

Bendrinė klasė

Užduotis

Failuose „Bakalaurai.txt“ ir „Magistrantai.txt“ duota informacija apie bakalauro ir magistro studijose besimokančius studentus: *vardas, pavardė, studento pažymėjimo nr., grupės pavadinimas, pažymių vidurkis, finansavimo statusas* (VF – valstybės finansuojama, VNF – valstybės nefinansuojama vieta). Magistrantų duomenų faile pridėtas papildomas laukas — *magistrinio darbo tema*.

Faile „Grupes.txt“ pateikta informacija apie grupes: *studijų pakopa* („B“ – bakalauro, „M“ – magistro), *kursas, grupės pavadinimas, specializacija, kuratorius*.

Atlikite šiuos žingsnius:

- 1) Sukurkite abstrakčią bazinę klasę *Studentas*, savybės: *vardas, pavarde, pazymejimoNr, vidurkis, statusas*. Klasėje turi būti konstruktorius be parametrų, ir konstruktorius su parametrais. Statusui sukurkite enumeratorių.
- 2) Sukurkite išvestines klases *Bakalauras* ir *Magistrantas*. Abi jos turi turėti konstruktorius be parametrų, ir su parametrais. Klasė *Magistrantas* turės papildomą savybę *tema*.
- 3) Sukurkite bendrinę klasę *Grupe*, savybės: *pavadinimas, kursas, specializacija, kuratorius, studentuSarasas*. Klasė turi turėti konstruktorių su parametrais. Uždrauskite grupės pavadinimo keitimą.
- 4) *Form* klasėje sukurkite žodynus(Dictionary), skirtus bakalauro ir magistrantų grupėms saugoti. Žodyno raktas – grupės pavadinimas, reikšmė – grupės objektas.
- 5) Sukurkite grupių skaitymo metodą.
- 6) Sukurkite grafinę vartotojo sąsają. Į pagrindinio lango formą įkelkite komponentus *MenuStrip* ir *TabControl*.
- 7) Sukurkite metodą, kuris dinamiškai sukurtų puslapį kiekvienai grupei.
- 8) Sukurkite bakalauro ir magistrantų studentų skaitymo metodus. Nuskaitytą studento duomenis, iškart įrašykite juos į atitinkamą grupę.
- 9) Papildykite dinamiškai kuriamą puslapį, atvaizduokite informaciją apie grupei priklausančius studentus.

Pradiniai duomenys

Grupes.txt
B; 1; IFX-5/1; -; Jonaitis Jonas
B; 1; IFX-5/2; -; Jonaitis Jonas
B; 2; IFX-4/1; -; Kazlauskas Kazys
B; 3; IFP-3/1; Programavimas; Jankauskas Juozas
B; 3; IFM-3/1; Multimedija; Stasiulis Stasys
M; 1; IFT-1; Tinklai; Andriukaitis Andrius
M; 1; IFPI-1; Programų inžinerija; Andriukaitis Andrius

Bakalaurai.txt
Petraitis; Petras; 123456; IFX-5/1; 8.5; VF
Arnas;Sofauskas;91357;IFX-5/1;6.8;VF
Gailė;Pagalvytė;92468;IFX-5/2;7.2;VNF
Jonas;Kėdžius;93579;IFX-4/1;8.9;VF
Radvilė;Staliukaitė;94680;IFX-5/1;9.2;VF
Ignas;Lovauskas;90864;IFX-4/1;4.6;VNF
Indrė;Langaitė;99753;IFX-4/1;7.8;VF
Diana;Paveikslaitė;98642;IFX-5/1;9.6;VF
Dainius;Lentauskas;97531;IFP-3/1;5.6;VNF
Raminta;Foteliūtė;96420;IFX-5/2;8.8;VF
Raigardas;Knygius;95790;IFP-3/1;6.5;VNF
Giedrius;Palangiauskas;91470;IFM-3/1;7.5;VF
Rasa;Plaktukaitė;92683;IFX-5/2;8.5;VF
Ieva;Pjūklaitė;90273;IFM-3/1;7.8;VF
Gintaras;Grėbliauskas;90876;IFX-4/1;7.7;VF
Vladimiras;Lygiauskas;93261;IFX-5/1;6.9;VNF
Jomantė;Dėklaitė;97365;IFX-5/2;5.6;VNF

Magistrantai.txt
Petraitis; Petras; 123456; IFT-1; 8.8; VF;6G pritaikymo galimybės
Simona;Tinklaitė;92725;IFT-1;8.6;VF;Naujų tinklo protokolų kūrimas
Audrius;Dalgiauskas;97362;IFPI-1;8.9;VF;Išmanus programinės įrangos testavimas
Audronė;Grindžiūtė;99827;IFPI-1;9.1;VF;Daiktų interneto pritaikymo galimybės
Vytenis;Dygliauskas;94725;IFT-1;8.7;VF;Gramatikos evoliucija ir jos praktiniai taikymai

Pirmas žingsnis.

- Sukurkite abstrakčią bazinę klasę *Studentas*, bei enumeratorių *Statusas*

```
//-----  
  
enum Statusas  
{  
    VF, //valstybės finansuojama vieta  
    VNF //valstybės nefinansuojama vieta  
}  
//-----  
  
abstract class Studentas  
{  
    public Studentas()  
    { }  
    public Studentas(string vardas, string pavarde, string pazymejimoNr, double  
vidurkis, Statusas statusas)  
    {  
        Vardas = vardas;  
        Pavarde = pavarde;  
        PazymejimoNr = pazymejimoNr;  
        Statusas = statusas;  
    }  
  
    public string Vardas { get; set; }  
    public string Pavarde { get; set; }  
    public string PazymejimoNr { get; set; }  
    public double Vidurkis { get; set; }  
    public Statusas Statusas { get; set; }  
  
}  
  
//-----
```

Antras žingsnis.

- Sukurkite išvestines klases *Bakalauras* ir *Magistrantas*

```
//-----  
  
class Bakalauras : Studentas  
{  
    public Bakalauras()  
    {  
    }  
    public Bakalauras(string vardas, string pavarde, string pazymejimoNr, double  
vidurkis, Statusas statusas)  
        : base(vardas, pavarde, pazymejimoNr, vidurkis, statusas)  
    {  
    }  
}  
  
//-----  
  
class Magistrantas : Studentas  
{
```

```

        public string Tema { get; set; }

        public Magistrantas()
        {
        }

        public Magistrantas(string vardas, string pavarde, string pazymejimoNr, double
vidurkis, Statusas statusas, string tema)
            : base(vardas, pavarde, pazymejimoNr, vidurkis, statusas)
        {
            Tema = tema;
        }
    }
}
//-----

```

Trečias žingsnis.

- Sukurkite bendrinę klasę *Grupe*

```

//-----
class Grupe<T> where T : Studentas
{
    public Grupe(string pavadinimas, int kursas, string specializacija, string
kuratorius)
    {
        Pavadinimas = pavadinimas;
        Kursas = kursas;
        Specializacija = specializacija;
        Kuratorius = kuratorius;
        StudentuSarasas = new List<T>();
    }

    public string Pavadinimas { get; private set; } //uždraudžiame pavadinimo keitimą
    public int Kursas {get; private set;}
    public string Specializacija { get; set; }
    public string Kuratorius { get; set; }

    public List<T> StudentuSarasas { get; private set; }
}
//-----

```

Atkreipkite dėmesį, jog tipo parametras *T* yra apribotas – kaip parametą galėsime nurodyti tik tas klases, kurių bazinė klasė yra *Studentas*.

Taip pat atkreipkite dėmesį, jog tipo parametras perduodamas sąrašui (List).

Ketvirtas žingsnis.

- Sukurkite pradinių duomenų failų konstantas:

```

//-----
const string CFGrupes = "..\\..\\Grupes.txt";
const string CFBakalaurai = "..\\..\\Bakalaurai.txt";
const string CFMagistrantai = "..\\..\\Magistrantai.txt";
//-----

```

- *Form* klasėje sukurkite žodynus(Dictionary), skirtus bakalaurų ir magistrantų grupėms saugoti.

```
//-----
    private Dictionary<string, Grupe<Bakalauras>> bakalauruGrupes = new
Dictionary<string, Grupe<Bakalauras>>();
    private Dictionary<string, Grupe<Magistrantas>> magistrantuGrupes = new
Dictionary<string, Grupe<Magistrantas>>();
//-----
```

Atkreipkite dėmesį, kaip klasei Grupe nurodomas tipo parametras.

Patikrinkite, kas atsitiks, jei pamėginsite nurodyti kitą tipo parametą (pvz. `string`).

Penktas žingsnis.

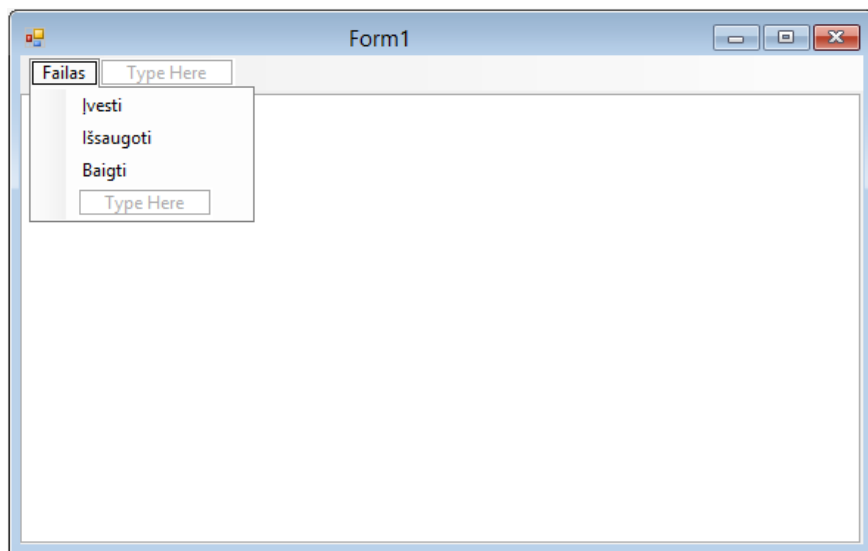
- Sukurkite grupių skaitymo metodą. Grupės iš karto dėsime į atitinkamą žodyną.

```
//-----
    static void SkaitytiGrupes(string failas, Dictionary<string, Grupe<Bakalauras>>
bakalauruGrupes, Dictionary<string, Grupe<Magistrantas>> magistrantuGrupes)
    {
        using (StreamReader reader = new StreamReader(failas,
Encoding.GetEncoding(1257)))
        {
            string line;
            while ((line = reader.ReadLine()) != null)
            {
                string[] parts = line.Split(';');
                string pozymis = parts[0].Trim();
                int kursas = int.Parse(parts[1].Trim());
                string grupe = parts[2].Trim();
                string specializacija = parts[3].Trim();
                string kuratorius = parts[4].Trim();

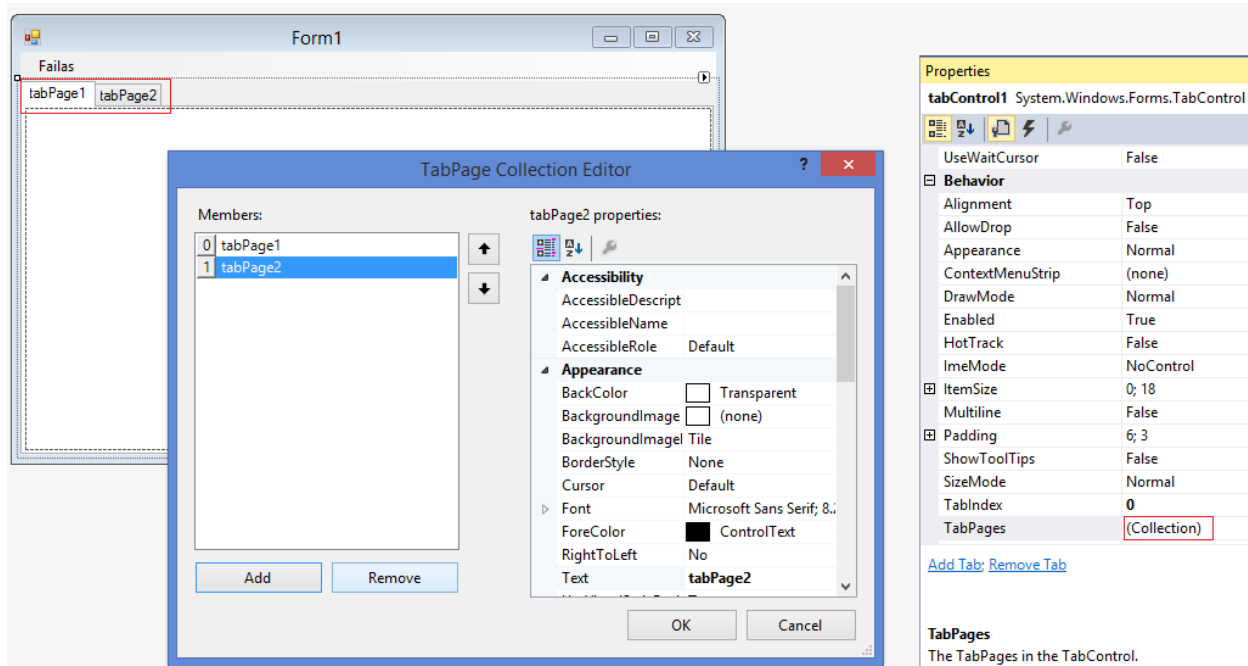
                switch (pozymis)
                {
                    case "B":
                        Grupe<Bakalauras> bg = new Grupe<Bakalauras>(grupe, kursas,
specializacija, kuratorius);
                        bakalauruGrupes.Add(grupe, bg);
                        break;
                    ;
                    case "M":
                        Grupe<Magistrantas> mg = new Grupe<Magistrantas>(grupe, kursas,
specializacija, kuratorius);
                        magistrantuGrupes.Add(grupe, mg);
                        break;
                }
            }
        }
    }
//-----
```

Šeštas žingsnis.

- Sukurkite pagrindinę klasę *Program*, su grafine vartotojo sąsaja.
- Į pagrindinio lango formą įkelkite komponentus *MenuStrip* ir *TabControl*.



- Įdėjus komponentą *TabControl*, jame bus automatiškai sugeneruoti keli puslapiai. Ištrinkite juos.



Septintas žingsnis.

- Sukurkite metodą, kuris dinamiškai sukurtų puslapį kiekvienai grupei.

```
//-----  
private void SukurtiGrupesPuslapi<T>(Grupe<T> grupe) where T : Studentas  
{  
    TabPage puslapis = new TabPage(grupe.Pavadinimas);  
    tabControl1.TabPages.Add(puslapis);  
  
    FlowLayoutPanel skydelis = new FlowLayoutPanel();  
    skydelis.Dock = DockStyle.Fill;  
    skydelis.FlowDirection = FlowDirection.TopDown;  
    puslapis.Controls.Add(skydelis);  
  
    Label kursas = new Label();  
    kursas.Text = "Kursas: " + grupe.Kursas;  
    kursas.AutoSize = true;  
    skydelis.Controls.Add(kursas);  
  
    Label specializacija = new Label();  
    specializacija.Text = "Specializacija: " + grupe.Specializacija;  
    specializacija.AutoSize = true;  
    skydelis.Controls.Add(specializacija);  
  
    Label kuratorius = new Label();  
    kuratorius.Text = "Kuratorius: " + grupe.Kuratorius;  
    kuratorius.AutoSize = true;  
    skydelis.Controls.Add(kuratorius);  
    //TODO vėliau studentų sąrašo atvaizdavimą pridėsime čia  
    puslapis.Refresh();  
}  
//-----
```

- Įgyvendinkime meniu punktą *Įvesti*.

```
//-----  
private void ivestiMeniu_Click(object sender, EventArgs e)  
{  
    SkaitytiGrupes(CFGrupes, bakalauruGrupes, magistrantuGrupes);  
  
    foreach(KeyValuePair<string, Grupe<Bakalauras>> irasas in bakalauruGrupes)  
    {  
        SukurtiGrupesPuslapi(irasas.Value);  
    }  
  
    foreach (KeyValuePair<string, Grupe<Magistrantas>> irasas in magistrantuGrupes)  
    {  
        SukurtiGrupesPuslapi(irasas.Value);  
    }  
}  
//-----
```

- Patikrinkite programos veikimą.

Aštuntas žingsnis.

- Sukurkite bakalaūrų ir magistrantų studentų skaitymo metodus.

```
//-----
static void SkaitytiBakalaurus(string failas, Dictionary<string,
Grupe<Bakalauras>> bakalauruGrupes)
{
    using (StreamReader reader = new StreamReader(failas,
Encoding.GetEncoding(1257)))
    {
        string line;
        while ((line = reader.ReadLine()) != null)
        {
            string[] parts = line.Split(';');

            string vardas = parts[0].Trim();
            string pavarde = parts[1].Trim();
            string pazymejimoNr = parts[2].Trim();
            string grupe = parts[3].Trim();
            double vidurkis = double.Parse(parts[4].Trim());
            Statusas statusas = (Statusas)Enum.Parse(typeof(Statusas),
parts[5].Trim());

            Bakalauras bakalauras = new Bakalauras(vardas, pavarde, pazymejimoNr,
vidurkis, statusas);
            bakalauruGrupes[grupe].StudentuSarasas.Add(bakalauras);
        }
    }
}
```



```

    }
}

//-----
static void SkaitytiMagistrantus(string failas, Dictionary<string,
Grupe<Magistrantas>> magistrantuGrupes)
{
    using (StreamReader reader = new StreamReader(failas,
Encoding.GetEncoding(1257)))
    {
        string line;
        while ((line = reader.ReadLine()) != null)
        {
            string[] parts = line.Split(';');

            string vardas = parts[0].Trim();
            string pavarde = parts[1].Trim();
            string pazymejimoNr = parts[2].Trim();
            string grupe = parts[3].Trim();
            double vidurkis = double.Parse(parts[4].Trim());
            Statusas statusas = (Statusas)Enum.Parse(typeof(Statusas),
parts[5].Trim());
            string tema = parts[6].Trim();

            Magistrantas magistrantas = new Magistrantas(vardas, pavarde,
pazymejimoNr, vidurkis, statusas, tema);
            magistrantuGrupes[grupe].StudentuSarasas.Add(magistrantas);
        }
    }
}

```

- Papildykite ivestiMeniu_Click metodą. Po grupių duomenų nuskaitymo pridėkite šias eilutes:

```

//-----
SkaitytiBakalaurus(CFBakalaurai, bakalauruGrupes);
SkaitytiMagistrantus(CFMagistrantai, magistrantuGrupes);
//-----

```

Devintas žingsnis.

- Papildykite SukurtiGrupesPuslapi metoda, pridėkite studento informacijos atvaizdavimą.

```

DataGridView tinklelis = new DataGridView();

tinklelis.AutoGenerateColumns = false;
tinklelis.RowHeadersVisible = false;
tinklelis.AllowUserToAddRows = false;
tinklelis.AllowUserToDeleteRows = false;

tinklelis.AutoSize = true;
tinklelis.BackgroundColor = Color.White;

BindingSource duomenuModelis = new BindingSource();
foreach (Studentas s in grupe.StudentuSarasas)

```

```

{
    duomenųModelis.Add(s);
}
//kiekvienas studento objektas atitiks vieną eilutę tinklelyje
tinklelis.DataSource = duomenųModelis;

//apsirašom tinklelio stulpelius
DataGridViewColumn column = new DataGridViewTextBoxColumn();
column.DataPropertyName = "PazymejimoNr"; //nurodome Studento objekto savybės
pavadinimą
column.Name = "Paž. nr."; //nurodome stulpelio pavadinimą
column.ReadOnly = true;
tinklelis.Columns.Add(column);

column = new DataGridViewTextBoxColumn();
column.DataPropertyName = "Vardas";
column.Name = "Vardas";
column.ReadOnly = true;
tinklelis.Columns.Add(column);

column = new DataGridViewTextBoxColumn();
column.DataPropertyName = "Pavarde";
column.Name = "Pavarde";
column.ReadOnly = true;
tinklelis.Columns.Add(column);

DataGridViewComboBoxColumn combo = new DataGridViewComboBoxColumn();
combo.DataSource = Enum.GetValues(typeof(Statusas));
combo.DataPropertyName = "Statusas";
combo.Name = "Statusas";
tinklelis.Columns.Add(combo);

//tikriname T tipą, ir esant reikalui pridedame papildomą stulpelį
if (typeof(T) == typeof(Magistrantas))
{
    column = new DataGridViewTextBoxColumn();
    column.DataPropertyName = "Tema";
    column.Name = "Darbo tema";
    column.ReadOnly = true;
    column.AutoSizeMode = DataGridViewAutoSizeColumnMode.DisplayedCells;
    tinklelis.Columns.Add(column);
}

skydelis.Controls.Add(tinklelis);

```

Atkreipkite dėmesį, kaip studento objekto savybė susiejama su atitinkamu `DataGridView` stulpeliu.

- Patikrinkite programos veikimą.

Form1

Failas

IFX-5/1 IFX-5/2 IFX-4/1 IFP-3/1 IFM-3/1 IFT-1 IFPI-1

Kursas: 3
Specializacija: Programavimas
Kuratorius: Jankauskas Juozas

Paž. nr.	Vardas	Pavarde	Statusas	
97531	Dainius	Lentauskas	VNF	▼
95790	Raigardas	Knygius	VNF	▼

Savarankiško darbo užduotis

- 1) Papildykite klasę *Grupe* metodu *GeriausiasStudentas*, kuris rastų ir gražintų studentą, kurio pažymių vidurkis aukščiausias.
- 2) Pakoreguokite programą taip, kad kiekviename puslapyje būtų parodyta geriausiai besimokančio studento duomenys.
- 3) Sukurkite dinamiškai du puslapius „Visi bakalaurai“ ir „Visi magistrantai“, juose parodykite atitinkamų studentų sąrašus. Studentus išrikiuokite pagal pavardę ir vardą. Įgyvedinimui sukurkite meniu punktą *Parodyti sąrašus*. (neprivalomas)