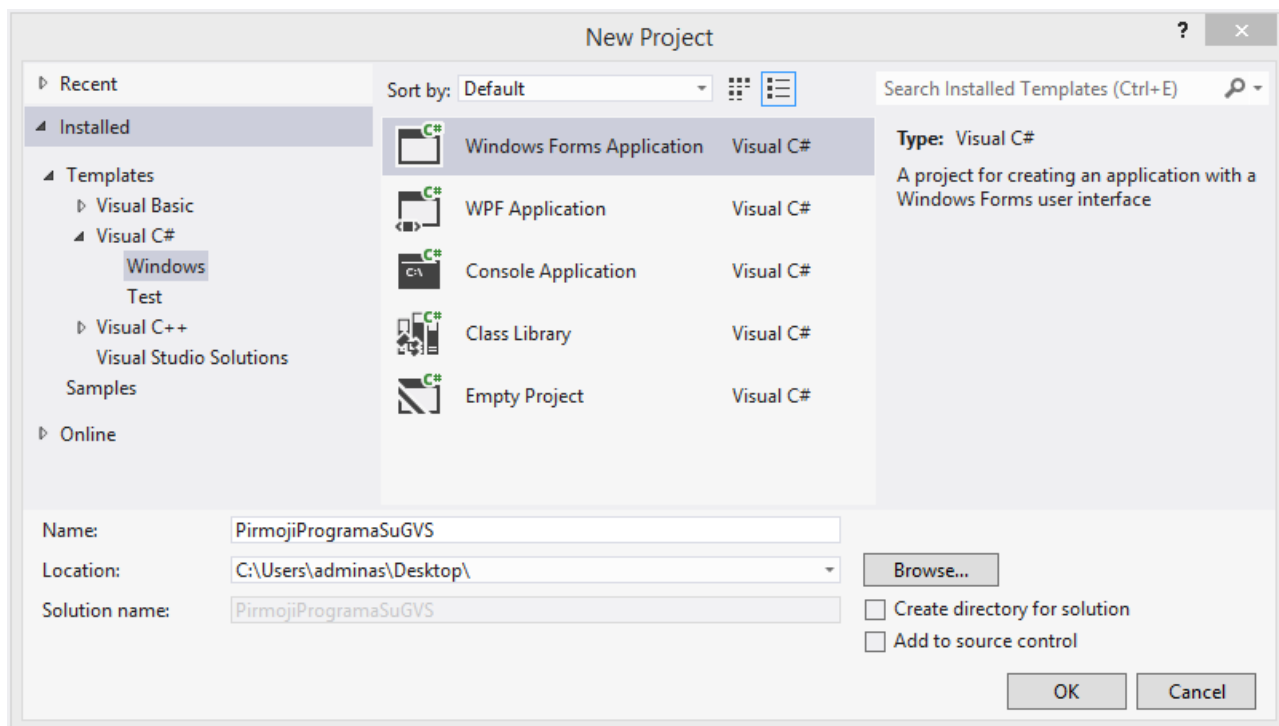


I. Grafinės vartotojo sąsajos kūrimas

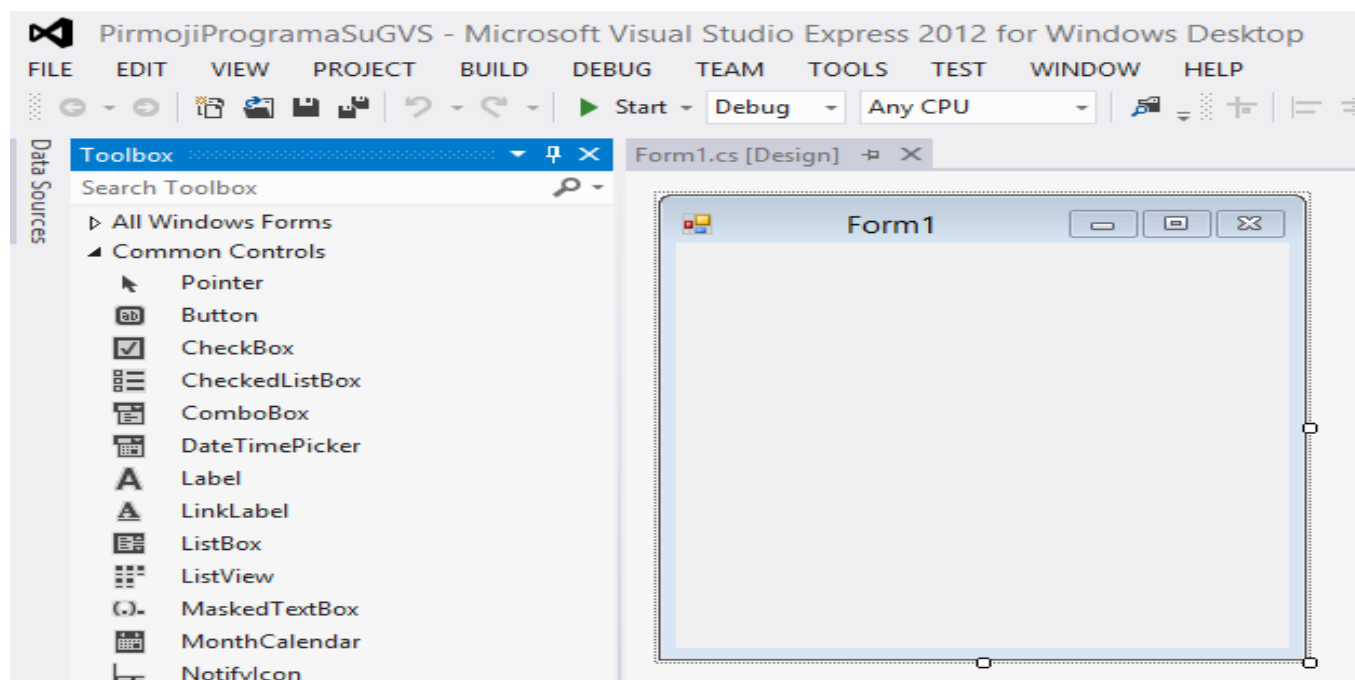
Grafinės vartotojo sąsajos (GVS, angl. GUI) kūrimui skirti du šablonai: Windows formų aplikacijų (*Windows Forms Application*) ir WPF (*Windows Presentation Foundation*) šablonai. Windows formų aplikacijų šablonas pasirodė su .NET Framework 1.0 versija, o WPF – su .NET Framework 3.0 versija. Mes naudosime pirmąjį šabloną.

I.1. Naujo projekto kūrimas

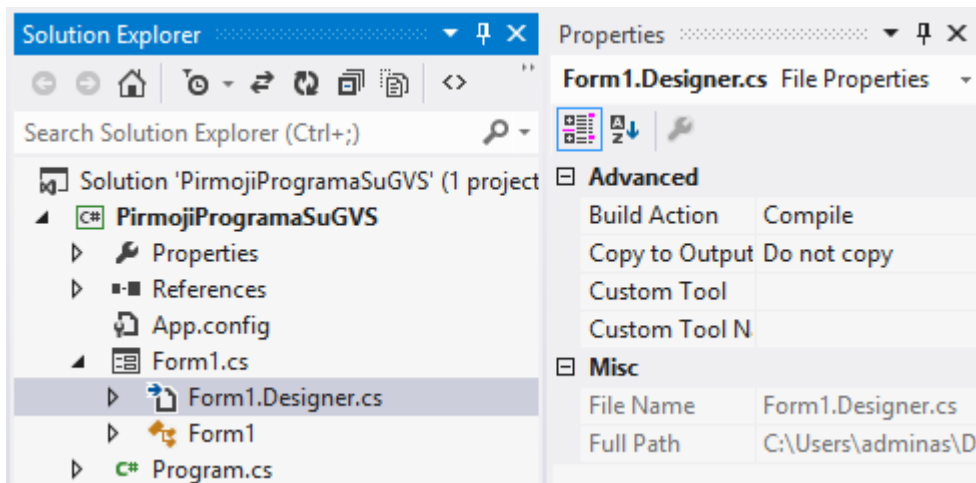
Atvėrus Microsoft Visual 2012 aplinką, pasirenkama kurti naują projektą (*New Project*) ir atsidariusiame lange pasirenkamos nuostatos, kaip parodyta paveikslėlyje. Kuriamam projektui suteikiamas pavadinimas **PirmojiProgramaSuGVS**.



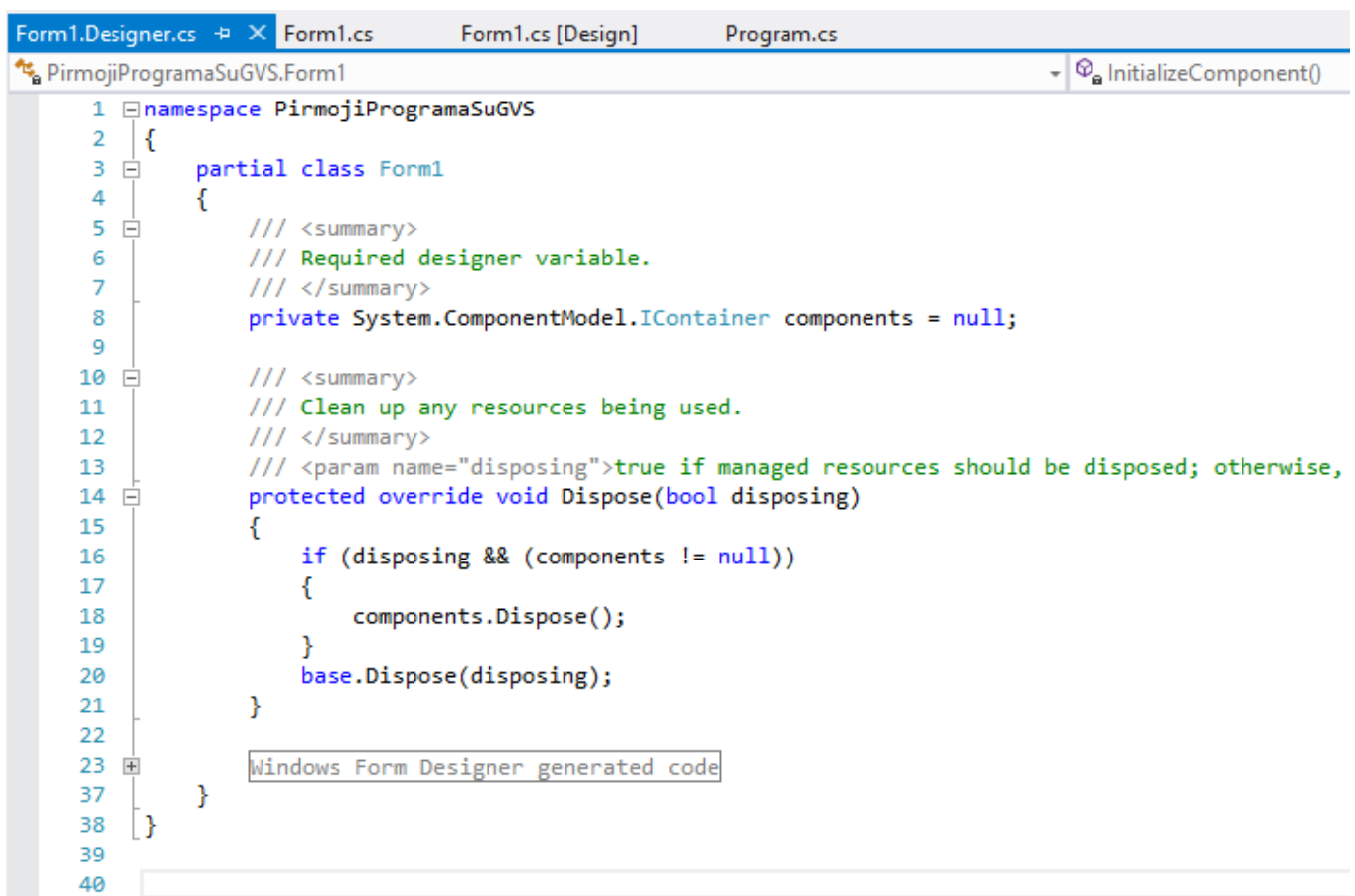
Sukūrus projektą ekrane atsiveria formos langas (**Form1.cs[Design]**), į kurį bus galima įkelti programos valdymui reikalingus komponentus (label, button ir kt.) iš *ToolBox* lango. Kitais žodžiais tariant sukuriams klasės **Form1** objektas – forma. Šios formos (lango) dydį galima keisti: didinti, mažinti, panaudojant pelę.



Projekto sukūrimo metu taip pat sukuriami failai, kuriuos galite matyti **Solution Explorer** lange. Jeigu šio lango nesimato, jį galima išsikviesti pasirinkus meniu *View->Solution Explorer*.



Faile **Form1.Designer.cs** bus galima pamatyti formoje **Form1** esančius komponentus, jų vardus, tipus, savybes ir panašiai. Įkeliant į formą komponentus, šis failas automatiškai bus papildomas.



Faile **Form1.cs** yra pateiktas klasės **Form1** aprašas. Šį failą papildysime uždavinio sprendimui reikalingomis priemonėmis: konstantomis, kintamaisiais, objektais, metodais ir pan.

```
Form1.Designer.cs  Form1.cs  Form1.cs [Design]  Program.cs
PirmojiProgramaSuGVS.Form1  Form1()

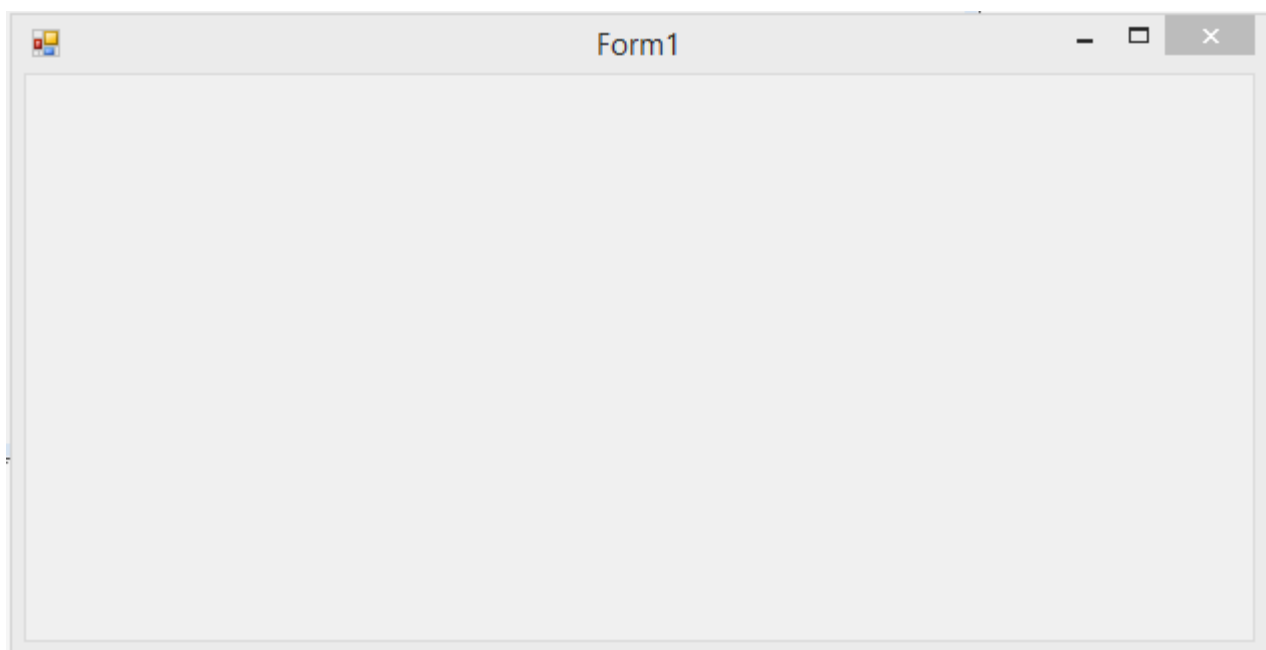
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace PirmojiProgramaSuGVS
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19     }
20 }
21
```

Faile **Program.cs** yra pateiktas klasės **Program** aprašas. Ši klasė turi metodą **Main()**, kuris pradeda programos su grafine vartotojo sąsaja vykdymą.

```
Form1.Designer.cs  Form1.cs  Form1.cs [Design]  Program.cs  Main()
PirmojiProgramaSuGVS.Program  Main()

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using System.Windows.Forms;
6
7  namespace PirmojiProgramaSuGVS
8  {
9      static class Program
10      {
11          /// <summary>
12          /// The main entry point for the application.
13          /// </summary>
14          [STAThread]
15          static void Main()
16          {
17              Application.EnableVisualStyles();
18              Application.SetCompatibleTextRenderingDefault(false);
19              Application.Run(new Form1());
20          }
21      }
22  }
23
```

Įvykdžius programą, sugeneruojamas tuščias formos langas su visomis lango valdymo priemonėmis.



I.2. Nesudėtingos programos grafinės vartotojo sąsajos kūrimas

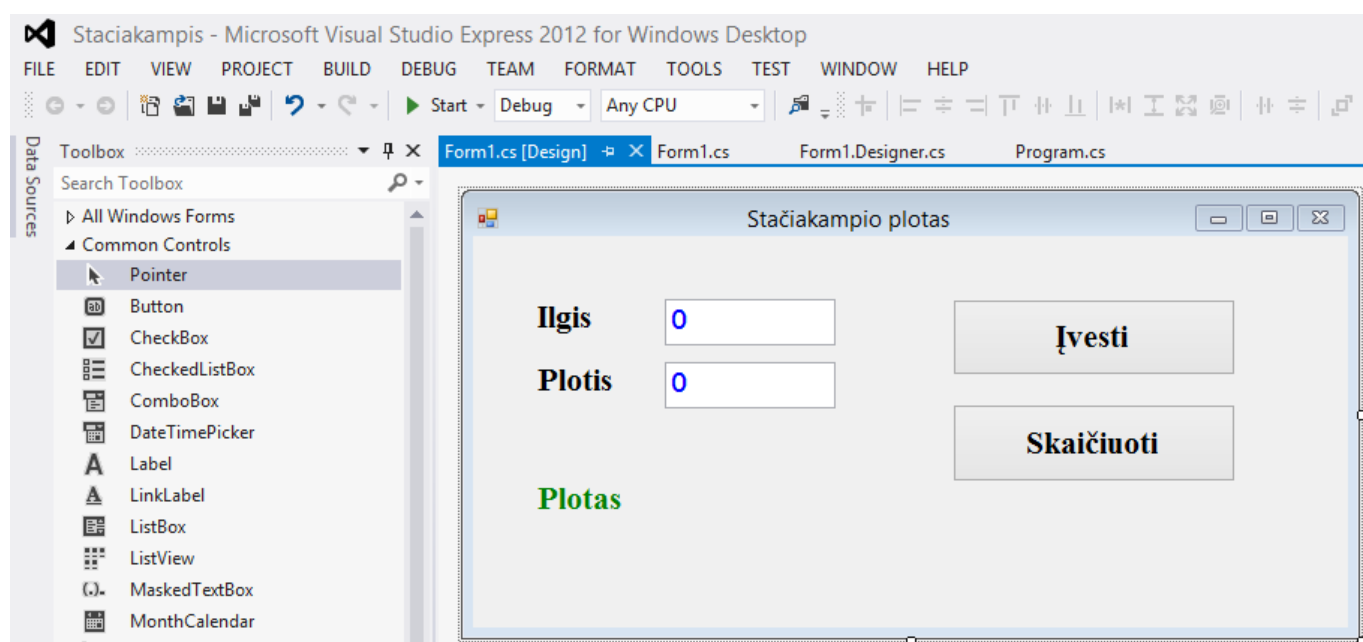
Grafinės vartotojo sąsajos kūrimą pademonstruosime paprastu pavyzdžiu.

Užduotis: sukurkite programą su grafine vartotojo sąsaja (GVS) stačiakampio plotui rasti, kai yra žinomi jo kraštinių matmenys: ilgis ir plotis.

Užduoties sprendimui sukuriame projektą (*Windows Forms Application*), kurį pavadiname **Staciakampis**.

Programos sąsajos kūrimui naudosime trijų tipų komponentus: label, button ir textBox.

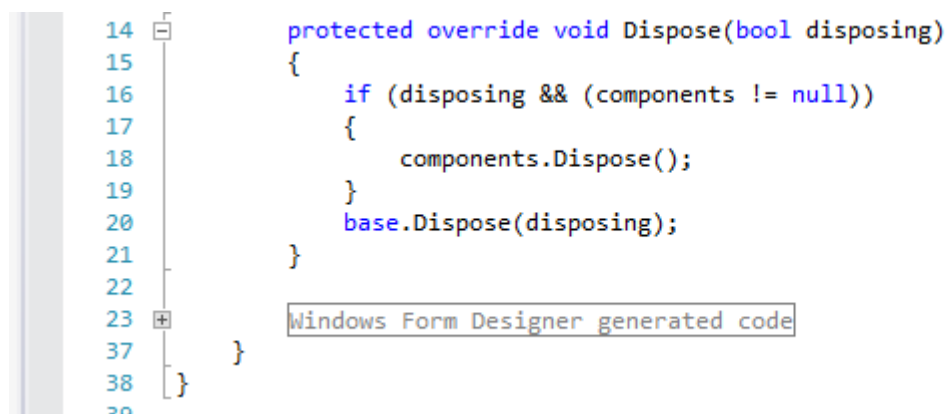
Komponentai į formą įkeliami iš lango *ToolBox*. Šiame lange pele pasirenkamas norimas komponentas ir formos lange nurodoma komponento pozicija (vieta) ir dydis. Vėliau šią poziciją ir dydį galima keisti. Jeigu *ToolBox* ekrane nesimato, jį galima išsikviesti pasirinkus meniu *View->ToolBox*.



Lentelėje pateikta visų formos komponentų vardai, paskirtis ir pakeistos savybės.

| Komponento vardas (savybė Name) | Komponento paskirtis | Pakeistos savybės |
|------------------------------------|--|--|
| Form1 | Programos langas | Text: Stačiakampio plotas |
| label1 | Stačiakampio ilgis | Text: Ilgis Font: Times New Roman, Bold, 16, script: Baltic |
| label2 | Stačiakampio plotis | Text: Plotis Font: Times New Roman, Bold, 16 script: Baltic |
| label3 | Stačiakampio plotas (rezultatui parodyti) | Text: Plotas Font: Times New Roman, Bold, 16 script: Baltic ForeColor: Green |
| textBox1 | Stačiakampio ilgiui užrašyti | Font: Courier New, Bold, 16 ForeColor: Blue Lines: String[] Array, 0 |
| textBox2 | Stačiakampio pločiui užrašyti | Font: Courier New, Bold, 16 ForeColor: Blue Lines: String[] Array, 0 |
| button1 | Duomenims įvesti | Text: Įvesti Font: Times New Roman, Bold, 16 script: Baltic |
| button2 | Skaiciavimams atlikti | Text: Skaiciuoti Font: Times New Roman, Bold, 16 script: Baltic Enabled: false (mygtukas pasyvus) |

Faile **Form1.Designer.cs** galima matyti automatiškai sugeneruotą formos visų komponentų aprašą, paspaudus *Windows Form Designer generated code*.

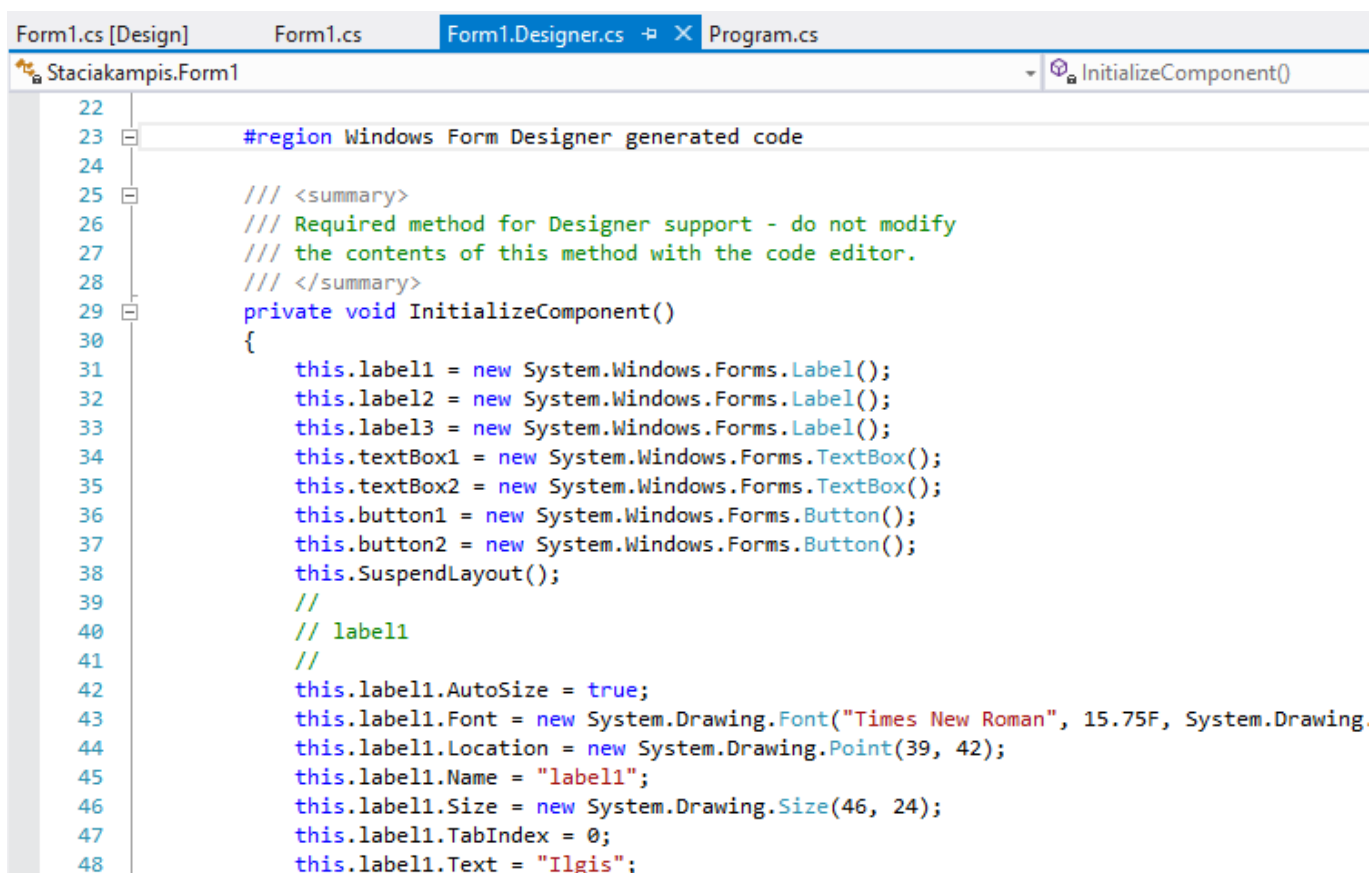


```

14  protected override void Dispose(bool disposing)
15  {
16      if (disposing && (components != null))
17      {
18          components.Dispose();
19      }
20      base.Dispose(disposing);
21  }
22
23  Windows Form Designer generated code
37
38  }
39

```

Šio failo fragmentas pateikiamas žemiau:



```

22
23 #region Windows Form Designer generated code
24
25 /// <summary>
26 /// Required method for Designer support - do not modify
27 /// the contents of this method with the code editor.
28 /// </summary>
29 private void InitializeComponent()
30 {
31     this.label1 = new System.Windows.Forms.Label();
32     this.label2 = new System.Windows.Forms.Label();
33     this.label3 = new System.Windows.Forms.Label();
34     this.textBox1 = new System.Windows.Forms.TextBox();
35     this.textBox2 = new System.Windows.Forms.TextBox();
36     this.button1 = new System.Windows.Forms.Button();
37     this.button2 = new System.Windows.Forms.Button();
38     this.SuspendLayout();
39     //
40     // label1
41     //
42     this.label1.AutoSize = true;
43     this.label1.Font = new System.Drawing.Font("Times New Roman", 15.75F, System.Drawing.
44     this.label1.Location = new System.Drawing.Point(39, 42);
45     this.label1.Name = "label1";
46     this.label1.Size = new System.Drawing.Size(46, 24);
47     this.label1.TabIndex = 0;
48     this.label1.Text = "Ilgis";

```

Failą **Form1.cs** papildome reikalingais kintamaisiais (*ilgis*, *plotis*, *plotas*) ir mygtukų paspaudimo metodais. Mygtuko paspaudimo („Click“) metodas sukuriamas *automatiškai* (!), du kartus spragtelint pele formos lange **Form1.cs[Design]** ant atitinkamo mygtuko:

```

private void button1_Click(object sender, EventArgs e)
{
}

```

Aprašius kintamuosius ir sukūrus mygtukų paspaudimo metodus ir parašius komentarus, failas atrodo taip:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Staciakampis
{
    public partial class Form1 : Form
    {
        /// <summary>
        /// Programos kintamieji
        /// </summary>
        int ilgis, plotis,      // stačiakampio matmenys
            plotas;           // stačiakampio plotas
        /// <summary>
        /// Konstruktorius: formos komponentams suteikia reikšmes
        /// </summary>
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

```

/// <summary>
/// Mygtuko "Įvesti" atliekami veiksmai.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button1_Click(object sender, EventArgs e)
{
    ilgis = Convert.ToInt32(textBox1.Text);
    plotis = Convert.ToInt32(textBox2.Text);
    button2.Enabled = true;
}
/// <summary>
/// Mygtuko "Skaičiuoti" atliekami veiksmai.
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void button2_Click(object sender, EventArgs e)
{
    plotas = ilgis * plotis;
    label3.Text = "Stačiakampio plotas: " + Convert.ToString(plotas);
}
//-----
// Vartotojo metodai
//-----
}
}

```

Mygtuko `button1_Click()` metode yra nuskaityti pradiniai duomenys (stačiakampio ilgis ir plotis), užrašyti komponentuose `textBox1` ir `textBox2`, ir mygtukas `button2` padaromas aktyviu (bus galima atlikti skaičiavimus), savybei `Enabled` suteikiama reikšmė `true`.

Mygtuko `button2_Click()` metode yra apskaičiuojamas stačiakampio plotas ir po to parodoma ši ploto reikšmė ekrane, pakeičiant komponento `label3` savybę `Text`.

Paleidus programą (*Debug->Start Without Debugging*), užrašius norimas reikšmes ir atlikus skaičiavimus programos langas atrodo taip:



Savarankiško darbo užduotis:

- Papildyti programą (klasę `Form1`, už `button2_Click()` metodo) stačiakampio ploto skaičiavimo metodu `Plotas`:

```

/// <summary>
/// Apskaičiuoja stačiakampio plotą
/// </summary>
/// <param name="ilg"> stačiakampio ilgis </param>

```

```

/// <param name="plot"> stačiakampio plotis </param>
/// <returns> gražina stačiakampio plotą </returns>
static int Plotas(int ilg, int plot)
{
    return ilg * plot;
}

```

- Patikrinkite šio metodo veikimą, pakeisdami `buton2_Click()` metodą.
- Papildyti programą mygtuku **Baigti**, jo paspaudimo metode (`Click`) užrašykite kreipinį į metodą: `Close()`;

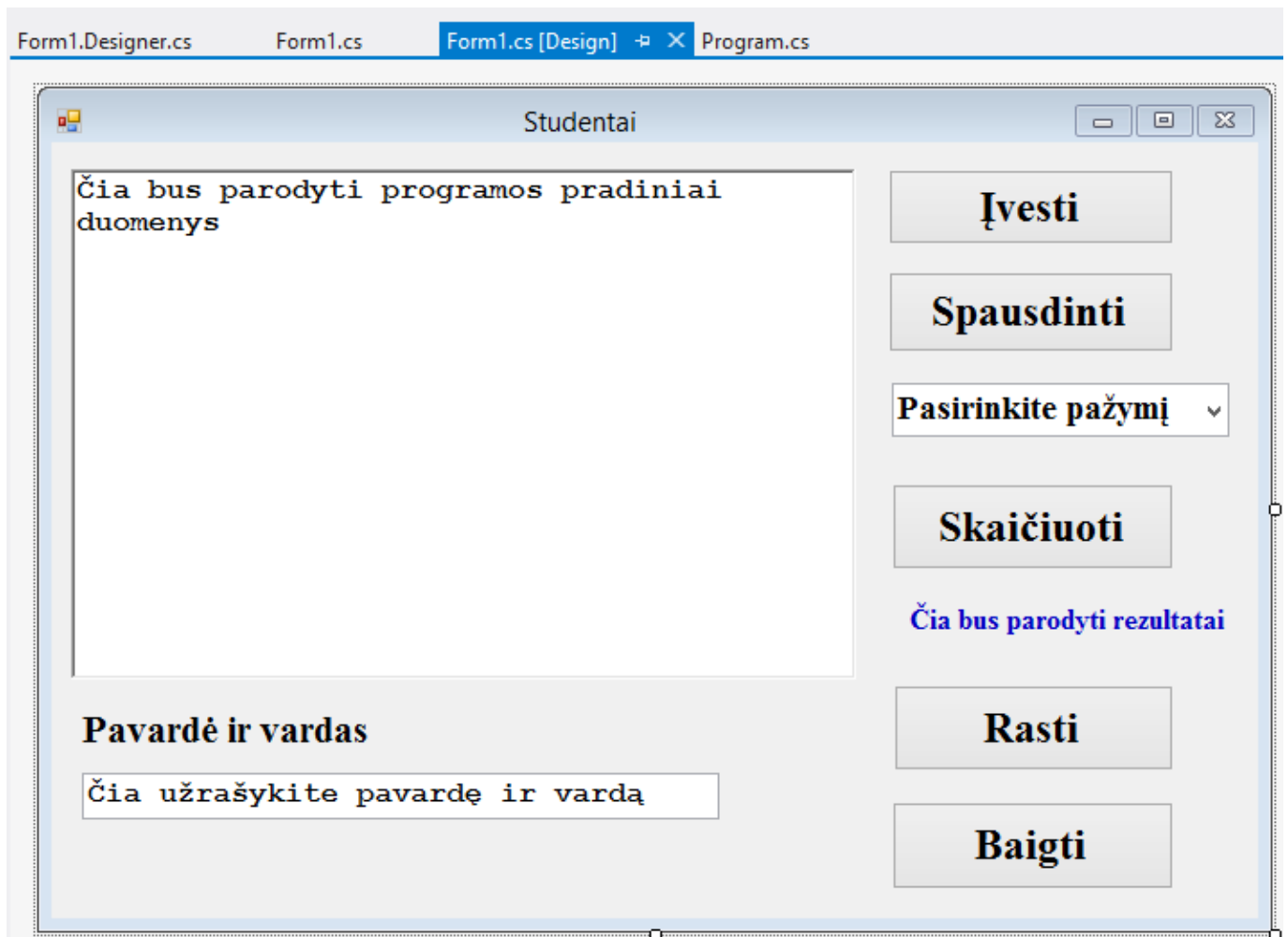
I.3. Sudėtingesnės programos grafinės vartotojo sąsajos kūrimas

Užduotis: Faile "Studentai.txt" duota informacija apie informatikos fakulteto I kurso studentų pažymius (pvz., testo, programavimo žinių patikrinimo, įvertinimai): studento pavardė ir vardas, pažymys. Taip pat yra duota universiteto žinių vertinimo sistema: pažymys ir pažymio žodinė reikšmė.

Sprendimo eiga:

1. Sukurkite grafinę vartotojo sąsają.
2. Sukurkite „tuščius“ mygtukų paspaudimo (`Click`) metodus.
3. Sudarykite reikalingas klases: dvi duomenų klases ir vieną konteinerinę klasę.
4. Sukurkite objektų rinkinį ir konteinerį duomenims saugoti.
5. Sukurkite duomenų failą.
6. Skaitykite duomenis iš duomenų failo (metodas).
7. Spausdinkite duomenis lentele rezultatų faile (metodas).
8. Raskite, kiek studentų gavo nurodytą įvertinimą (metodas). Parodykite ekrane.
9. Raskite, kokį įvertinimą gavo nurodytas studentas (metodas). Parodykite ekrane.

Užduoties sprendimui sukuriame projektą (*Windows Forms Application*), kurį pavadiname `GUI_Studentai`. Į formos `Form1` langą sukeliame komponentus ir jų savybėms suteikiame reikalingas reikšmes.



Iš failo `Form1.Designer.cs` fragmento matosi, kokio tipo komponentai yra įdėti į formą ir kokie vardai (pakeista savybė `Name`) jiems suteikti. Kaip vėliau matysite, komponentų vardų pakeitimas prasmingais vardais labai padeda rašant programą.

```

Start - Debug - Any CPU
Form1.Designer.cs Form1.cs Form1.cs [Design] Program.cs
GUI_Studentai.Form1
148 //
149 // Form1
150 //
151 this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
152 this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
153 this.ClientSize = new System.Drawing.Size(672, 432);
154 this.Controls.Add(this.baigti);
155 this.Controls.Add(this.pavardeVrd);
156 this.Controls.Add(this.pavardeVardas);
157 this.Controls.Add(this.rasti);
158 this.Controls.Add(this.skaiciuoti);
159 this.Controls.Add(this.rezultatas);
160 this.Controls.Add(this.vertinimai);
161 this.Controls.Add(this.spausdinti);
162 this.Controls.Add(this.iversti);
163 this.Controls.Add(this.rezultatai);
164 this.Name = "Form1";
165 this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
166 this.Text = "Studentai";
167 this.ResumeLayout(false);
168 this.PerformLayout();
169
170 }
171
172 #endregion
173
174 private System.Windows.Forms.RichTextBox rezultatai;
175 private System.Windows.Forms.Button iversti;
176 private System.Windows.Forms.Button spausdinti;
177 private System.Windows.Forms.Button skaiciuoti;
178 private System.Windows.Forms.Button rasti;
179 private System.Windows.Forms.Button baigti;
180 private System.Windows.Forms.ComboBox vertinimai;
181 private System.Windows.Forms.Label rezultatas;
182 private System.Windows.Forms.Label pavardeVardas;
183 private System.Windows.Forms.TextBox pavardeVrd;
184
185 }

```

Lentelėje pateikta visų formos komponentų vardai, paskirtis ir pakeistos savybės.

| Komponento vardas (savybė Name) | Komponento paskirtis | Pakeistos savybės |
|------------------------------------|--|---|
| Form1 | Programos langas | Text: Studentai |
| rezultatai (tipas RichTextBox) | Atvaizduoti failo turinį | Font: Courier New, Bold, 12 Script: Baltic |
| iversti (tipas Button) | Įvesti duomenis | Text: Įvesti Font: Times New Roman, Bold, 18 Script: Baltic |
| spausdinti (tipas Button) | Spausdinti duomenis | Text: Spausdinti Font: Times New Roman, Bold, 18 Script: Baltic |
| skaiciuoti (tipas Button) | Suskaiciuoti ir parodyti studentų kiekį | Text: Skaičiuoti Font: Times New Roman, Bold, 18 Script: Baltic |
| rasti (tipas Button) | Rasti nurodyto studento įvertinimą ir jį parodyti | Text: Rasti Font: Times New Roman, Bold, 18 |

| | | |
|--------------------------------|---|---|
| | | Script: Baltic |
| baigti (tipas Button) | Uždaryti programos langą | Text: Baigti Font: Times New Roman, Bold, 18 Script: Baltic |
| vertinimai (tipas ComboBox) | Pažymio pasirinkimui iš sąrašo | Text: Pasirinkite pažymį Font: Times New Roman, Bold, 14 Script: Baltic |
| rezultatas (tipas Label) | Suskaičiuoto kiekio parodymui | Text: Čia bus parodyti rezultatai Font: Times New Roman, Bold, 12 Script: Baltic ForeColor: Blue |
| pavardeVardas (tipas Label) | Paaiškinamasis užrašas virš komponento pavardeVrd | Text: Pavardė ir vardas Font: Times New Roman, Bold, 16 Script: Baltic |
| pavardeVrd (tipas textBox1) | Studento pavardei ir vardui užrašyti | Text: Čia užrašykite pavardę ir vardą Font: Courier New, Bold, 12 Script: Baltic |

Sukūrus programos valdymo grafinę sąsają galima pereiti prie mygtukų paspaudimo metodų sukūrimo. Prisiminkite iš ankstesnės užduoties, kaip ir kokiame faile buvo kuriami tokie metodai. Sugeneruotą šių metodų kamieną užpildysime turiniu vėliau, kai turėsime tam reikalingas priemones (klases, objektus, metodus). Neužmirškite pakomentuoti atskirų mygtukų paspaudimo metodų.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
//-----
using System.IO; // reikalinga darbui su failais
//-----
namespace GUI_Studentai
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            //-----
            // Nurodyti mygtukai padaromi pasyviais
            spausdinti.Enabled = false;
            skaiciuoti.Enabled = false;
            rasti.Enabled = false;
            //-----
        }
        //=====
        // GRAFINĖS SĄSAJOS VALDYMO METODAI
        //=====

        /// <summary>
        /// Mygtuko "Ivesti" atliekami veiksmai
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void iversti_Click(object sender, EventArgs e)
        {
            // užpildysime vėliau
        }

        /// <summary>
```

```

    /// Mygtuko "Spausdinti" atliekami veiksmai
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void spausdinti_Click(object sender, EventArgs e)
    {
        // užpildysime vėliau
    }

    /// <summary>
    /// Mygtuko "Skaičiuoti" atliekami veiksmai
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void skaiciuoti_Click(object sender, EventArgs e)
    {
        // užpildysime vėliau
    }

    /// <summary>
    /// Mygtuko "Rasti" atliekami veiksmai
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void rasti_Click(object sender, EventArgs e)
    {
        // užpildysime vėliau
    }

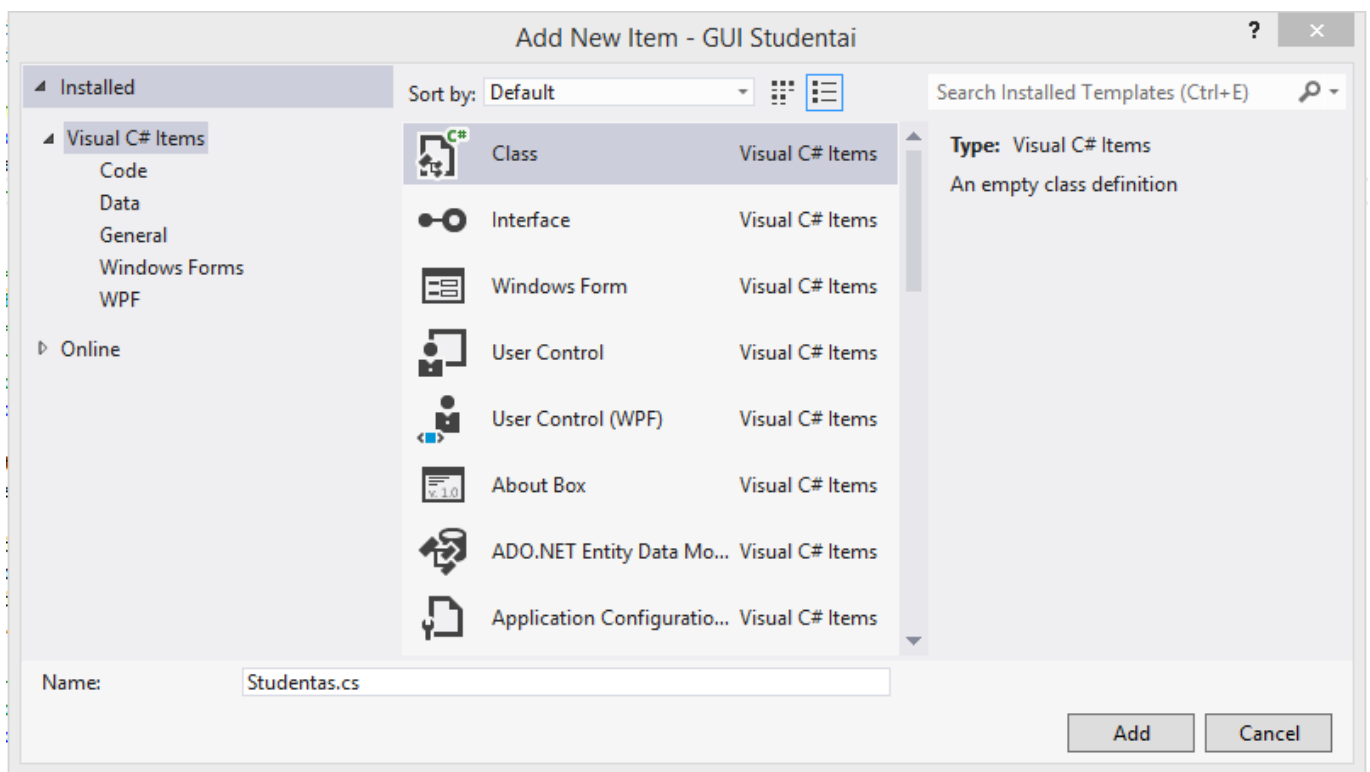
    /// <summary>
    /// Mygtuko "Baigti" atliekami veiksmai
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void baigti_Click(object sender, EventArgs e)
    {
        // užpildysime vėliau
    }
    //=====
}
}

```

Išbandykite programos veikimą. Pastebėjote, kad paspaudus bet kurį mygtuką, niekas neįvyksta. Tai yra dėl to, kad jokie veiksmai mygtukų paspaudimo metoduose kol kas neatliekami.

Toliau kursime reikalingas klases uždavinio sprendimui. *Kiekvieną klasę dėsime į atskirą failą.* Tai pademonstruosime su viena klase. Kitas klases į atskirus failus sudėsite savarankiškai.

Tam tikslui **Solution Explorer** lange virš projekto **GUI Studentai** paspaudus dešinįjį pelės klavišą ir pasirinkus iš atsivėrusio kontekstinio meniu punktą **Add**, o tolimesniame submeniu **class...**, pasirodys langas, kuriame užrašysime kuriamos klasės failo pavadinimą **Studentas.cs**.



Sukurtame klasės šablone užrašysime klasės kintamuosius, konstruktorių ir užklotą metodą ToString().

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace GUI_Studentai
{
    /// <summary>
    /// Klasė vieno studento duomenims saugoti
    /// </summary>
    class Studentas
    {
        public string PavVrd { get; set; } // savybė: studento pavardė ir vardas
        public int Pazym { get; set; } // savybė: pažymys (įvertinimas)
        /// <summary>
        /// Klasės konstruktorius: savybėms suteikia reikšmes
        /// </summary>
        /// <param name="pavv"> pavardė ir vardas </param>
        /// <param name="pazym"> pažymys </param>
        public Studentas(string pavv, int pazym)
        {
            PavVrd = pavv;
            Pazym = pazym;
        }
        /// <summary>
        /// Užklotas metodas ToString()
        /// </summary>
        /// <returns> grąžina suformatuotą eilutę </returns>
        public override string ToString()
        {
            string eilute;
            eilute = string.Format("{0, -20} {1, 2}", PavVrd, Pazym);
            return eilute;
        }
    }
}
```

Analogiškai sukursime ir kitus du failus. Pirmajame faile (**Studentai.cs**) aprašysime konteinerinę klasę Studentai studentų duomenims saugoti, antrajame faile (**Pazymys.cs**) – klasę Pazymys vieno pažymio aprašui saugoti.

Failas **Studentai.cs**:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

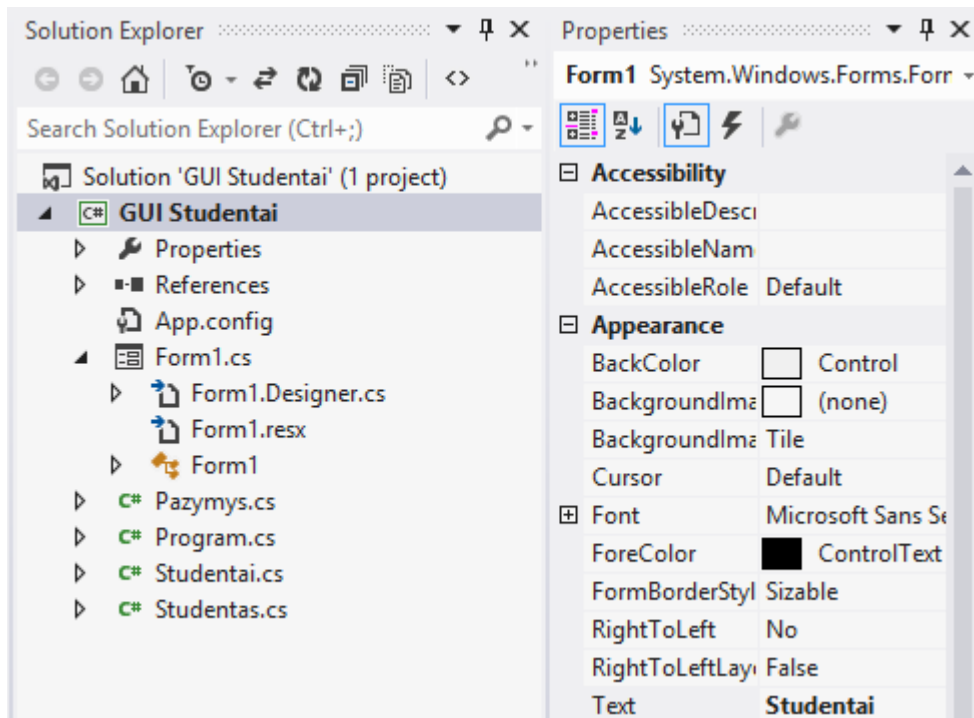
namespace GUI_Studentai
{
    /// <summary>
    /// Klasė (KONTEINERINĖ) studentų duomenims saugoti
    /// </summary>
    class Studentai
    {
        const int Cn = 500;           // studentų masyvo dydis
        private Studentas[] Stud;     // studentų objektų masyvas
        public int Kiek { get; set; }  // savybė: studentų skaičius
        /// <summary>
        /// Klasės konstruktorius: suteikia kintamiesiems reikšmes
        /// </summary>
        public Studentai()
        {
            Kiek = 0;
            Stud = new Studentas[Cn];
        }
        /// <summary>
        /// Gražina nurodyto indekso studento objektą.
        /// </summary>
        /// <param name="i"> indeksas </param>
        /// <returns> gražina studento objektą </returns>
        public Studentas ImtiStudenta(int i) { return Stud[i]; }
        /// <summary>
        /// Įrašo į studentų objektų masyvą naują studentą.
        /// </summary>
        /// <param name="stud"> studentas </param>
        public void DetiStudenta(Studentas stud) { Stud[Kiek++] = stud; }
    }
}
```

Failas **Pazymys.cs**:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace GUI_Studentai
{
    /// <summary>
    /// Klasė pažymio duomenims saugoti
    /// </summary>
    class Pazymys
    {
        public int Pazym { get; set; } // savybė: pažymys (skaičius: 1..10)
        public string PazZodR { get; set; } // savybė: pažymio žodinė reikšmė
        /// <summary>
        /// Klasės konstruktorius: savybėms suteikia reikšmes.
        /// </summary>
        /// <param name="paz"> pažymys </param>
        /// <param name="pazR"> pažymio žodinė reikšmė </param>
        public Pazymys(int paz, string pazR)
        {
            Pazym = paz;
            PazZodR = pazR;
        }
    }
}
```

Solution Explorer lange matosi visi į projektą (GUI Studentai) naujai įtraukti failai:



Toliau failą **Form1.cs** papildysime failų vardų konstantomis, objektų konteineriu, pažymių objektų masyvu ir pereisime prie 4 užduotyje išvardintų ir programos veikimui reikalingų metodų kūrimo. Pateiksime programos fragmentus.

```
namespace GUI_Studentai
{
    //-----
    public partial class Form1 : Form
    {
        /// <summary>
        /// KONSTANTOS
        /// </summary>
        const string CFd = "..\\..\\Studentai.txt"; // duomenų failo vardas
        const string CFr = "..\\..\\Rezultatai.txt"; // rezultatų failo vardas
        /// <summary>
        /// KINTAMIEJI (OBJEKTAI, OBJEKTŲ MASYVAI)
        /// </summary>
        Studentai TestasMas; // studentų testo rezultatai (Konteineris)
        Pazymys [] Pazymiai = new Pazymys[10] // pažymių objektų masyvas
        {
            new Pazymys(10, "Puikiai"),
            new Pazymys(9, "Labai gerai"),
            new Pazymys(8, "Gerai"),
            new Pazymys(7, "Vidutiniškai"),
            new Pazymys(6, "Patenkinamai"),
            new Pazymys(5, "Silpnai"),
            new Pazymys(4, "Nepatenkinamai"),
            new Pazymys(3, "Nepatenkinamai"),
            new Pazymys(2, "Nepatenkinamai"),
            new Pazymys(1, "Nepatenkinamai")
        };
        //-----
        public Form1()
        {
            InitializeComponent();
            ...
        }
        ...
    }
}
```

```

/// <summary>
/// Mygtuko "Baigti" atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void baigti_Click(object sender, EventArgs e)
{
    // užpildysime vėliau
}
//=====
// Čia užrašysime 4 savo sukurtus metodus.
//=====
}

```

Pirmasis, duomenų skaitymo, metodas:

```

/// <summary>
/// Skaityti visus duomenis iš failo į konteinerį.
/// </summary>
/// <param name="fv"> duomenų failo vardas </param>
/// <returns> gražina suformuoto konteinerio nuorodą </returns>
static Studentai SkaitytiStudKont(string fv)
{
    Studentai StudentaiKont = new Studentai();
    using (StreamReader srautas = new StreamReader(fv,
        Encoding.GetEncoding(1257)))
    {
        string eilute; // visa duomenų failo eilutė
        while ((eilute = srautas.ReadLine()) != null)
        {
            string[] eilDalis = eilute.Split(';');
            string pavVrd = eilDalis[0];
            int pazym = int.Parse(eilDalis[1]);
            Studentas studentas = new Studentas(pavVrd, pazym);
            StudentaiKont.DetiStudenta(studentas);
        }
    }
    return StudentaiKont;
}

```

Antrasis, duomenų spausdinimo, metodas:

```

/// <summary>
/// Spausdina konteinerio duomenis lentelė faile.
/// </summary>
/// <param name="fv"> rezultatų failo vardas </param>
/// <param name="StudentaiKont"> studentų objektų konteineris </param>
/// <param name="antraste"> užrašas virš lentelės </param>
static void SpausdintiStudKont(string fv, Studentai StudentaiKont,
    string antraste)
{
    const string virsus =
        "-----\r\n"
        + " Nr.   Pavardė ir vardas      Pažymys \r\n"
        + "-----";

    using (var fr = new StreamWriter(File.Open(fv, FileMode.Append),
        Encoding.GetEncoding(1257)))
    {
        fr.WriteLine("\n " + antraste);
        fr.WriteLine(virsus);
        for (int i = 0; i < StudentaiKont.Kiek; i++)
        {
            Studentas stud = StudentaiKont.ImtiStudenta(i);
            fr.WriteLine("{0, 3}   {1}", i + 1, stud);
        }
        fr.WriteLine("-----\n");
    }
}

```


Trečiasis, kiekio skaičiavimo, metodas:

```
/// <summary>
/// Suskaičiuoja studentų, kurių pažymiai lygūs nurodytam pažymiui, skaičių.
/// </summary>
/// <param name="StudentaiKont"> studentų objektų konteineris </param>
/// <param name="pazymys"> nurodytas pažymys </param>
/// <returns> gražina suskaičiuotą studentų skaičių </returns>
static int Kiekis(Studentai StudentaiKont, int pazymys)
{
    int kiek = 0;
    for (int i = 0; i < StudentaiKont.Kiek; i++)
    {
        Studentas stud = StudentaiKont.ImtiStudenta(i);
        if (stud.Pazym == pazymys)
            kiek++;
    }
    return kiek;
}
```

Ketvirtasis, studento paieškos, metodas:

```
/// <summary>
/// Ieško nurodytos pavardės ir vardo studento.
/// </summary>
/// <param name="StudentaiKont"> studentų objektų konteineris </param>
/// <param name="pavVrd"> studento pavardė ir vardas </param>
/// <returns> gražina studento indeksą arba -1,
///           jeigu ieškomo studento nėra </returns>
static int StudentoIndeksas(Studentai StudentaiKont, string pavVrd)
{
    for (int i = 0; i < StudentaiKont.Kiek; i++)
    {
        Studentas stud = StudentaiKont.ImtiStudenta(i);
        if (stud.PavVrd == pavVrd)
            return i;
    }
    return -1;
}
```

Dabar galima sugrįžti prie mygtukų paspaudimų (Click) metodų sudarymo ir jų kamienus užpildyti reikalingais veiksmiais.

```
//=====
// GRAFINĖS SAŠAJOS VALDYMO METODAI
//=====

/// <summary>
/// Mygtuko "Ivesti" atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void iversti_Click(object sender, EventArgs e)
{
    rezultatai.LoadFile(CFd, RichTextBoxStreamType.PlainText);
    TestasMas = SkaitytiStudKont(CFd);

    iversti.Enabled = false;
    spausdinti.Enabled = true;
    skaiciuoti.Enabled = true;
    rasti.Enabled = true;
}

/// <summary>
/// Mygtuko "Spausdinti" atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void spausdinti_Click(object sender, EventArgs e)
{
    SpausdintiStudKont(CFr, TestasMas,
```

```

        "Studentų sąrašas (testo rezultatai)");
    rezultatai.LoadFile(CFr, RichTextBoxStreamType.PlainText);
    vertinimai.SelectedIndex = 0; // parenkama 1-oji reikšmė
}
/// <summary>
/// Mygtuko "Skaiciuoti" atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void skaiciuoti_Click(object sender, EventArgs e)
{
    string ivertis = vertinimai.SelectedItem.ToString();
    string[] eilDalis = ivertis.Split(' ');
    int pazymys = Int32.Parse(eilDalis[0]);
    int kiekis = Kiekis(TestasMas, pazymys);
    if (kiekis > 0)
        rezultatas.Text = "Studentų skaičius: " + kiekis.ToString();
    else
        rezultatas.Text = "Tokių studentų nėra.";
}
/// <summary>
/// Mygtuko "Rasti" atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void rasti_Click(object sender, EventArgs e)
{
    pavardeVardas.Text = "Pavardė ir vardas"; // pakartotinam paspaudimui
    string pavVrd = pavardeVrd.Text;
    int index = StudentoIndeksas(TestasMas, pavVrd);
    if (index > -1)
    {
        Studentas stud = TestasMas.ImtiStudenta(index);
        int pazymys = stud.Pazym;
        pavardeVardas.Text = pavardeVardas.Text + " (pažymys: "
            + pazymys.ToString() + ")";
    }
    else
        pavardeVardas.Text = pavardeVardas.Text +
            " (Tokio studento (-ės) nėra.)";
}
/// <summary>
/// Mygtuko "Baigti" atliekami veiksmai
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void baigti_Click(object sender, EventArgs e)
{
    Close();
}
//=====

```

Kad programa gerai veiktų, dar reikia klasės Form1 konstruktorių papildyti rezultatų failo išvalymu ir žinių vertinimo sistemos įkėlimu į komponentą vertinimai (ComboBox).

```

//-----
public Form1()
{
    InitializeComponent();
    //-----
    // Nurodyti mygtukai padaromi pasyviais
    spausdinti.Enabled = false;
    skaiciuoti.Enabled = false;
    rasti.Enabled = false;

    if (File.Exists(CFr))
        File.Delete(CFr);

    // Komponento vertinimai užpildymas pažymiais

```

```

foreach (Pazymys paz in Pazymiai)
    vertinimai.Items.Add(paz.Pazym + " " + paz.PazZodR);
//-----
}

```

Sukurkime duomenų failo **Sudentai.txt** pavyzdį:

```

Jonaitis Jonas;      8;
Petraitis Petras;    7;
Antanaitis Antanas;  10;
Giedraitis Giedrius; 5;
Onaitytė Ona;        8;
Juozaitis Juozas;    4;
Ramunaitė Ramunė;    5;

```

Paleiskime ir išbandykime programos veikimą. Pavyzdžiui, vertinimo sistemoje pasirinkus *8 Gerai*, studentų skaičių gavusių tokį įvertinimą bus 2, o pavardės ir vardo laukelyje užrašius *Antanaitis Antanas* bus parodytas jo gautas pažymys *10*.

The screenshot shows a Windows application window titled "Studentai". Inside the window, there is a table titled "Studentų sąrašas (testo rezultatai)". The table has three columns: "Nr.", "Pavardė ir vardas", and "Pažymys". The table contains seven rows of student data. To the right of the table, there are several buttons: "Įvesti", "Spausdinti", "Skaičiuoti", "Rasti", and "Baigti". Below the "Skaičiuoti" button, there is a label "Studentų skaičius: 2". Below the "Rasti" button, there is a text input field containing "Antanaitis Antanas".

| Nr. | Pavardė ir vardas | Pažymys |
|-----|---------------------|---------|
| 1 | Jonaitis Jonas | 8 |
| 2 | Petraitis Petras | 7 |
| 3 | Antanaitis Antanas | 10 |
| 4 | Giedraitis Giedrius | 5 |
| 5 | Onaitytė Ona | 8 |
| 6 | Juozaitis Juozas | 4 |
| 7 | Ramunaitė Ramunė | 5 |

Buttons: Įvesti, Spausdinti, 8 Gerai (dropdown), Skaičiuoti, Studentų skaičius: 2, Rasti, Baigti

Search field: Pavardė ir vardas (pažymys: 10) Antanaitis Antanas

Čia buvo pateiktas vienas galimas, mūsų manymu, paprasčiausias programos su grafine vartotojo sąsaja kūrimo kelias. Galimi ir kitokie sprendimo eigos (programos rašymo) eiliškumo variantai. Taip pat galimi naudojimui ir kitokie komponentai bei jų išdėstymas formoje. Gera (patogi, suprantama) programos vartotojo sąsaja nereikalauja programuotojo tolimesnio dalyvavimo jos eksploatavime.