
2. Klasė

Susipažinsite su:

- klasės duomenų tipu ir objektu – klasės tipo kintamuoju;
- objekto laukų (vidinių kintamųjų) reikšmių naudojimu ir keitimu;
- metodais, grąžinančiais rezultatus per vardą;
- metodais, grąžinančiais rezultatus per parametrus.

2.1 Viena klasė

- Viena klasė, keli objektai.
- Objekto laukų reikšmių keitimas ir naudojimas taikant objekto klasės metodus.
- Reikšmių skaičiavimai, paieška.

1 uždutis. Plytos ir namas.

Gamykla gamina kelių skirtingų dydžių plytas. Žinomas kiekvieno tipo plytų aukštis, plotis ir ilgis. Parašykite programą, kuri suskaičiuotų, kiek reikia kiekvieno tipo plytų, norint išmūryti 4 vienodas namo sienas. Langų ir durų nėra. Sienos mūrijamos tik iš vieno tipo plytų.

Pradiniai duomenys ir rezultatai.

	Pradiniai duomenys			Rezultatai
	Ilgis	Plotis	Aukštis	
Pirmas plytų tipas	250 mm	120 mm	88 mm	Pirmo tipo plytų reikia: 12544 Antro tipo plytų reikia: 16900 Trečio tipo plytų reikia: 19672
Antras plytų tipas	240 mm	115 mm	71 mm	
Trečias plytų tipas	240 mm	115 mm	61 mm	
Namo viena siena	12 m	0,23 m	3 m	

- **Programos kūrimo eiga.**
 - Sukuriama klasė vienos plytos duomenims saugoti.
 - Pagrindiniame metode `Main()` skelbiami trys objektai, skirti kiekvieno plytos tipo duomenims saugoti.
 - Namo sienos ilgis, plotis ir aukštis perskaičiuojami plytomis. Gauti skaičiai sudauginami. Gautas sienos tūris plytomis ir bus ieškomas plytų skaičius. Visi rezultatai pateikiami sveikaisiais skaičiais.
 - Skaičiuojami kiekvieno tipo plytų kiekiai namo sienoms mūryti.

Pirmas žingsnis.

- Sukurkite klasę `Plyta`, skirtą plytos duomenims saugoti. Klasės aprašą užrašykite programos failo pradžioje, prieš pagrindinį metodą `Main()`.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//-----
// Vienos klasės naudojimas, plytos charakteristikos
//-----
namespace plyta
{
    /** Klasė plytos duomenims saugoti
    @class Plyta */
    class Plyta
    {
        private int ilgis,      // plytos ilgis, milimetrais
                  plotis,      // plytos plotis, milimetrais
                  aukštis;      // plytos aukštis, milimetrais

        //-----
        /** Plytos duomenys
        @param ilgis - ilgio reikšmė
```

```

@param pločioReikšmė - pločio reikšmė
@param aukščioReikšmė - aukščio reikšmė */
//-----
public Plyta(int ilgis, int pločioReikšmė, int aukščioReikšmė)
{
    this.ilgis = ilgis;
    plotis = pločioReikšmė;
    aukštis = aukščioReikšmė;
}

/** grąžina plytos ilgį */
public int ImtiIlgį() { return ilgis; }

/** grąžina plytos plotį */
public int ImtiPlotį() { return plotis; }

/** grąžina plytos aukštį */
public int ImtiAukštį(){ return aukštis;}
}

```

```

class Program
{
    static void Main(string[] args)
    {

    }
}

```

- Parašykite pagrindinį metodą Main(), kurioje būtų vieno tipo plytų objektas. Spausdinkite objekto duomenis.

```

//-----
static void Main(string[] args)
{
    Plyta p1;
    p1 = new Plyta(250, 120, 88);
    Console.WriteLine("Plytos aukštis: {0,3:d} \nPlytos plotis: {1, 4:d} \n"
        + "Plytos ilgis: {2, 5:d}\n",
        p1.ImtiAukštį(), p1.ImtiPlotį(), p1.ImtiIlgį());
    Console.WriteLine("Programa baigė darbą!");
}

```

```

//-----

```

- Patikrinkite, kaip programa dirba. Ekrane turite matyti:

<pre> Plytos aukštis: 88 Plytos plotis: 120 Plytos ilgis: 250 Programa baigė darbą! Press any key to continue . . . </pre>

Antras žingsnis.

- Papildykite programą kintamaisiais, skirtais namo sienos aukščiui, ilgiui ir pločiui metrais saugoti:

```

double sienosIlgis = 12.0,
sienosPlotis = 0.23,
sienosAukštis = 3.0;

```

- Papildykite programą veiksmiais, skaičiuojančiais, kiek plytų reikia namo sienoms sumūryti:

```

int k1; // Pirmo tipo plytų kiekis
k1 = (int)(sienosIlgis * 1000 / p1.ImtiIlgį() *
sienosPlotis * 1000 / p1.ImtiPlotį() *
sienosAukštis * 1000 / p1.ImtiAukštį());
Console.WriteLine("1-o tipo plytų reikia: {0,6:d} \n ", (4 * k1));

```

- Išbandykite, kaip veikia programa. Ekrane turėtumėte matyti:

```
Plytos aukštis: 88
Plytos plotis: 120
Plytos ilgis: 250

1-o tipo plytų reikia: 12544

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

Trečias žingsnis.

- Papildykite programą objektu, skirtu antro tipo plytos duomenims saugoti ir spausdinti:

```
Plyta p2;
p2 = new Plyta(240, 115, 71);
Console.WriteLine("Plytos aukštis: {0,3:d} \nPlytos plotis: {1, 4:d} \n"
    + "Plytos ilgis: {2, 5:d}\n",
    p2.ImtiAukštį(), p2.ImtiPlotį(), p2.ImtiIlgį());
```

- Išbandykite, kaip veikia programa. Ekrane turėtumėte matyti:

```
Plytos aukštis: 88
Plytos plotis: 120
Plytos ilgis: 250

1-o tipo plytų reikia: 12544

Plytos aukštis: 71
Plytos plotis: 115
Plytos ilgis: 240

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

- Papildykite programą veiksmiais, skaičiuojančiais kiek plytų reikia namo sienoms mūryti iš antro tipo plytų:

```
int k2; // Antro tipo plytų kiekis
k2 = (int)(sienosIlgis * 1000 / p2.ImtiIlgį() *
    sienosPlotis * 1000 / p2.ImtiPlotį() *
    sienosAukštis * 1000 / p2.ImtiAukštį());
Console.WriteLine("2-o tipo plytų reikia: {0,6:d} \n ", (4 * k2));
```

- Išbandykite, kaip veikia programa. Ekrane turėtumėte matyti:

```
Plytos aukštis: 88
Plytos plotis: 120
Plytos ilgis: 250

1-o tipo plytų reikia: 12544

Plytos aukštis: 71
Plytos plotis: 115
Plytos ilgis: 240

2-o tipo plytų reikia: 16900

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

Ketvirtas žingsnis.

- Parašykite metodą, kuris suskaičiuotų, kiek plytų reikia vienai namo sienai mūryti:

```
static int VienaSiena(Plyta p, double sienosPlotis, double sienosIlgis,
    double sienosAukštis)
```

- Metodo tekstas gali būti toks:

```
//-----
/** Skaičiuoja ir grąžina, kiek plytų reikia vienai sienai išmūryti
@param p - objektas, kuriame yra vienos plytos charakteristikos
@param sienosPlotis - sienos pločio reikšmė
@param sienosIlgis - sienos ilgio reikšmė
```

```

@param sienosAukštis - sienos aukščio reikšmė */
//-----
static int VienaSiena(Plyta p, double sienosPlotis, double sienosIlgis,
                    double sienosAukštis)
{
    return (int)(sienosIlgis * 1000 / p.ImtiIlgį() *
                sienosPlotis * 1000 / p.ImtiPlotį() *
                sienosAukštis * 1000 / p.ImtiAukštį());
}
//-----

```

- Metodą VienaSiena() įrašykite po pagrindinio metodo Main().
- Pakeiskite plytų skaičiaus skaičiavimus kreipiniais į metodą. Pavyzdžiui, k1 kintamąjį, jo reikšmės skaičiavimą ir gauto rezultato spausdinimą galima užrašyti taip:

```

Console.WriteLine("1-o tipo plytų reikia: {0,6:d} \n ",
                 (4 * VienaSiena(p1,sienosPlotis,sienosIlgis,sienosAukštis)));

```

- Išbandykite, kaip veikia programa. Ekrane turėtumėte matyti tuos pačius rezultatus.
- Tokius keitimus padarykite ir su antro tipo plytomis. Išbandykite, kaip veikia programa. Ekrane turėtumėte matyti tuos pačius rezultatus.

Penktas žingsnis.

- Papildykite programą skaičiavimais su trečio tipo plytomis.
- Išbandykite programą. Ekrane turite matyti, kad trečio tipo plytų reikia 19672 (duomenys yra duomenų lentelėje).

Šeštasis žingsnis.

- Parašykite metodą plytos duomenims ekrane spausdinti:

```

//-----
/** Spausdina plytos charakteristikas
@param p - objektas, kuriame yra vienos plytos charakteristikos */
//-----
static void SpausdintiPlytą(Plyta p)
{
    Console.WriteLine("Plytos aukštis: {0,3:d} \nPlytos plotis: {1, 4:d} \n"
                    + "Plytos ilgis: {2, 5:d}\n",
                    p.ImtiAukštį(), p.ImtiPlotį(), p.ImtiIlgį());
}
//-----

```

- Pakeiskite plytų duomenų spausdinimo sakinius kreipiniais į metodą SpausdintiPlytą().
- Išbandykite programą. Rezultatai turi nepasikeisti.

Programos patikrinimas.

- Pakeiskite namo sienų dydžius. Skaičiuotuvu suskaičiuokite, kiek kokio tipo plytų reikia. Patikrinkite, ar programos rezultatai sutampa su suskaičiuotais.
- Patikrinkite, kaip dirba programa, kai sienos dydžiai mažesni už plytą.

Programos papildymas.

Namas yra bokšto formos. Sieną vieną ir ji yra cilindro formos. Parašykite metodą, kuris suskaičiuotų, kiek reikia plytų cilindro formos sienai mūryti. Tokią sieną aprašo trys dydžiai: cilindro skersmuo, aukštis ir sienos storis. Suskaičiuokite, kiek reikia kiekvieno tipo plytų bokštui mūryti.

2 užduotis. Lietuvos keliai.

Duoti trijų Lietuvos kelių duomenys:

- | | | |
|--|------------------|---------------------------------|
| 1) pavadinimas: "Kaunas - Vilnius", | ilgis: 105,0 km, | leistinas greitis: 110 km/val.; |
| 2) pavadinimas: "Kaunas - Alytus", | ilgis: 65,6 km, | leistinas greitis: 90 km/val.; |
| 3) pavadinimas: "Vilnius - Panevėžys", | ilgis: 136,0 km, | leistinas greitis: 120 km/val. |
-

Užrašykite klasę kelio duomenims (pavadinimas, ilgis, leistinas greitis) saugoti ir šios klasės objektus aukščiau nurodytų kelių duomenims saugoti. Parašykite programą, kuri rastų:

- per kiek laiko nuvažiuosime iš Alytaus į Panevėžį, jei važiuosime leistinu greičiu maršrutu Alytus - Kaunas - Vilnius – Panevėžys;
- kuris kelias yra ilgiausias (rasti kelio pavadinimą).

Pradiniai duomenys ir rezultatai.

Pradiniai duomenys
1) pavadinimas: "Kaunas - Vilnius", ilgis: 105,0 km, leistinas greitis: 110 km/val.;
2) pavadinimas: "Kaunas - Alytus", ilgis: 65,6 km, leistinas greitis: 90 km/val.;
3) pavadinimas: "Vilnius - Panevėžys", ilgis: 136,0 km, leistinas greitis: 120 km/val.
Rezultatai
Iš Alytaus į Panevėžį nuvažiuosime per 2.82 val.
Ilgiausias kelias: Vilnius - Panevėžys

Programos kūrimo eiga.

- Sukuriama klasė vieno kelio duomenims saugoti.
- Pagrindiniame metode `Main()` skelbiami trys užrašytos klasės objektai, skirti trijų Lietuvos kelių duomenims saugoti.
- Taikant klasės sąsajos metodus, suteikiamos reikšmės objektų kintamiesiems.
- Randamas laikas, per kurį galima nuvažiuoti iš Alytaus į Panevėžį, važiuojant leistinu greičiu maršrutu Alytus - Kaunas - Vilnius - Panevėžys.
- Randamas ilgiausio kelio pavadinimas.

Pirmas žingsnis.

- Sukurkite klasę, skirtą kelio duomenims saugoti.

```
//-----  
/** Klasė kelio duomenims saugoti  
@class Kelias */  
class Kelias  
{  
    private string pav;           // kelio pavadinimas  
    private double ilgis;         // kelio ilgis  
    private int lgr;              // leistinas greitis km/val.  
  
    //-----  
    /** Kelio duomenys  
    @param ilgis - kelio ilgio reikšmė  
    @param pav - kelio pavadinimas  
    @param lgr - leistinas greitis km/val. reikšmė */  
    //-----  
    public Kelias(string pav, double ilgis, int lgr)  
    {  
        this.pav = pav;  
        this.ilgis = ilgis;  
        this.lgr = lgr;  
    }  
  
    /** įrašo leistiną greitį km/val. */  
    public void DėtileistGreitį(int lg) { lgr = lg; }  
  
    /** grąžina kelio pavadinimą */  
    public string ImtiPav() { return pav; }  
  
    /** grąžina kelio ilgį */  
    public double ImtiIlgį() { return ilgis; }  
  
    /** grąžina leistiną greitį km/val. */  
    public int ImtileistGreitį() { return lgr; }  
}  
//-----
```

- Parašykite pagrindinį metodą `Main()`, kurioje būtų aprašyti reikalingi objektai ir suteiktos reikšmės šių objektų kintamiesiems. Spausdinkite objektų duomenis.

```
//-----
//-----
// Vienos klasės naudojimas
//-----
namespace kelias
{

// Klasė kelio duomenims saugoti

//-----
class Program
{
    static void Main(string[] args)
    {
        // Duomenų priskyrimas ir spausdinimas
        Kelias k1, k2, k3;           // objektai
        k1 = new Kelias("Kaunas - Vilnius", 105.0, 110);
        k2 = new Kelias("Kaunas - Alytus", 65.6, 90);
        k3 = new Kelias("Vilnius - Panevėžys", 136.0, 120);

        Console.WriteLine("Keliai (pavadinimas,\t\t ilgis,\t\t leistinas greitis:)");
        Console.WriteLine("{0},\t {1,10:f2}, {2,15:d}",
            k1.ImtiPav(), k1.ImtiIlgį(), k1.ImtiLeistGreitį());
        Console.WriteLine("{0},\t {1,10:f2}, {2,15:d}", k2.ImtiPav(),
            k2.ImtiIlgį(), k2.ImtiLeistGreitį());
        Console.WriteLine("{0},\t {1,10:f2}, {2,15:d} \n\n", k3.ImtiPav(),
            k3.ImtiIlgį(), k3.ImtiLeistGreitį());
        Console.WriteLine("Programa baigė darbą!");
    }
}
//-----
```

- Patikrinkite programos veikimą. Ekrane turite matyti:

Keliai (pavadinimas,	ilgis,	leistinas greitis:)
Kaunas - Vilnius,	105,00,	110
Kaunas - Alytus,	65,60,	90
Vilnius - Panevėžys,	136,00,	120

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

Antras žingsnis.

- Papildykite programą važiavimo laiko skaičiavimo ir spausdinimo veiksmams.

```
// Važiavimo laiko radimas
double laikas = k2.ImtiIlgį() / k2.ImtiLeistGreitį() +
    k1.ImtiIlgį() / k1.ImtiLeistGreitį() +
    k3.ImtiIlgį() / k3.ImtiLeistGreitį();
Console.WriteLine("Iš Alytaus į Panevėžį nuvažiuosime per {0,5:f2} val. ", laikas);
```

- Patikrinkite programos veikimą. Ekrane turite matyti:

Keliai (pavadinimas,	ilgis,	leistinas greitis:)
Kaunas - Vilnius,	105,00,	110
Kaunas - Alytus,	65,60,	90
Vilnius - Panevėžys,	136,00,	120

Iš Alytaus į Panevėžį nuvažiuosime per 2,82 val.
Programa baigė darbą!
Press any key to continue . . .

Trečias žingsnis.

- Papildykite programą ilgiausio kelio radimo ir spausdinimo veiksmams.

```
// Ilgiausio kelio radimas
string maxPav = k1.ImtiPav();
double maxIlgis = k1.ImtiIlgį();
if (k2.ImtiIlgį() > maxIlgis)
{
    maxPav = k2.ImtiPav(); maxIlgis = k2.ImtiIlgį();
}
if (k3.ImtiIlgį() > maxIlgis)
{
    maxPav = k3.ImtiPav(); maxIlgis = k3.ImtiIlgį();
}
Console.WriteLine();
Console.WriteLine("Ilgiausias kelias: {0}", maxPav);
```

- Patikrinkite programos veikimą. Ekrane turite matyti:

```
Keliai (pavadinimas,      ilgis,      leistinas greitis:)
Kaunas - Vilnius,        105,00,        110
Kaunas - Alytus,          65,60,          90
Vilnius - Panevėžys,     136,00,        120

Iš Alytaus į Panevėžį nuvažiuosime per 2,82 val.

Ilgiausias kelias: Vilnius - Panevėžys
Programa baigė darbą!
Press any key to continue . . .
```

Programos papildymas.

Papildykite programą veiksmams, nustatančiais, kuriame kelyje leistinas greitis yra mažiausias.

2.2. Dvi klasės

- Skirtingų klasių objektai.
- Reikšmių įvedimas klaviatūra. Skaičiavimai.

1 užduotis.

Sprendime sukurkite naują projektą. Į jį nukopijuokite 2.1 skyrelio pirmojo pratimo (1 užduotis) programą ir ją papildykite klase namo sienos duomenims saugoti.

Pirmas žingsnis.

Nukopijuokite nurodyto pratimo programą. Pašalinkite sakinius ir kintamuosius, skirtus darbui su konkrečia siena. Taip pat pašalinkite veiksmus su antro ir trečio tipo plytomis.

Antras žingsnis.

- Sukurkite klasę, skirtą namo sienos duomenims saugoti. Jos aprašą užrašykite prieš pagrindinį metodą Main().

```
//-----
/** Klasė sienos duomenims saugoti
@class Siena */
class Siena
{
    private double ilgis,      // sienos ilgis, metrais
                  plotis,      // sienos plotis, metrais
                  aukštis;      // sienos aukštis, metrais

    //-----
    /** Sienos duomenys
```

```

@param ilgis - nauja ilgio reikšmė
@param plotis - nauja pločio reikšmė
@param aukštis - nauja aukščio reikšmė */
//-----
public Siena(double ilgis, double plotis, double aukštis)
{
    this.ilgis = ilgis;
    this.plotis = plotis;
    this.aukštis = aukštis;
}

/** grąžina sienos ilgį */
public double ImtiIlgį() { return ilgis; }

/** grąžina sienos plotį */
public double ImtiPlotį() { return plotis; }

/** grąžina sienos aukštį */
public double ImtiAukštį() { return aukštis; }
}
//-----

```

- Papildykite pagrindinį metodą Main() namo sienos objektu. Spausdinkite duomenis.

```

//-----
static void Main(string[] args)
{
    Plyta p1;
    p1 = new Plyta(250, 120, 88);
    SpausdintiPlytą(p1);

    Siena s1;
    s1 = new Siena(12.0, 0.23, 3.0);
    Console.WriteLine("Sienos aukštis:\t {0,6:f2} \nSienos plotis:\t {1, 6:f2} \n"
        + "Sienos ilgis:\t {2, 6:f2}\n",
        s1.ImtiAukštį(), s1.ImtiPlotį(), s1.ImtiIlgį());
    Console.WriteLine("Programa baigė darbą!");
}
//-----
/** Spausdina plytos charakteristikas
@param p - objektas, kuriame yra vienos plytos charakteristikos */
//-----
static void SpausdintiPlytą(Plyta p)
{
    Console.WriteLine("Plytos aukštis: {0,3:d} \nPlytos plotis: {1, 4:d} \n"
        + "Plytos ilgis: {2, 5:d}\n",
        p.ImtiAukštį(), p.ImtiPlotį(), p.ImtiIlgį());
}
//-----

```

- Patikrinkite, kaip programa dirba. Ekrane turite matyti:

```

Plytos aukštis: 88
Plytos plotis: 120
Plytos ilgis: 250

Sienos aukštis: 3,00
Sienos plotis: 0,23
Sienos ilgis: 12,00

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

```

Trečias žingsnis.

- Parašykite metodą, kuri suskaičiuotų, kiek reikia plytų duotai sienai sumūryti:

```

//-----
/** Skaičiuoja ir grąžina, kiek plytų reikia vienai sienai išmūryti
@param p - objektas, kuriame yra vienos plytos charakteristikos

```



```

@param s - objektas, kuriame yra sienos charakteristikos */
//-----
static int ReikiaPlytu(Plyta p, Siena s)
{
    return (int)(s.ImtiIlgį() * 1000 / p.ImtiIlgį() *
                s.ImtiPlotį() * 1000 / p.ImtiPlotį() *
                s.ImtiAukštį() * 1000 / p.ImtiAukštį());
}
//-----

```

- Papildykite programą sakiniu:

```
Console.WriteLine("1-o tipo plytų reikia: {0,6:d} \n ", (4 * ReikiaPlytu(p1, s1)));
```

- Patikrinkite, kaip dirba programa. Ar rezultatai teisingi? Kadangi veiksmuose yra sveikieji ir realieji skaičiai, o rezultatas – sveikasis skaičius, tuomet galima vienos plytos paklaida.

Ketvirtas žingsnis.

- Papildykite programą objektais, skirtais antro ir trečio tipo plytų duomenims saugoti.
- Atlikite skaičiavimus. Patikrinkite, ar gaunami teisingi rezultatai.

Programos papildymas.

- Namo ilgis ir plotis nevienodi. Jeigu namo plotį pažymėsime a , ilgį b , o aukštį c , tuomet bus dvi sienos, kurių dydis bus $a * c$ ir dvi sienos, kurių dydis bus $b * c$. Sukurkite du objektus sienų duomenims saugoti.
- Papildykite programą skaičiavimais, kiek reikia kiekvieno tipo plytų tokio namo sienoms sumūryti.
- Patikrinkite, ar programa teisingai skaičiuoja.

Savarankiško darbo užduotis3

Statoma stačiakampio formos pilis su cilindriniais gynybiniais bokštais kampuose. Yra žinomi duomenys apie sienas: aukštis, plotis ir ilgis. Bokštai yra cilindro formos ir yra žinomi jų aukščiai, skersmenys ir storai. Pilis mūrijama iš vieno tipo plytų. Parašykite programą, kuri suskaičiuotų, kiek plytų reikia piliai sumūryti.

2 užduotis. Kelionės kaina.

Duoti Lietuvos kelių duomenys:

- | | | |
|--|------------------|---------------------------------|
| 1) pavadinimas: "Kaunas - Vilnius", | ilgis: 105,0 km, | leistinas greitis: 110 km/val.; |
| 2) pavadinimas: "Kaunas - Alytus", | ilgis: 65,6 km, | leistinas greitis: 90 km/val.; |
| 3) pavadinimas: "Vilnius - Panevėžys", | ilgis: 136,0 km, | leistinas greitis: 120 km/val. |

Duoti automobilių duomenys:

- | | | |
|------------------------------------|--------------------------|--------------------------------|
| 1) pavadinimas: "Opel Meriva", | degalų tipas: benzinas, | degalų sąnaudos 100 km: 7,5 l; |
| 2) pavadinimas: "Volkswagen Golf", | degalų tipas: dyzelinas, | degalų sąnaudos 100 km: 6,3 l. |

Sukurkite dvi klases ir šių klasių objektus aukščiau nurodytų kelių ir automobilių duomenims saugoti.

Kiek kainuos degalai keliaujant maršrutu Alytus - Kaunas - Vilnius - Panevėžys kiekvienu automobiliu?

Benzino ir dyzelino kainos kelionės dieną yra žinomos ir įvedamos klaviatūra.

Pradiniai duomenys ir rezultatai.

Pradiniai duomenys
<p>Keliai:</p> <p>1) pavadinimas: "Kaunas - Vilnius", ilgis: 105,0 km, leistinas greitis: 110 km/val.;</p> <p>2) pavadinimas: "Kaunas - Alytus", ilgis: 65,6 km, leistinas greitis: 90 km/val.;</p> <p>3) pavadinimas: "Vilnius - Panevėžys", ilgis: 136,0 km, leistinas greitis: 120 km/val.</p> <p>Automobiliai:</p> <p>1) pavadinimas: "Opel Meriva", degalų tipas: benzinas, degalų sąnaudos 100 km: 7,5 l;</p> <p>2) pavadinimas: "Volkswagen Golf", degalų tipas: dyzelinas, degalų sąnaudos 100 km: 6,3 l.</p> <p>1 litro benzino kaina: 1,38 Eur, 1 litro dyzelino kaina: 1,29 Eur.</p>
Rezultatai
<p>Automobiliu Opel Meriva iš Alytaus į Panevėžį nuvažiuosime už 31,73 Eur.</p> <p>Automobiliu Volkswagen Golf iš Alytaus į Panevėžį nuvažiuosime už 24,92 Eur.</p>

- **Programos kūrimo eiga.**

- Sprendime sukuriamas naujas projektas. Į jį nukopijuojama 2.1 skyrelio 2 užduoties programa.
- Sukuriama nauja klasė automobilio duomenims saugoti.

-
- Pagrindiniame metode `Main()` skelbiami du nauji automobilio klasės objektai.
 - Suteikiamos reikšmės objektų kintamiesiems.
 - Klaviatūra įvedamos benzino ir dyzelino kainos.
 - Randama, kiek kainuos kelionė iš Alytaus į Panevėžį per Kauną ir Vilnių pirmuoju automobiliu.
 - Randama, kiek kainuos kelionė iš Alytaus į Panevėžį per Kauną ir Vilnių antruoju automobiliu.

Pirmas žingsnis.

- Sukurkite naują projektą. Nukopijuokite 2.1 skyrelio 2 užduoties programą. Pašalinkite iš programos veiksmus, surandančius laiką, per kurį galima nuvažiuoti iš Alytaus į Panevėžį, ir ilgiausio kelio pavadinimą.

Antras žingsnis.

- Sukurkite klasę, skirtą automobilio duomenims saugoti.

```
//-----  
/** 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      - automobilio pavadinimas  
    @param degalai  - degalų tipas  
    @param sąnaudos - kuro sąnaudų 100 km reikšmė */  
    //-----  
    public Auto(string pav, string degalai, double sąnaudos)  
    {  
        this.pav = pav;  
        this.degalai = degalai;  
        this.sąnaudos = sąnaudos;  
    }  
  
    /** grąžina automobilio pavadinimą */  
    public string ImtiPav() { return pav; }  
  
    /** grąžina degalų tipą */  
    public string ImtiDegalus() { return degalai; }  
  
    /** grąžina sąnaudų kiekį */  
    public double ImtiSąnaudas() { return sąnaudos; }  
}  
//-----
```

- Papildykite pagrindinį metodą `Main()` dviejų objektų, saugančių automobilių duomenis, aprašais. Suteikite reikšmę šių objektų kintamiesiems ir spausdinkite objektų duomenis.

```
Auto a1, a2;  
    a1 = new Auto("Opel Meriva", "Benzinas", 7.5);  
    a2 = new Auto("Volkswagen Golf", "Dyzelinas", 6.3);  
Console.WriteLine("Automobiliai:");  
Console.WriteLine("{0},\t\t {1},\t {2,8:f2}",  
    a1.ImtiPav(), a1.ImtiDegalus(), a1.ImtiSąnaudas());  
Console.WriteLine("{0},\t {1},\t {2,8:f2}\n",  
    a2.ImtiPav(), a2.ImtiDegalus(), a2.ImtiSąnaudas());
```

- Patikrinkite programos veikimą. Ekrane turite matyti:

```

Keliai (pavadinimas,      ilgis,      leistinas greitis:)
Kaunas - Vilnius,        105,00,      110
Kaunas - Alytus,          65,60,       90
Vilnius - Panevėžys,     136,00,     120

Automobiliai:
Opel Meriva,              Benzinas,      7,50
Volkswagen Golf,          Dyzelinas,     6,30

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

```

Trečias žingsnis.

- Papildykite programą kintamaisiais, kuriuose būtų saugomos benzino ir dyzelino kainos. Įveskite šių kintamųjų reikšmes klaviatūra.

```

double benzkaina, dyzkaina;
// Degalų kainų įvedimas klaviatūra
Console.Write("Kiek kainuoja 1 litras benzino? ");
benzkaina = double.Parse(Console.ReadLine());
Console.Write("Kiek kainuoja 1 litras dyzelino? ");
dyzkaina = double.Parse(Console.ReadLine());
Console.WriteLine();

```

- Patikrinkite programos veikimą, kai 1 litras benzino kainuoja 1,38 Eur, o 1 litras dyzelino – 1,29 Eur. Ekrane turite matyti:

```

Keliai (pavadinimas,      ilgis,      leistinas greitis:)
Kaunas - Vilnius,        105,00,      110
Kaunas - Alytus,          65,60,       90
Vilnius - Panevėžys,     136,00,     120

Automobiliai:
Opel Meriva,              Benzinas,      7,50
Volkswagen Golf,          Dyzelinas,     6,30

Kiek kainuoja 1 litras benzino? 1,38
Kiek kainuoja 1 litras dyzelino? 1,29

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

```

Ketvirtas žingsnis.

- Parašykite metodą, kuri suskaičiuotų kelionės kainą:

```

//-----
/** Suskaičiuoja ir grąžina kelionės kainą
@param atstumas - kelionės ilgis (km)
@param sąnaudos - kuro sąnaudos 100 km
@param litroKaina - vieno litro degalų kaina (Eur) */
static double KelionėsKaina(double atstumas, double sąnaudos, double litroKaina)
{
    return (atstumas / 100) * sąnaudos * litroKaina;
}
//-----

```

- Papildykite programą kintamaisiais ir veiksmiais, reikalingais kelionės kainai skaičiuoti ir spausdinti.

```

// Kelionės išlaidų skaičiavimas
double kaina1, kaina2;
double atstumas = k1.ImtiIlgi() + k2.ImtiIlgi() + k3.ImtiIlgi();
if (a1.ImtiDegalus() == "Benzinas")
    kaina1 = KelionėsKaina(atstumas, a1.ImtiSąnaudas(), benzkaina);
else
    kaina1 = KelionėsKaina(atstumas, a1.ImtiSąnaudas(), dyzkaina);
if (a2.ImtiDegalus() == "Benzinas")
    kaina2 = KelionėsKaina(atstumas, a2.ImtiSąnaudas(), benzkaina);
else

```

```

        kaina2 = KelionėsKaina(atstumas, a2.ImtiSąnaudas(), dyzkaina);
Console.WriteLine("Automobiliu {0} iš Alytaus į Panevėžį nuvažiuosime už " +
        "{1,7:f2} Eur.", a1.ImtiPav(), kaina1);
Console.WriteLine("Automobiliu {0} iš Alytaus į Panevėžį nuvažiuosime už " +
        "{1,7:f2} Eur.\n", a2.ImtiPav(), kaina2);
Console.WriteLine("Programa baigė darbą!");

```

- Patikrinkite programos veikimą. Ekrane turite matyti:

```

Keliai <pavadinimas,          ilgis,          leistinas greitis:>
Kaunas - Vilnius,           105,00,           110
Kaunas - Alytus,            65,60,            90
Vilnius - Panevėžys,        136,00,           120

Automobiliai:
Opel Meriva,                Benzinas,          7,50
Volkswagen Golf,           Dyzelinas,         6,30

Kiek kainuoja 1 litras benzino? 1,38
Kiek kainuoja 1 litras dyzelino? 1,29

Automobiliu Opel Meriva iš Alytaus į Panevėžį nuvažiuosime už 31,73 Eur.
Automobiliu Volkswagen Golf iš Alytaus į Panevėžį nuvažiuosime už 24,92 Eur.

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

```

Programos papildymas.

Papildykite programą klase `Degalai` (tipas, 1 litro kaina) ir šios klasės objektais benzino ir dyzelino duomenims saugoti. Pertvarkykite kelionės iš Alytaus į Panevėžį kainos skaičiavimą.