metodą parametrais, skirtais failo vardui perduoti ir grąžinti perskaitytam nuomos punkto pavadinimui:

```
//-----
/** Skaity duomenis iš failo
@param fv - failo vardas
@param D - objektų rinkinys dviračių duomenims saugoti
@param n - dviračių skaičius
@param m - kritinis dviračių naudojimo amžius
@param pav - nuomos punkto pavadinimas */
//-----
static void Skaityti(string fv, Dviratis[] D, out int n, out int m, out string pav)
{
    int metai;
    double kaina;
    using (StreamReader reader = new StreamReader(fv))
    {
        string line;
        line = reader.ReadLine();
        string[] parts;
        pav = line;
    }
}
```

```

        line = reader.ReadLine();
        m = int.Parse(line);
        line = reader.ReadLine();
        n = int.Parse(line);
        for (int i = 0; i < n; i++)
        {
            line = reader.ReadLine();
            parts = line.Split(';');
            metai = int.Parse(parts[0]);
            kaina = double.Parse(parts[1]);
            D[i] = new Dviratis(metai, kaina);
        }
    }
}
//-----

```

- Pakeiskite ir papildykite failų vardų konstantų sąrašą:

```

const string CFd1 = "...\\...\\Duum1.txt";
const string CFd2 = "...\\...\\Duum2.txt";

```

- Parašykite kreipinį į duomenų skaitymo metodą pirmo nuomos punkto duomenims skaityti:  
Skaityti(CFd1, D1, out n1, out am1, out pav1);

- Patikrinkite, ar pirmojo dviračių nuomos punkto duomenys perskaityti teisingai.

### 🔹 Antras žingsnis.

- Papildykite programą sakiniais, skirtais antrojo nuomos punkto duomenims skaityti. Patikrinkite, ar antrojo dviračių nuomos punkto duomenys perskaityti teisingai.

### 🔹 Trečias žingsnis.

- Papildykite programą sakiniais, skaičiuojančiais, kiek abiejuose nuomos punktuose yra tinkamų naudoti dviračių ir kokia jų piniginė vertė bei vidutinis amžius. Spausdinkite gautus rezultatus.
- Patikrinkite, ar gauti rezultatai teisingi.
- Papildykite programą sakiniais, skaičiuojančiais, kiek abiejuose nuomos punktuose yra netinkamų naudoti dviračių ir kokia jų piniginė vertė bei vidutinis amžius. Spausdinkite gautus rezultatus.
- Patikrinkite, ar gauti rezultatai teisingi.

### 🔹 Ketvirtas žingsnis.

- Parašykite sakinius, kuriais nustatoma, kuriame punkte yra daugiau tinkamų naudoti dviračių.
- Parašykite sakinius, kuriais nustatoma, kurio punkto turimų dviračių piniginė vertė yra didesnė.

### Savarankiško darbo užduotis.

Barboros turimos valiutos surašytos faile B.txt, o Anupro – faile A.txt. Vienoje eilutėje – vienos šalies valiutos duomenys: pinigai (pvz., doleriai ir centai) ir kursas, perskaičiuojant į eurus ir centus. Parašykite programą, kuri suskaičiuotų, kiek pinigų eurai ir centais turi Barbora ir kiek Anupras atskirai ir kiek abu bendrai.

## 3.3. Struktūriniai duomenys.

- Didžiausios (mažiausios) reikšmės paieška.
- Naujo objektų rinkinio formavimas.

### Užduotis. Dviratis.

Dviejuose dviračių nuomos punktuose yra įvairių modelių dviračiai. Žinomas kiekvieno dviračio modelis, kiekis, pagaminimo metai ir kaina.

Parašykite programą, kuri:

- surastų, kuriame dviračių nuomos punkte yra seniausias dviratis;
- surastų, kuriame dviračių nuomos punkte yra brangiausias dviratis ir kokia jo piniginė vertė;
- suformuotų dviračių modelių, kurie yra tuo pačiu metu abiejuose nuomos punktuose, rinkinį.

**Pradiniai duomenys.**

Pirmojo nuomos punkto duomenys	Paaiškinimai
Lėk su vėjeliu 5	Nuomos punkto pavadinimas
ORION;                   2; 1935;       365,1;	Įrašų skaičius
POLARIS;               1; 1988;       87,09;	Dviračio modelis, kiekis, pagaminimo metai ir kaina
INDUSTRIERAD; 10; 1995;   58,10;	Dviračio modelis, kiekis, pagaminimo metai ir kaina
Bauer;                 25; 2008;     285,58;	Dviračio modelis, kiekis, pagaminimo metai ir kaina
HERREN;               15; 2012;   1040,42;	Dviračio modelis, kiekis, pagaminimo metai ir kaina

Antrojo nuomos punkto duomenys	Paaiškinimai
Neskubėk ir būsi pirmas 6	Nuomos punkto pavadinimas
VOLTA;                 1; 1955;     290;02;	Įrašų skaičius
POLARIS;               2; 1990;     100;13;	Dviračio modelis, kiekis, pagaminimo metai ir kaina
BANANA;               3; 1995;     145;06;	Dviračio modelis, kiekis, pagaminimo metai ir kaina
INDUSTRIERAD; 10; 2010;   260;94;	Dviračio modelis, kiekis, pagaminimo metai ir kaina
ORION;                 15; 2012;   1040;42;	Dviračio modelis, kiekis, pagaminimo metai ir kaina
POLARIS;               20; 2012;   945;07;	Dviračio modelis, kiekis, pagaminimo metai ir kaina

**Programos kūrimo eiga.**

- Ankstesnio pratimo programos teksto kopija modifikuojama darbui su dviračių rinkiniais, kai žinomi dviračių modelių pavadinimai.
- Parašomas objektų rinkinio duomenims spausdinti faile metodas.
- Parašomas metodas seniausiam dviračiui rasti nurodytame objektų rinkinyje.
- Parašomas metodas brangiausiam dviračiui rasti nurodytame objektų rinkinyje.
- Parašomas metodas, skirtas rasti nurodyto modelio dviračiui nurodytame objektų rinkinyje.
- Parašomi metodai, skirti formuoti bendram objektų sąrašui iš turimų dviejų.

**➊ Pirmas žingsnis.**

- Paruoškite duomenų failus, suteikdami jiems vardus Nuoma1.txt ir Nuoma2.txt.
- Papildykite duomenų failus duomenimis pagal pateiktą pavyzdį.
- Nukopijuokite ankstesnio pratimo programos tekstą ir išsaugokite nauju vardu.
- Pašalinkite visus metodus, išskyrus duomenų skaitymo iš failo. Iš pagrindinio metodo Main() pašalinkite kreipinius į pašalintus metodus.
- Papildykite klasę Dviratis kintamaisiais, skirtais modelio pavadinimui ir dviračių kiekiui saugoti:

```
private int kiek;
private string pav;
```

- Klasės konstruktorių Dviratis() papildykite parametrais, skirtais dviračio modelio vardui ir kiekiui dėti. Konstruktoriaus antraštė:

```
public Dviratis(string pav, int kiek, int metai, double kaina );
```

- Papildykite klasę metodais, grąžinančiais dviračio modelio pavadinimą ir jų kiekį:

```
public string ImtiPavadinima() { return pav; }
public int ImtiKiek() { return kiek; }
```

- Įvykdykite programą. Jeigu nėra klaidų, programa sėkmingai baigia darbą nieko neskaičiuodama.

**➋ Antras žingsnis.**

- Modifikuokite duomenų skaitymo iš failo metodą:

```
//-----
/** Skaity duomenis iš failo
 * @param Fd - failo vardas
 * @param D - objektų rinkinys dviračių duomenims saugoti
 * @param n - dviračių skaičius
 * @param pav - nuomos punkto pavadinimas */
//-----
static void Skaityti(string Fd, Dviratis[] D, out int n, out string pav)
{
```



```

using (StreamReader reader = new StreamReader(Fd))
{
    string eil; int kiekn; int metain; double kainan;
    string line;
    line = reader.ReadLine();
    string[] parts;
    pav = line;
    line = reader.ReadLine();
    n = int.Parse(line);
    for (int i = 0; i < n; i++)
    {
        line = reader.ReadLine();
        parts = line.Split(';');
        eil = parts[0];
        kiekn = int.Parse(parts[1]);
        metain = int.Parse(parts[2]);
        kainan = double.Parse(parts[3]);
        D[i] = new Dviratis(eil, kiekn, metain, kainan);
    }
}
}
//-----

```

- Parašykite pagrindinio metodo Main() duomenų skaitymo sakinius, patikrinkite, ar gerai skaitomi duomenys.

```

//-----
const string CFd1 = "...\\...\\Nuoma1.txt";
const string CFd2 = "...\\...\\Nuoma2.txt";
//-----
static void Main(string[] args)
{
    // Pirmojo dviračių nuomos punkto
    Dviratis [] D1 = new Dviratis[Cn]; // dviračių duomenys
    int n1; // dviračių skaičius
    string pav1; // nuomos punkto pavadinimas

    // Antrojo dviračių nuomos punkto
    Dviratis [] D2 = new Dviratis[Cn]; // dviračių duomenys
    int n2; // dviračių skaičius
    string pav2; // nuomos punkto pavadinimas

    Skaityti(CFd1, D1, out n1, out pav1);
    Skaityti(CFd2, D2, out n2, out pav2);

    // -----
    // I nuomos punktas, kontrolinis spausdinimas
    Console.WriteLine("Pirmas nuomos punktas: {0}", pav1);
    Console.WriteLine("Dviračių kiekis {0}\n", n1);
    Console.WriteLine("Pavadinimas Kiekis Pagaminimo metai Kaina");
    for (int i = 0; i < n1; i++)
        Console.WriteLine("{0,-12} {1,4:d} {2,3:d} {3, 7:f2}",
            D1[i].ImtiPavadinimą(), D1[i].ImtiKiekį(),
            D1[i].ImtiMetus(), D1[i].ImtiKainą());
    Console.WriteLine();

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

```

### ☛ Trečias žingsnis.

- Parašykite objektų rinkinio spausdinimo faile metodą:

```

//-----
/** Spausdina duomenis į failą
@param fv - rezultatų failo vardas
@param D - objektų rinkinys dviračių duomenims saugoti
@param kiek - dviračių skaičius
@param pav - nuomos firmos pavadinimas */

```

```
//-----
static void SpausdintiDuomenis(string fv, Dviratis[] D, int nkiek, string pav)
{
    const string virsus =
        "|-----|-----|-----|-----|\r\n"
        + "    Pavadinimas    |    Kiekis    |    Pagaminimo    |    Kaina    |\r\n"
        + "    |                |                |    metai        |    (eurų)   |\r\n"
        + "    |-----|-----|-----|-----|";
    using (var fr = File.AppendText(fv))
    {
        fr.WriteLine("Nuomos firma: {0}", pav);
        fr.WriteLine(virsus);
        Dviratis tarp;
        for (int i = 0; i < nkiek; i++)
        {
            tarp = D[i];
            fr.WriteLine("| {0,-15} | {1,8} | {2,5:d} | {3,7:F2} |",
                tarp.ImtiPavadinimą(), tarp.ImtiKieki(),
                tarp.ImtiMetus(), tarp.ImtiKainą());
        }
        fr.WriteLine("-----");
    }
}
//-----
```

- Papildykite failų vardų konstantų sąrašą rezultatų failo vardo CFrez konstanta.
- Parašykite pagrindiniame metode Main() egzistuojančio rezultatų failo išmetimo sakinį:

```
if (File.Exists(CFrez))
    File.Delete(CFrez);
```

- Papildykite pagrindinį metodą Main() krepiniiais į metodą Spausdinti(), kuriais būtų išspausdinami nuomos punktų duomenys. Patikrinkite, ar rezultatų faile gavote tokias lenteles:

Nuomos firma: Lėk su vėjeliu

Pavadinimas	Kiekis	Pagaminimo metai	Kaina (eurų)
ORION	2	1935	365,10
POLARIS	1	1988	87,09
INDUSTRIERAD	10	1995	58,10
Bauer	25	2008	285,58
HERREN	15	2012	1040,42

Nuomos firma: Neskubėk ir būsi pirmas

Pavadinimas	Kiekis	Pagaminimo metai	Kaina (eurų)
VOLTA	1	1955	290,00
POLARIS	2	1990	100,00
BANANA	3	1995	145,00
INDUSTRIERAD	10	2010	260,00
ORION	15	2012	1040,00
POLARIS	20	2012	945,00

#### 🕒 Ketvirtas žingsnis.

- Sukurkite metodą, kuris duotame objektų rinkinyje surastų seniausio dviračio indeksą:

```
//-----
/** Gražina dviračio, kurio metų skaičius yra mažiausias, indeksą
@param D - objektų rinkinys
@param n - objektų skaičius rinkinyje */
```

```
//-----
static int Seniausias(Dviratis [] D, int n)
{
    int k = 0;
    for (int i = 0; i < n; i++)
        if (D[i].ImtiMetus() < D[k].ImtiMetus())
            k = i;
    return k;
}
//-----
```

- Papildykite pagrindinį metodą Main() veiksmams, kuriais būtų surandamas nuomos punktas, kuriame yra seniausias dviratis:

```
using (var fr = File.AppendText(CFrez))
{
    if (D1[Seniausias(D1, n1)].ImtiMetus() < D2[Seniausias(D2, n2)].ImtiMetus())
        fr.WriteLine("Seniausias dviratis yra nuomos punkte {0} ", pav1);
    else fr.WriteLine("Seniausias dviratis yra nuomos punkte {0} ", pav2);
}
```

Galima parašyti kitaip:

```
int ind1 = Seniausias(D1, n1);
int ind2 = Seniausias(D2, n2);
using (var fr = File.AppendText(CFrez))
{
    if (D1[ind1].ImtiMetus() < D2[ind2].ImtiMetus())
        fr.WriteLine("Seniausias dviratis yra nuomos punkte {0} ", pav1);
    else fr.WriteLine("Seniausias dviratis yra nuomos punkte {0} ", pav2);
}
```

- Patikrinkite, ar programa dirba teisingai.
- Pakeiskite duomenis taip, kad seniausias dviratis būtų kitame nuomos punkte. Patikrinkite, ar programa dirba teisingai.
- Pakeiskite duomenis taip, kad abiejuose punktuose yra vienodi seniausi dviračiai. Kurio punkto pavadinimą matote? Patikrinkite programą taip, kad šiuo atveju programa ekrane rodytų abiejų punktų pavadinimus.
- Patikrinkite, kaip programa dirba, kai punkte / punktuose yra keli seniausi dviračiai.

## 🔗 Penktas žingsnis.

- Sukurkite metodą, kuris nustatytų, ar objektų sąrašas yra nurodyto modelio dviratis:

```
//-----
/** Gražina surasto dviračio masyve indeksą arba -1, jeigu dviračio masyve nėra
@param D - objektų rinkinys
@param n - objektų skaičius rinkinyje
@param pav - ieškomo modelio pavadinimas */
//-----
static int YraModelis(Dviratis [] D, int n, string pav, int metai)
{
    for (int i = 0; i < n; i++)
        if (D[i].ImtiPavadinimą() == pav && D[i].ImtiMetus() == metai) return i;
    return -1;
}
//-----
```

- Papildykite klasę metodu

```
public void PapildytiKiekį(int k) { kiek = kiek + k; }
```

- Sukurkite metodą, kuris dviračių modelių sąrašą papildytų dviračiais iš duoto objektų rinkinio. Įrašymo metu atsižvelgiama į pagaminimo metus:

```
//-----
/** Objektų rinkinį papildo duomenimis iš kito objektų rinkinio
// Jeigu objektų rinkinio D tokio pat modelio dviratis yra objektų rinkinyje Dr,
// tuomet didinamas kiekis, kitaip - papildomas nauju objektu
@param n - objektų skaičius rinkinyje D
@param D - objektų rinkinys, iš kurio pildo
```

```

@param nr - objektų skaičius rinkinyje Dr
@param Dr - objektų rinkinys, kurį pildo */
//-----
static void Formuoti(Dviratis [] D, int n, Dviratis [] Dr, ref int nr)
{
    int k;
    for (int i = 0; i < n; i++)
    {
        k = YraModelis(Dr, nr, D[i].ImtiPavadinimą(), D[i].ImtiMetus());
        if (k >= 0)
            Dr[k].PapildytiKiekį(D[i].ImtiKiekį()); // didinamas kiekis

        else
        {
            Dr[nr] = D[i]; // papildomas rinkinys
            nr++;
        }
    }
}
//-----

```

- Papildykite pagrindinį metodą Main() veiksmams, kuriais formuojamas abiejuose nuomos punktuose esančių dviračių modelių rinkinys.

```

Dviratis[] Dr = new Dviratis[Cn];
int nr;
nr = 0;
Formuoti(D1, n1, Dr, ref nr);
Formuoti(D2, n2, Dr, ref nr);
SpausdintiDuomenis(CFrez, Dr, nr, "Modelių sąrašas");

```

- Patikrinkite, ar rezultatų faile išspausdinta teisinga modelių lentelė:

Modelių sąrašas

Pavadinimas	Kiekis	Pagaminimo metai	Kaina (eurų)
ORION	2	1935	365,10
POLARIS	1	1988	87,09
INDUSTRIERAD	10	1995	58,10
Bauer	25	2008	285,58
HERREN	15	2012	1040,42
VOLTA	1	1955	290,00
POLARIS	2	1990	100,00
BANANA	3	1995	145,00
INDUSTRIERAD	10	2010	260,00
ORION	15	2012	1040,00
POLARIS	20	2012	945,00

#### Savarankiško darbo užduotis.

- Sukurkite metodą, kuris surastų duotame objektų rinkinyje brangiausią dviratį.
- Papildykite programą veiksmams, kuriais būtų randama, kuriame dviračių nuomos punkte yra brangiausias dviratis ir kokia jo piniginė vertė.

### 3.4. Simbolių eilutės keli žodžiai.

- Objektų rinkinys.
- Reikšmių įvedimas iš failo.
- Reikšmių spausdinimas į failą lentele.
- Metodai.

**Užduotis.** Firmos automobiliai.

Failo Duomenys.txt pirmoje eilutėje yra vienos UAB automobilių kiekis, kitose eilutėse – automobilių duomenys: pavadinimas, degalų tipas, degalų sąnaudos 100 km.

Sukurkite klasę automobilio duomenims saugoti. Į šios klasės objektų masyvą perskaitykite duomenis iš failo Duomenys.txt. Masyvo objektų reikšmės spausdinkite lentelę į failą Rezultatai.txt.

**Pradiniai duomenys ir rezultatai.**

Pradiniai duomenys faile Duomenys.txt		
8 Škoda Fabia;benzinas; 7,1; Opel Meriva;benzinas; 7,5; Volkswagen Golf;dyzelinas; 6,3; Opel Astra;dyzelinas; 6,1; Toyota Corolla;benzinas; 6,8; Toyota Yaris;benzinas; 5,9; Peugeot 207;dyzelinas; 5,5; Škoda Octavia;dyzelinas; 7,3;		
Rezultatai faile Rezultatai.txt		
Automobilių kiekis 8 Automobilių sąrašas:		
-----	-----	-----
Pavadinimas	Degalai	Sąnaudos (l/100 km)
-----	-----	-----
Škoda Fabia	benzinas	7,10
Opel Meriva	benzinas	7,50
Volkswagen Golf	dyzelinas	6,30
Opel Astra	dyzelinas	6,10
Toyota Corolla	benzinas	6,80
Toyota Yaris	benzinas	5,90
Peugeot 207	dyzelinas	5,50
Škoda Octavia	dyzelinas	7,30
-----	-----	-----

**Programos kūrimo eiga.**

- Sukuriama klasė vieno automobilio duomenims saugoti. Parašomi klasės sąsajos metodai.
- Pagrindiniame metode Main() skelbiamas automobilio klasės objektų masyvas.
- Parašomas metodas duomenims iš failo įvesti į objektų masyvą.
- Paruošiamas duomenų failas "Duomenys.txt".
- Parašomas metodas objektų masyvo reikšmėms spausdinti lentelę į failą.
- Pagrindinis metodas papildomas kreipiniais į parašytus metodus.

**U Pirmas žingsnis.**

- Sukurkite klasę Auto ir jos sąsajos metodus:

```
//-----
/** Klasė automobilio duomenims saugoti
@class Auto */
class Auto
{
    private string pav,           // automobilio pavadinimas
                degalai;         // degalų tipas
    private double sąnaudos;      // kuro sąnaudos 100 km

    //-----
    /** Automobilio duomenys
    @param pav - nauja pavadinimo reikšmė
    @param degalai - nauja degalų tipo reikšmė
    @param sąnaudos - nauja sąnaudų reikšmė */
    //-----
    public Auto(string pav, string degalai, double sąnaudos)
    {
```

```

        this.pav = pav;           // automobilio pavadinimas
        this.degalai = degalai;   // automobilio pavadinimas
        this.snaudos = snaudos;  // kuro snaudos 100 km
    }

    /** grąžina automobilio pavadinimas */
    public string ImtiPav() { return pav; }

    /** grąžina degalų tipą */
    public string ImtiDegalus() { return degalai; }

    /** grąžina degalų snaudas */
    public double ImtiSnaudos() { return snaudos; }
}
//-----

```

- Pagrindiniame metode `Main()` aprašykite klasės `Auto` objektų rinkinį `A` ir kintamąjį, kuriame bus saugomas UAB automobilių kiekis.

```

//-----
static void Main(string[] args)
{
    Auto [] A = new Auto[100];    // automobilių duomenys
    int na;                       // automobilių kiekis

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

```

### ➤ Antras žingsnis.

- Parašykite duomenų įvedimo metodą.

```

//-----
/** Skaityti duomenis iš failo
@param Fd      - failo vardas
@param A       - objektų rinkinys automobilių duomenims saugoti
@param kiek    - dviračių skaičius */
//-----
static void Skaityti(string Fd, Auto[] A, out int kiek)
{
    using (StreamReader reader = new StreamReader(Fd))
    {
        string pav, degalai; double snaudos;
        string line;
        line = reader.ReadLine();
        string[] parts;
        kiek = int.Parse(line);
        for (int i = 0; i < kiek; i++)
        {
            line = reader.ReadLine();
            parts = line.Split(';');
            pav = parts[0];
            degalai = parts[1];
            snaudos = double.Parse(parts[2]);
            A[i] = new Auto(pav, degalai, snaudos);
        }
    }
}
//-----

```

Naudodamiesi pavyzdžiu paruoškite duomenų failą `Duomenys.txt`.

### ➤ Trečias žingsnis.

- Parašykite objektų masyvo reikšmių spausdinimo metodą.

```

//-----
/** Spausdina duomenis į failą
@param fv      - rezultatu failo vardas
@param A       - objektų rinkinys automobilių duomenims saugoti
@param nkiek   - automobilių skaičius */

```

```
//-----
static void Spausdinti(string fv, Auto[] A, int nkiek)
{
    const string virsus =
        "|-----|-----|-----|\r\n"
        + "|               |         |          |\r\n"
        + "|   Pavadinimas   |   Degalai   | Snaudos (l/100 km) |\r\n"
        + "|               |         |          |\r\n"
        + "|-----|-----|-----|";
    using (var fr = File.AppendText(fv))
    {
        fr.WriteLine("Automobilių kiekis {0}", nkiek);
        fr.WriteLine("Automobilių sąrašas:");
        fr.WriteLine(virsus);
        Auto tarp;
        for (int i = 0; i < nkiek; i++)
        {
            tarp = A[i];
            fr.WriteLine("| {0,-15} | {1,-9} | {2,8:f2} |",
                tarp.ImtiPav(), tarp.ImtiDegalus(),
                tarp.ImtiSnaudos());
        }
        fr.WriteLine("-----");
    }
}
//-----
```

#### 🔗 Ketvirtas žingsnis.

- Papildykite pagrindinį metodą Main() esančio rezultatų failo ištrynimo veiksmu.

```
if (File.Exists(CFrez))
    File.Delete(CFrez);
```

- Papildykite pagrindinį metodą Main() kreipiniais į įvedimo ir spausdinimo metodus.

```
Skaityti(CFd, A, out na);
Spausdinti(CFrez, A, na);
```

- Įvykdysite programą. Rezultatų faile "Rezultatai.txt" matysite:

```
Automobilių kiekis 8
Automobilių sąrašas:
```

Pavadinimas	Degalai	Snaudos (l/100 km)
Škoda Fabia	benzinas	7,10
Opel Meriva	benzinas	7,50
Volkswagen Golf	dyzelinas	6,30
Opel Astra	dyzelinas	6,10
Toyota Corolla	benzinas	6,80
Toyota Yaris	benzinas	5,90
Peugeot 207	dyzelinas	5,50
Škoda Octavia	dyzelinas	7,30

#### Programos papildymas.

Pakeiskite programą taip, kad automobilių sąrašo lentelėje būtų spausdinami eilučių numeriai.

**Savarankiško darbo užduotis.**

Yra žinomi  $n$  ( $5 < n < 100$ ) Lietuvos kelių duomenys (pavadinimas, ilgis, leistinas greitis). Surašykite šiuos duomenis į failą. Parašykite programą, kuri įvestų duomenis iš failo į objektų masyvą ir spausdintų lentelę į kitą failą.

**3.5. Veiksmai su objektais rinkinyje.**

- Sumos, sandaugos, kiekio skaičiavimas.

**Užduotis.** Firmos automobilių degalų sąnaudos.

Failo "Duomenys.txt" pirmoje eilutėje yra vienos UAB automobilių kiekis, kitose eilutėse yra automobilių duomenys: pavadinimas, degalų tipas, degalų sąnaudos 100 km.

Sukurkite klasę automobilio duomenims saugoti. Į šios klasės objektų masyvą perskaitykite duomenis iš failo "Duomenys.txt".

Kiek degalų 100 km vidutiniškai sunaudoja vienas automobilis?

Duomenis ir rezultatus spausdinkite į failą "Rezultatai.txt".

**Pradiniai duomenys ir rezultatai.**

Pradiniai duomenys faile "Duomenys.txt"	
8	
Škoda Fabia;benzinas; 7,1;	
Opel Meriva;benzinas; 7,5;	
Volkswagen Golf;dyzelinas; 6,3;	
Opel Astra;dyzelinas; 6,1;	
Toyota Corolla;benzinas; 6,8;	
Toyota Yaris;benzinas; 5,9;	
Peugeot 207;dyzelinas; 5,5;	
Škoda Octavia;dyzelinas; 7,3;	

Rezultatai faile "Rezultatai.txt"			
Automobilių kiekis 8			
Automobilių sąrašas:			
	Pavadinimas	Degalai	Sąnaudos (l/100 km)
	Škoda Fabia	benzinas	7,10
	Opel Meriva	benzinas	7,50
	Volkswagen Golf	dyzelinas	6,30
	Opel Astra	dyzelinas	6,10
	Toyota Corolla	benzinas	6,80
	Toyota Yaris	benzinas	5,90
	Peugeot 207	dyzelinas	5,50
	Škoda Octavia	dyzelinas	7,30
Vidutinės degalų sąnaudos: 6,56 litro/100 km			

**Programos kūrimo eiga.**

- Ankstesnio pratimo programa kopijuojama į atskirą katalogą.
- Parašomas metodas klasės Auto objektų masyvo lauko sąnaudos reikšmių vidurkiui skaičiuoti.
- Pagrindinis metodas Main() papildomas parašyto metodo grąžintos reikšmės spausdinimu.

**➊ Pirmas žingsnis.**

- Nukopijuokite ankstesnio pratimo programą kartu su klase Auto ir duomenų failu Duomenys.txt.

**➋ Antras žingsnis.**

- Parašykite vidurkio skaičiavimo metodą.

```
//-----
/** Skaičiuoja ir grąžina objektų masyvo lauko sąnaudos vidutinę reikšmę
@param A - objektų rinkinys automobilių duomenims saugoti
```



```

@param kiek - automobilių skaičius */
//-----
static double VidSnaudos(Auto [] A, int kiek)
{
    double sum = 0;
    for (int i=0; i<kiek; i++)
    {
        sum = sum + A[i].ImtiSnaudos();
    }
    return sum / kiek;
}
//-----

```

### U Trečias žingsnis.

- Papildykite pagrindinį metodą Main() parašyto metodo grąžintos reikšmės spausdinimu.

```

using (var fr = File.AppendText(CFrez))
{
    fr.WriteLine("Vidutinės degalų sąnaudos: {0,7:f2} litro/100 km" ,
        VidSnaudos(A, na));
}

```

- Įvykdykite programą. Rezultatų faile "Rezultatai.txt" matysite:

Automobilių kiekis 8

Automobilių sąrašas:

Pavadinimas	Degalai	Sąnaudos (l/100 km)
Škoda Fabia	benzinas	7,10
Opel Meriva	benzinas	7,50
Volkswagen Golf	dyzelinas	6,30
Opel Astra	dyzelinas	6,10
Toyota Corolla	benzinas	6,80
Toyota Yaris	benzinas	5,90
Peugeot 207	dyzelinas	5,50
Škoda Octavia	dyzelinas	7,30

Vidutinės degalų sąnaudos: 6,56 litro/100 km

### Programos papildymas.

Parašykite metodą dyzelinių automobilių kiekiui rasti. Papildykite programą šio kiekio radimu ir spausdinimu į rezultatų failą.

### Savarankiško darbo užduotis.

Modifikuokite paskutinio pratimo programą, kuri papildomai rastų:

- kiek degalų 100 km vidutiniškai sunaudoja vienas benzininis automobilis;
- įvertintų situaciją, kad gali nebūti nė vieno benzininio automobilio.

## 3.6. Kontroliniai klausimai

- Kiek ir kokių kintamųjų reikia, norint aprašyti duomenų masyvą arba objektų rinkinį?
- Kuo skiriasi reikšmių sumos radimo algoritmas nuo aritmetinio vidurkio skaičiavimo algoritmo ir kas juose yra bendra?
- Kokia turi būti pradinė kintamojo, skirto didžiausiai reikšmei masyve (rinkinyje) rasti, reikšmė?
- Kuo skiriasi mažiausios reikšmės radimo algoritmas nuo algoritmo, skirto rasti mažiausią reikšmę turintį objektą rinkinyje?
- Kaip surasti, kiek objektų rinkinyje yra objektų, turinčių vienodą didžiausią reikšmę?
- Duotas masyvo aprašas: `double[] AA = new double[35];`  
Kuris masyvo elemento panaudojimo atvejų yra teisingas?

- a. `double x = A[0]+3.5;`
  - b. `A[3.5] = 2.5;`
  - c. `A[0] = A[35]+2;`
  - d. `A[7] = "2.5";`
7. Duotas masyvo aprašas: `double[] AA = new double[35]`. Kintamajame `S` reikia sukaupti teigiamų masyvo `A` elementų sumą.  
Kokia turi būti kintamojo `S` pradinė reikšmė?
- a. 0
  - b. -1
  - c. bet koks teigiamas skaičius
  - d. pradinės reikšmės priskirti nereikia
8. Duotas masyvo aprašas: `double[] AA = new double[35]`. Reikia rasti masyvo `A` didžiausio elemento indeksą.  
Kokia turi būti kintamojo, kuriame saugosime ieškomą indeksą, pradinė reikšmė?
- a. bet kurio elemento indeksas
  - b. bet kurio elemento reikšmė
  - c. bet koks teigiamas skaičius
  - d. bet koks neigiamas skaičius
9. Ką matysime ekrane, atlikus tokius veiksmus:
- ```
static int Suma(int [] A, int pr, int ga, int h)
{
    int s = 0;
    for (int i = pr; i < ga; s = s + A[i], i = i + h);
    return s;
}

int [] D = new int[8] { 5, 4, -3, 8, 0, 0, 0, 0 };
int [] B = new int[8] { 6, 5, -9, -4, 3, 0, 0, 0 };
Console.WriteLine("{0,4:d} {1,4:d}",
    Suma(D, 0, 4, 2), Suma(B, 1, 5, 2));
```
- 2       1
  - 14       1
  - 2       0
  - Programos fragmente yra klaidų.

### 3.7. Užduotys

#### U3–1. Sodas

- Sode auga daug skirtingų vaismedžių. Tekstiniame duomenų faile yra vaismedžio pavadinimas, jo aukštis ir amžius. Pirmoje failo eilutėje yra sodo savininko pavardė ir vardas. Sukurkite klasę `Vaismedis`, kuri turėtų kintamuosius pavadinimui, amžiui ir aukščiui saugoti. Raskite, koks aukščiausio vaismedžio aukštis ir koks seniausio vaismedžio amžius.
- Papildykite programą veiksmiais su dviem sodais. Kiekvieno sodo duomenys saugomi atskiruose failuose. Kuriame sode auga aukščiausias vaismedis? Surašykite į atskirą rinkinį visus abiejų sodų vaismedžius, kurių pavadinimas sutampa su aukščiausio vaismedžio pavadinimu

#### U3–2. Krepšinis

- Krepšinio mokykloje treniruotes lankančių sąrašas yra tekstiniame faile: būsimio krepšininko vardas ir pavardė, amžius ir ūgis. Pirmoje eilutėje yra krepšinio mokyklos pavadinimas. Sukurkite klasę `Krepšininkas`, kuri turėtų kintamuosius vardui su pavarde, amžiui bei ūgiui saugoti. Raskite, koks būsimų krepšininkų amžiaus vidurkis ir koks ūgio vidurkis.
- Papildykite programą veiksmiais su dviejų krepšinio mokyklų duomenimis. Kiekvienos mokyklos duomenys saugomi atskiruose failuose. Kurioje mokykloje aukščiausias sportininkas? Surašykite į atskirą rinkinį visus abiejų mokyklų sportininkus, kurių ūgis didesnis už vidurkį.

#### U3–3. Studentas

- Gūdžiame miške gyvena pabaisos. Duomenys apie jas yra tekstiniame faile: pavadinimas, ragu skaičius, uodegų skaičius. Pirmoje eilutėje yra studento, kuris tyrinėja tas pabaisas, vardas ir pavardė. Antroje eilutėje yra miško pavadinimas. Sukurkite klasę `Pabaisa`, kuri turėtų kintamuosius

pavadinimui, ragų ir uodegų kiekiui saugoti. Kiek iš viso miške yra ragų ir uodegų? Kiek ragų turi mažiausiai uodeguota pabaisa?

- Papildykite programą veiksmams su dviejų miškų duomenimis. Kiekvieno miško duomenys saugomi atskiruose failuose. Kurio studento miške yra pabaisa, turinti daugiausiai ragų? Surašykite į atskirą rinkinį visus abiejų miškų pabaisų, kurių ragų skaičius didesnis už turimų uodegų skaičių, duomenis.

#### U3–4. Monetos

- Monetų kolekcininko turimų monetų duomenys surašyti faile: monetos kilmės šalis, nominalas ir svoris. Pirmoje eilutėje yra kolekcininko vardas ir pavardė. Sukurkite klasę *Moneta*, kuri turėtų kintamuosius kilmės šalies pavadinimui, nominalui ir svoriui saugoti. Kuri moneta sunkiausia ir kokia visų monetų bendra piniginė vertė?
- Papildykite programą veiksmams su dviejų kolekcijų duomenimis. Kiekvienos kolekcijos duomenys saugomi atskiruose failuose. Kurioje kolekcijoje yra didžiausio nominalo moneta? Kuris kolekcininkas turtingiausias? Surašykite į atskirą rinkinį visus abiejų kolekcijų monetų, kurių nominalas yra 1, duomenis.

#### U3–5. Automobilis

- Studentas analizuoja automobilių tinkamumą kelionėms. Faile pirmoje eilutėje yra studento vardas ir pavardė. Toliau yra duomenys apie automobilius: markė, keleivių, telpančių į automobilį, skaičius, ir kuro, sunaudojamo 100 km, kiekiui saugoti. Sukurkite klasę *Automobilis*, kuri turėtų kintamuosius markę, keleivių, telpančių į automobilį, skaičiui ir kuro, sunaudojamo 100 km, kiekiui saugoti. Rasti, kiek kainuoja pigiausia kelionė (mažiausiai kuro sunaudoja) ir keli keleiviai telpa didžiausiame automobilyje.
- Papildykite programą veiksmams su dviejų studentų automobilių rinkiniais. Kiekvieno studento automobilių duomenys saugomi atskiruose failuose. Kurio studento sąrašas yra talpiausias automobilis? Surašykite į atskirą rinkinį visus abiejų studentų tyrinėtų automobilių, kurie sunaudoja 100 km kuro ne daugiau kaip talpiausias automobilis, kiekiui, duomenis.

#### U3–6. Knygos

- Studento perkamų knygų sąrašas yra faile: autorius, pavadinimas, kaina, puslapių skaičius. Pirmoje eilutėje yra studento vardas ir pavardė. Sukurkite klasę *Knyga*, kuri turėtų kintamuosius knygos pavadinimui, puslapių skaičiui ir knygos kainai saugoti. Kuri knyga brangiausia ir už kokią pinigų sumą studentas perka knygas?
- Papildykite programą veiksmams su dviejų studentų perkamomis knygomis. Kiekvieno studento perkamų knygų duomenys saugomi atskiruose failuose. Kurio studento sąrašas yra brangiausia knyga? Surašykite į atskirą rinkinį abiejų studentų visų knygų, kurių puslapių skaičius neviršija surastos brangiausios knygos puslapių skaičių, duomenis.

#### U3–7. Salė

- Koncertų salei siūlomų kėdžių tipų sąrašas yra faile: pavadinimas, užimamas plotas (stačiakampio formos), kaina. Pirmoje eilutėje yra firmos, kuri siūlo kėdes, pavadinimas. Sukurkite klasę *Kėdė*, kuri turėtų kintamuosius pavadinimui, kainai ir užimamam plotui saugoti. Raskite, kurio tipo kėdė užima didžiausią plotą ir kurio tipo kėdė yra brangiausia.
- Papildykite programą veiksmams su dviejų firmų siūlomų kėdžių rinkiniais. Kiekvienos firmos kėdžių duomenys saugomi atskiruose failuose. Kuriam sąrašas yra brangiausia kėdė? Surašykite į atskirą rinkinį abiejų sąrašų visus kėdžių, kurios yra vidutinio dydžio, duomenis. Vidutinė kėdė yra ta, kurios užimamas plotas skiriasi nuo turimų kėdžių užimamų plotų aritmetinio vidurkio ne daugiau, kaip 10 proc.

#### U3–8. Prekė

- Statomo namo sienų apšiltinimui siūlomų medžiagų sąrašas yra faile: pavadinimas, 1 m<sup>2</sup> kaina, storis. Pirmoje eilutėje yra statybinių prekių parduotuvės, kuri siūlo prekes, pavadinimas. Sukurkite klasę *Prekė*, kuri turėtų kintamuosius pavadinimui, kainai ir storiui saugoti. Raskite, kiek sąrašas yra medžiagų, kurių kaina neviršija namo savininko nurodytos kainos ir storis yra ribose [a, b].
- Papildykite programą veiksmams su dviejų parduotuvių siūlomų medžiagų rinkiniais. Kiekvienos parduotuvės medžiagų duomenys saugomi atskiruose failuose. Kuriam sąrašas yra brangiausia medžiaga? Surašykite į atskirą rinkinį visus abiejų sąrašų medžiagų, kurios tenkina namo savininko reikalavimus, duomenis.

#### U3–9. Lenta

- Sandėlyje esančių lentų tipų sąrašas yra faile: lentos medžio pavadinimas, lentos storis, ilgis, plotis ir kaina. Pirmoje eilutėje yra firmos, kuri prekiauja lentomis, pavadinimas. Sukurkite klasę *Lenta*, kuri turėtų kintamuosius lentos duomenims saugoti. Raskite, kurio tipo lentoms bus išleista daugiausiai pinigų, jeigu koncertų salės grindų ilgis yra *L*, o plotis *P*.
- Papildykite programą veiksmams su dviejų firmų siūlomų lentų rinkiniais. Kiekvienos firmos lentų duomenys saugomi atskiruose failuose. Kuriame sąraše yra brangiausia lenta? Surašykite į atskirą rinkinį visus abiejų sąrašų lentų, kurios yra vidutinio dydžio, duomenis. Vidutinė lenta yra ta, kurios užimamas plotas skiriasi nuo turimų lentų užimamų plotų aritmetinio vidurkio ne daugiau, kaip *p* proc.

### U3–10. Siuvykla

- Siuvyklos siuvamų kostiumų duomenys saugomi faile: modelio pavadinimas, medžiagos pavadinimas, pasiuvimui reikalingos medžiagos ilgis bei plotis ir atraižų kiekis (procentais, %). Pirmoje eilutėje yra siuvyklos, kuri siuva kostiumus, pavadinimas. Sukurkite klasę *Kostiumas*, kuri turėtų kintamuosius kostiumų duomenims saugoti. Kurio modelio kostiumui pasiūti reikia daugiausiai medžiagos ir kurio modelio kostiumui atraižų lieka mažiausiai?
- Papildykite programą veiksmams su dviejų siuvyklų siūlomų kostiumų rinkiniais. Kiekvienos siuvyklos duomenys saugomi atskiruose failuose. Kuriame sąraše yra kostiumas, kuriam reikia daugiausiai medžiagos? Surašykite į atskirą rinkinį visus abiejų sąrašų kostiumų modelius, kurie yra vidutinio dydžio, duomenis. Vidutinis kostiumas yra tas, kuriam pasiūti reikalingos medžiagos kiekis skiriasi nuo visų modelių reikalingos medžiagos aritmetinio vidurkio ne daugiau, kaip *p* proc.

### U3–11. Kepykla

- Kepykloje kepamos duonos ir pyragų asortimentas yra faile: kepinio pavadinimas, miltų pavadinimas, kepinio rupumas, kepaliuko svoris ir kepimo prieaugis (procentais, %). Pirmoje eilutėje yra kepyklos pavadinimas. Sukurkite klasę *Kepykla* kepamos produkcijos duomenims saugoti. Kuris kepinys mažiausiai rupus ir kurio kepimo prieaugis didžiausias?
- Papildykite programą veiksmams su dviejų kepyklų kepinų rinkiniais. Kiekvienos kepyklos kepinų duomenys saugomi atskiruose failuose. Kuriame sąraše yra kepinys, kurio kepimo prieaugis didžiausias. Surašykite į atskirą rinkinį visus abiejų sąrašų kepinus, kurie yra vidutinio svorio, duomenis. Vidutinis kepaliukas tas, kurio svoris skiriasi nuo visų kepinų kepaliukų aritmetinio vidurkio ne daugiau, kaip 10 proc.

### U3–12. Kepyklėlė

- Kepyklėlėje kepamų pyragaičių duomenys yra faile: pavadinimas, dešimčiai vienetų pagaminti reikalingų kiaušinių kiekis, miltų kiekis ir kitų priedų kiekis. Pirmoje eilutėje yra kepyklėlės pavadinimas. Sukurkite klasę *Pyragas* pyragaičio duomenims saugoti. Raskite, kuriai pyragaičių rūšiai iškepti reikia mažiausiai kiaušinių ir kurios rūšies pyragaičiai sunkiausi (miltų ir kitų priedų kiekis).
- Papildykite programą veiksmams su dviejų kepyklėlių pyragaičių rinkiniais. Kiekvienos kepyklėlės pyragaičių duomenys saugomi atskiruose failuose. Kuriame sąraše yra pyragaitis, kuriam iškepti reikia daugiausiai kiaušinių? Surašykite į atskirą rinkinį visus abiejų sąrašų pyragaičius, kurie yra vidutinio svorio (miltų ir kitų priedų kiekis), duomenis. Vidutinis pyragaičio svoris tas, kurio svoris skiriasi nuo visų kepinų aritmetinio vidurkio ne daugiau, kaip 10 proc.

### U3–13. Nuotraukos

- Nuotraukų kolekcijos duomenys yra faile: nuotraukos pavadinimas, tipas (spalvota ar nespalvota, stereo), aukštis ir plotis. Pirmoje eilutėje yra kolekcijos savininko pavardė ir vardas. Sukurkite klasę *Nuotrauka* nuotraukos duomenims saugoti. Raskite, kurio tipo yra didžiausia nuotrauka ir kiek kolekcijoje yra nurodyto tipo nuotraukų.
- Papildykite programą veiksmams su dviejų kolekcijų rinkiniais. Kiekvienos kolekcijos duomenys saugomi atskiruose failuose. Kurioje kolekcijoje yra didžiausia nuotrauka? Kurioje kolekcijoje yra daugiau nurodyto tipo nuotraukų? Surašykite į atskirą rinkinį visus abiejų sąrašų nespalvotų nuotraukų duomenis.

### U3–14. Transportas

- Miesto viešojo transporto maršrutų duomenys yra faile: maršruto pavadinimas, ilgis kilometrais ir stotelių skaičius. Pirmoje eilutėje yra transporto įmonės pavadinimas. Sukurkite klasę *Transportas* vieno maršruto duomenims saugoti. Raskite visų maršrutų bendrą ilgį ir kiek stotelių turi ilgiausias maršrutas.
- Papildykite programą veiksmams su dviejų transporto įmonių (pvz., autobusų ir troleibusų) rinkiniais. Kiekvienos įmonės duomenys saugomi atskiruose failuose. Kurio įmonės maršrutų bendras

ilgis didesnis ir kurios įmonės ilgiausias maršrutas turi daugiau stotelių? Surašykite į atskirą rinkinį abiejų sąrašų visus maršrutus, kuriuose yra ne mažiau, kaip n stotelių

### U3–15. Miestai

- Miestų duomenys yra faile: pavadinimas, plotas ir gyventojų skaičius. Pirmoje eilutėje yra valstybės pavadinimas. Sukurkite klasę `Miestas` vieno miesto duomenims saugoti. Raskite miestą, kurio gyventojų tankis didžiausias ir kiek valstybės miestuose gyvena žmonių.
- Papildykite programą veiksmiais su dviejų valstybių miestų rinkiniais. Kiekvienos valstybės duomenys saugomi atskiruose failuose. Kurios valstybės miestų gyventojų skaičius didesnis ir kurioje valstybėje yra tankiausiai apgyvendintas miestas. Surašykite į atskirą rinkinį visus abiejų sąrašų miestus, kuriuose yra ne mažiau, kaip k gyventojų, duomenis.

### U3–16. Atmintukai

- Atmintukų duomenys yra faile: markė, talpa ir kaina. Pirmoje eilutėje yra savininko pavardė ir vardas. Sukurkite klasę `Atmintukas` vieno atmintuko duomenims saugoti. Raskite kokia pigiausio atmintuko talpa ir už kokią pinigų sumą žmogus turi atmintukų.
- Papildykite programą veiksmiais su dviejų žmonių atmintukų rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Pas kurį žmogų yra didesnės talpos pigiausias atmintukas ir kuris žmogus turi atmintukų už didesnę kainą. Surašykite į atskirą rinkinį abiejų sąrašų visus nurodytos markės atmintukų duomenis.

### U3–17. Fonoteka

- Kūrinių duomenys yra faile: pavadinimas, autorius, grojimo trukmė ir failo dydis. Pirmoje eilutėje yra rinkinio savininko pavardė ir vardas. Sukurkite klasę `Fonoteka` vieno kūrinio duomenims saugoti. Kiek laiko reikia perklausyti visus turimus kūrinius? Kieno ir kokio kūrinio failas didžiausias?
- Papildykite programą veiksmiais su dviejų savininkų rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Kuris savininkas ilgiau savo kūrinius gali klausyti ir pas kurį savininką yra didžiausios apimties kūrinių failas? Surašykite į atskirą rinkinį visus abiejų sąrašų kūrinius, kurių trukmė neviršija n minučių.

### U3–18. Dienraštis

- Dienraščių duomenys yra faile: pavadinimas, tiražas ir kaina. Pirmoje eilutėje yra valstybės pavadinimas. Sukurkite klasę `Dienraštis` vieno dienraščio duomenims saugoti. Kurio dienraščio tiražo pinigine vertė didžiausia? Už kokią pinigų sumą yra pateikiama skaitytojui dienraščių?
- Papildykite programą veiksmiais su dviejų įmonių prenumeratorių rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Kurios įmonės prenumeratorių skaitomi dienraščiai už didesnę pinigų sumą ir kurioje įmonėje yra dienraštis, kurio pinigine vertė didesnė? Surašykite į atskirą rinkinį visus abiejų sąrašų dienraščius, kurių kaina neviršija n pinigų.

### U3–19. Žurnalas

- Žurnalų duomenys yra faile: pavadinimas, puslapių skaičius ir kaina. Pirmoje eilutėje yra valstybės pavadinimas. Sukurkite klasę `Žurnalas` vieno žurnalo duomenims saugoti. Kurio žurnalo puslapio kaina didžiausia? Kokia bendra visų žurnalų puslapių suma ir kokia vieno puslapio vidutinė kaina?
- Papildykite programą veiksmiais su dviejų valstybių žurnalų rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Kurioje valstybėje žurnalo puslapio kaina didžiausia? Kurios valstybės leidžiamų žurnalų vieno puslapio vidutinė kaina didesnė? Surašykite į atskirą rinkinį visus abiejų valstybių žurnalus, kurių kaina neviršija n pinigų.

### U3–20. Vaiduoklis

- Senose pilyse gyvena vaiduokliai. Duomenys apie juos yra faile: pavadinimas, amžius ir pasirodymo dažnumas. Pirmoje eilutėje yra valstybės pavadinimas. Sukurkite klasę `Vaiduoklis` vieno vaiduoklio duomenims saugoti. Koks seniausio vaiduoklio pasirodymo dažnumas? Koks valstybės pilyse gyvenančių vaiduoklių amžiaus vidurkis?
- Papildykite programą veiksmiais su dviejų valstybių vaiduoklių rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Kurios valstybės seniausio vaiduoklio pasirodymo dažnumas didesnis? Kurios valstybės vaiduoklių amžiaus vidurkis mažesnis? Surašykite į atskirą rinkinį visus abiejų valstybių vaiduoklius, kurių amžius neviršija n metų.

### U3–21. Batai

- Duomenys apie parduodamus batus yra faile: gamintojo pavadinimas, dydis, svoris ir kaina. Pirmoje eilutėje yra savininko (parduotuvės) pavadinimas. Sukurkite klasę `Batas` vieno bato duomenims saugoti. Kiek kainuoja sunkiausia batų pora? Kiek yra nurodytos firmos nurodyto dydžių batų?

- Papildykite programą veiksmiais su dviejų parduotuvių batų rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Kurios parduotuvės sunkiausia batų pora pigesnė? Kurioje parduotuvėje yra daugiau nurodytos firmos nurodyto dydžio batų? Surašykite į atskirą rinkinį visus abiejų parduotuvių batus, kurių kaina neviršija n pinigų.

### U3–22. Kaliausė

- Duomenys apie ūkininkų pagamintas kaliauses yra faile: tipas, aprėdų skaičius, svoris ir kaina. Pirmoje eilutėje yra ūkininko pavardė ir vardas. Sukurkite klasę `Kaliausė` vienos kaliausės duomenims saugoti. Kiek aprėdų turi sunkiausia kaliausė? Kokia vidutinė kaliausės kaina?
- Papildykite programą veiksmiais su dviejų ūkininkų kaliausių rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Kurio ūkininko sunkiausia kaliausė daugiau aprėdyta? Kurio ūkininko vidutinė kaliausės kaina mažesnė? Surašykite į atskirą rinkinį visus abiejų ūkininkų nurodyto tipo kaliausių duomenis.

### U3–23. Saldainiai

- Duomenys apie saldainius yra faile: pavadinimas, tipas ir kilogramo kaina. Pirmoje eilutėje yra savininko pavardė ir vardas. Sukurkite klasę `Saldainis` vieno saldainio duomenims saugoti. Kiek kainuoja saldainiai, jeigu kiekvieno pavadinimo yra po n kilogramų. Kokie nurodyto tipo saldainiai brangiausi?
- Papildykite programą veiksmiais su dviejų studentų saldainių rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Kurio studento rinkinys kainuoja brangiau, jeigu pirmasis studentas turi visų saldainių pavadinimų po n1 kilogramų, o antrasis – po n2? Surašykite į atskirą rinkinį visus abiejų studentų saldainių, kurių kaina didesne už k pinigų, duomenis.

### U3–24. Reklama

- Duomenys apie reklamas yra faile: užsakovas, minutės kaina, parodymų skaičius paroje. Pirmoje eilutėje yra TV pavadinimas. Sukurkite klasę `Reklama` vienos reklamos duomenims saugoti. Kiek pinigų sumoka reklamų užsakovai per parą? Kiek kainuoja ilgiausios trukmės reklama?
  - Papildykite programą veiksmiais su dviejų televizijų reklamų rinkiniais. Kiekvieno rinkinio duomenys saugomi atskiruose failuose. Kuri televizija per parą uždirba daugiau? Kurioje televizijoje ilgiausios trukmės reklama brangesnė? Surašykite į atskirą rinkinį visus abiejų televizijų reklamų, kurių minutės kaina didesne už k pinigų, duomenis.
-