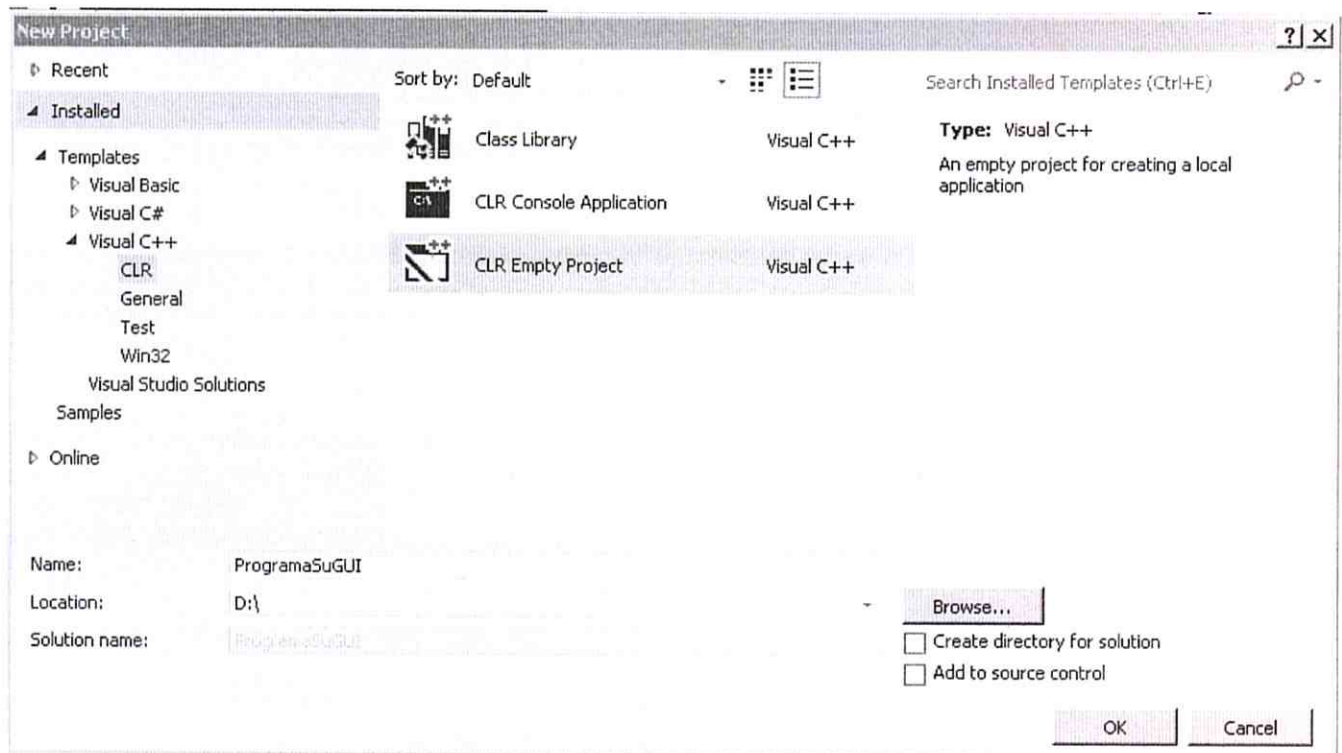


Kaip sukurti Windows formą MVS 2012 versijoje

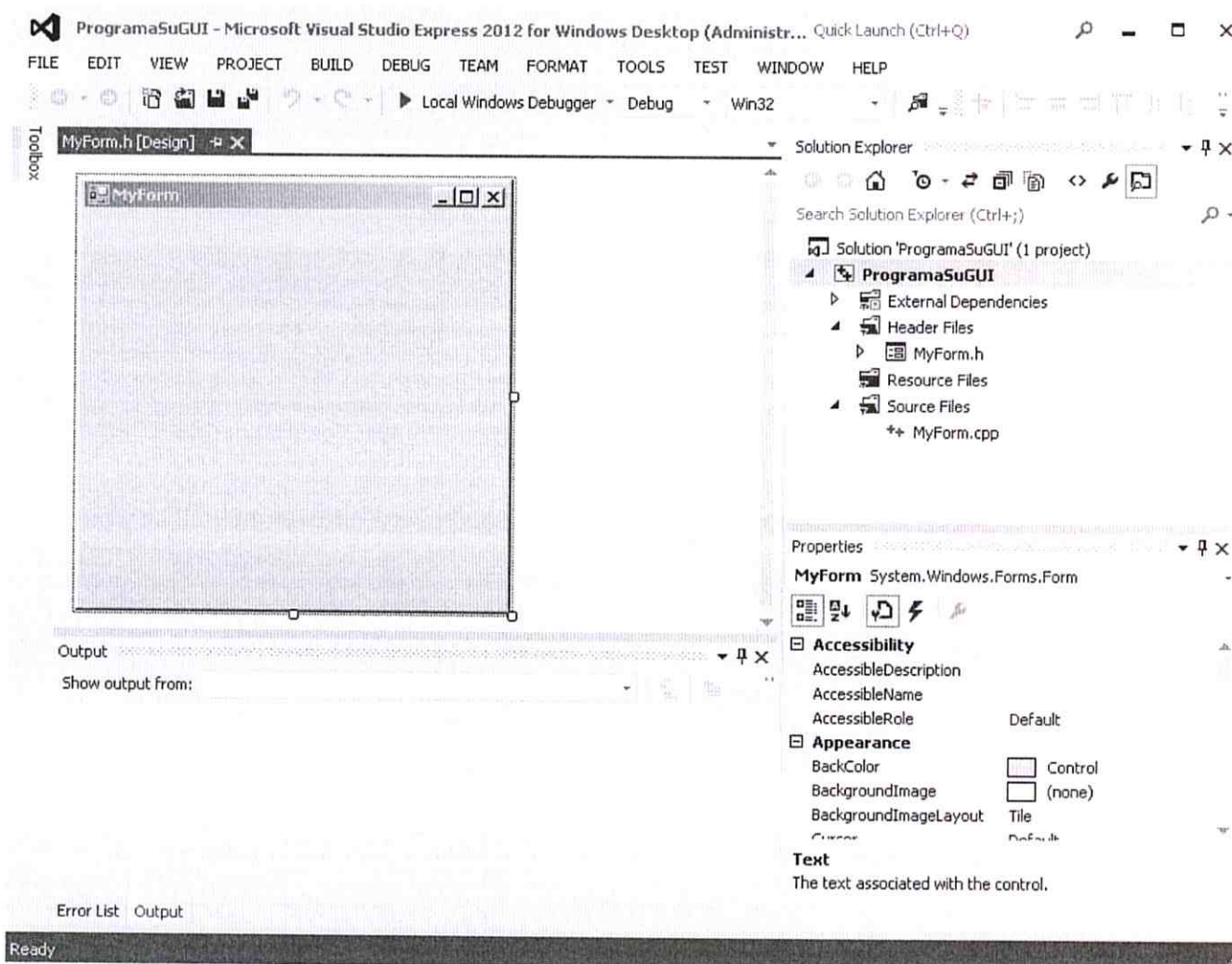
1. Meniu FILE → New Project → Visual C++ → CLR → CLR Empty Project → parenkama projekto saugojimo vieta ir projekto (programos) vardas (pvz.: **ProgramaSuGUI**).



2. Meniu PROJECT → Add New Item → UI → Windows Form (vienintelis pasirinkimas) → Name → MyForm.h (pasiūlomas automatiškai, tačiau šis vardas gali būti ir kitoks, pvz.: Langas.h) → Add.



3. Sukuriama forma (langas) MyForm:



4. Automatiškai sukurto failo MyForm.h tekstas (kuris parodomas paspaudus ant formos dešinijį pelės klavišą ir pasirinkus View Code):

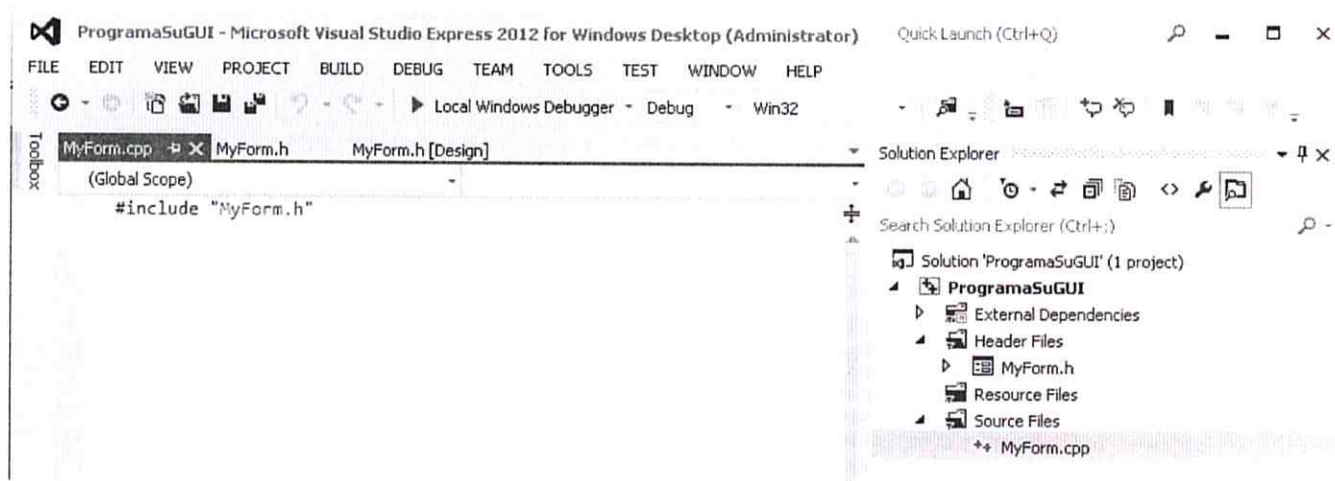
```
#pragma once
namespace ProgramaSuGUI {
    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;
    /// <summary>
    /// Summary for MyForm
    /// </summary>
    public ref class MyForm : public System::Windows::Forms::Form
    {
    public:
        MyForm(void)
        {
            InitializeComponent();
            //
            //TODO: Add the constructor code here
            //
        }

    protected:
        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        ~MyForm()
    }
```

```

        {
            if (components)
            {
                delete components;
            }
        }
private:
    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container ^components;
#pragma region Windows Form Designer generated code
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        this->components = gcnew System::ComponentModel::Container();
        this->Size = System::Drawing::Size(300,300);
        this->Text = L"MyForm";
        this->Padding = System::Windows::Forms::Padding(0);
        this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;
    }
#pragma endregion
    };
}
5. Papildyti failą MyForm.cpp
#include "MyForm.h"

```



tekstu:

```

//-----
using namespace System;
using namespace System::Windows::Forms;

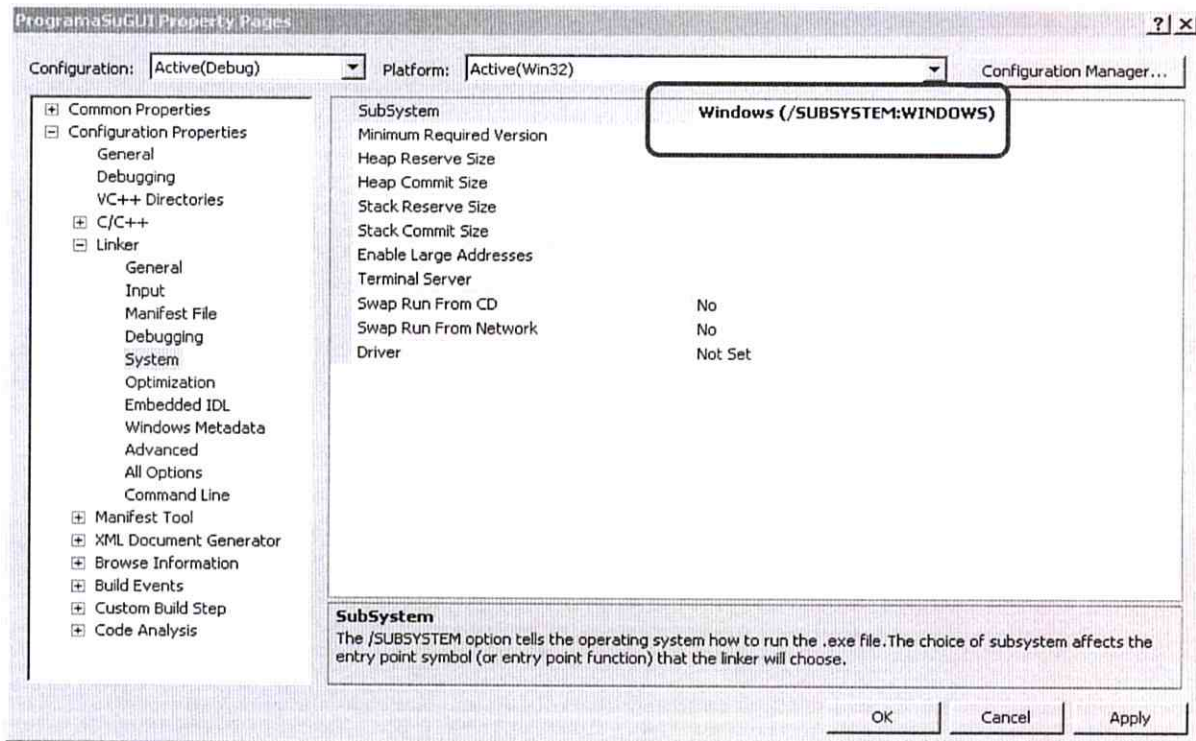
[STAThread]
int main(array<System::String ^> ^args)
{
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    ProgramaSuGUI::MyForm form; // Eilutė, kuri gali būti koreguojama
    Application::Run(%form);
}
//-----

```

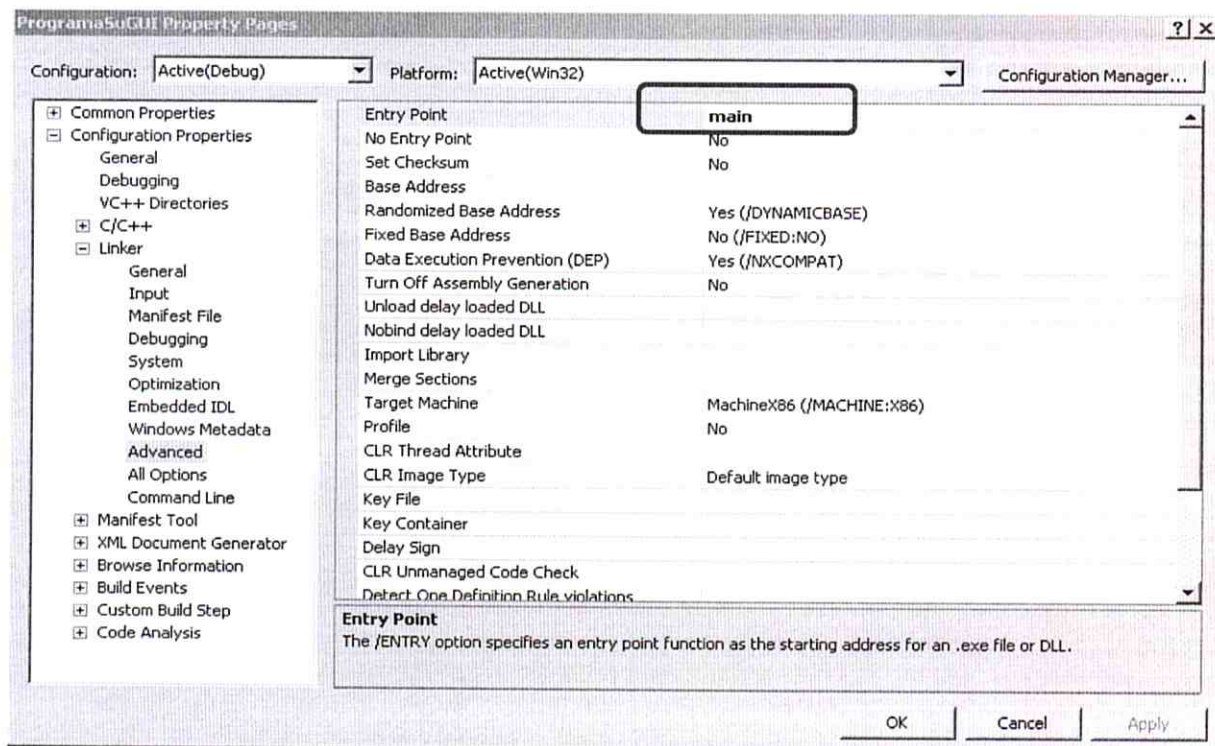
Pastaba: Eilutėje, kuri gali būti koreguojama, vietoje paryškinto žodžio reikia užrašyti namespace vardą ProgramaSuGUI, kuris yra MyForm.h faile. Šis vardas dažniausiai sutampa su projekto vardu.

6. Meniu PROJECT → ProgramaSuGUI Properties:

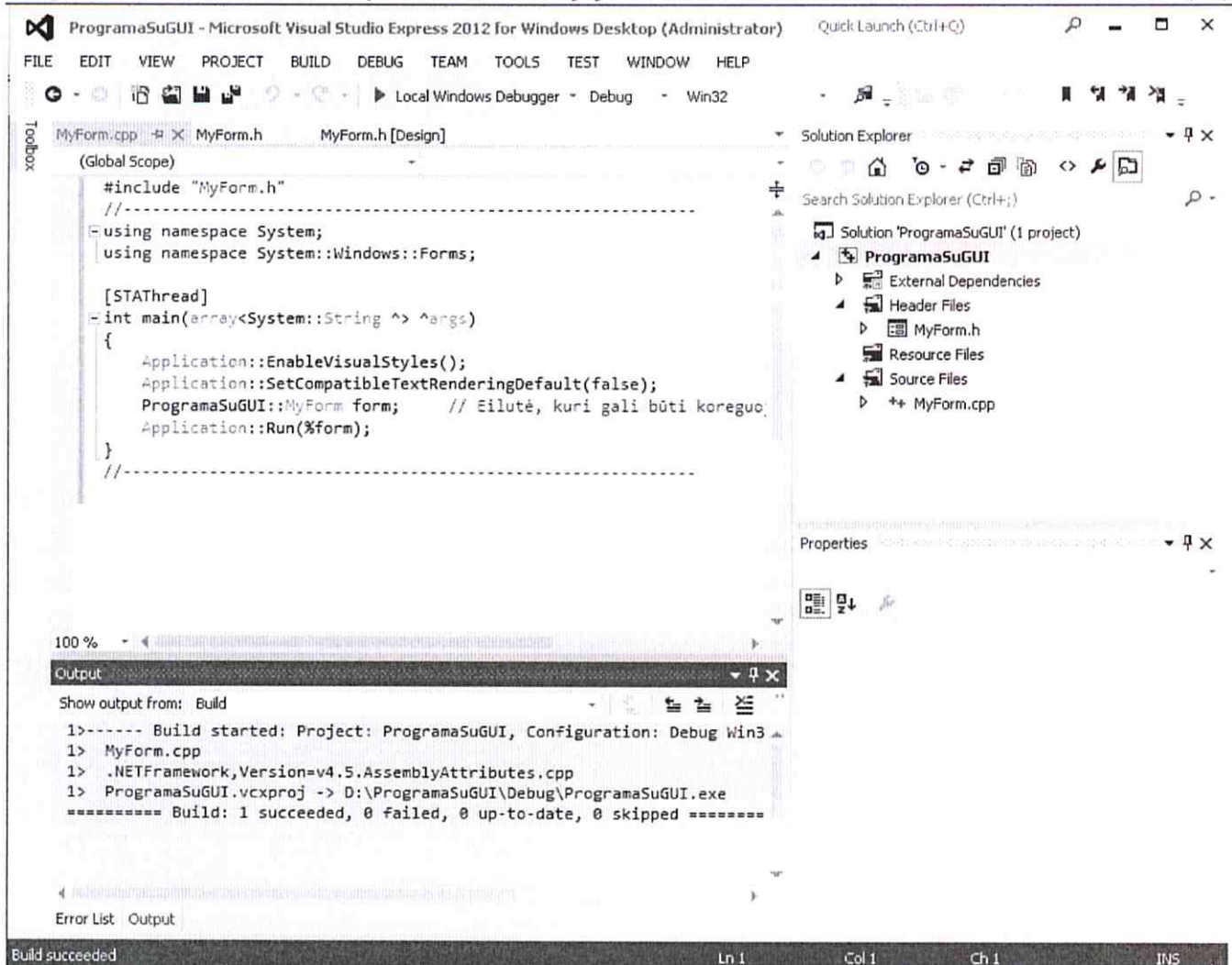
6.1 Linker → System → SubSystem (žiūr. pav.) → Paspusti mygtuką Apply → Paspusti mygtuką OK.



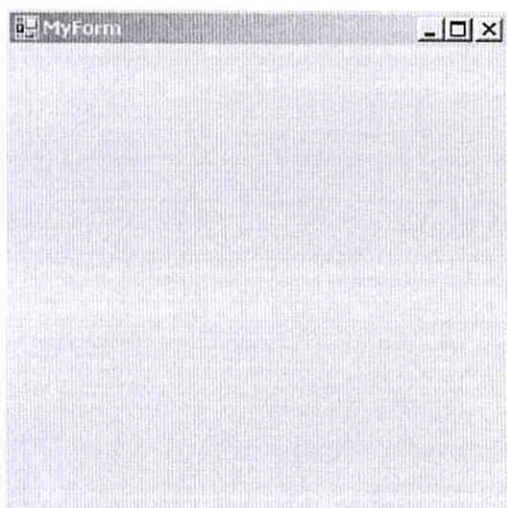
6.2 Linker → Advanced → Entry Point → **main** (kaip programoje) → Paspusti mygtuką OK (žiūr. pav.).



7. Kompiliuoti programą (Meniu BUILD → Build Solution).



8. Vykdyti programą (Meniu DEBUG → Start Without Debugging). Sukuriamas programos (projekto) grafinis langas, turintis visas savybes - jį galima didinti, mažinti, uždaryti:



9. Baigti darbą su programa (Meniu FILE → Close Solution).

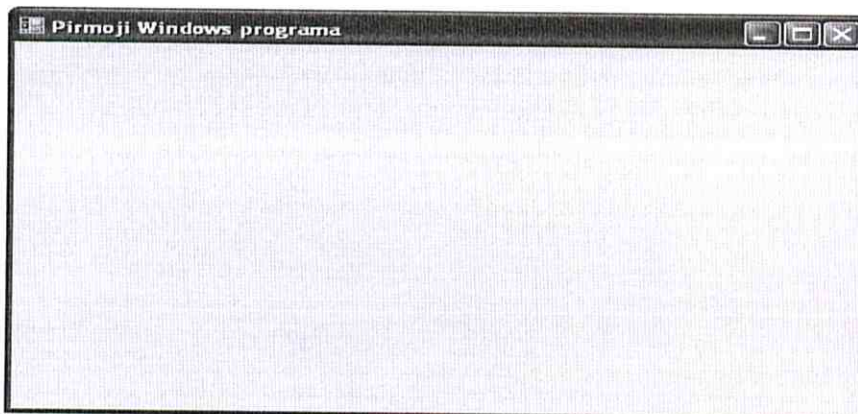
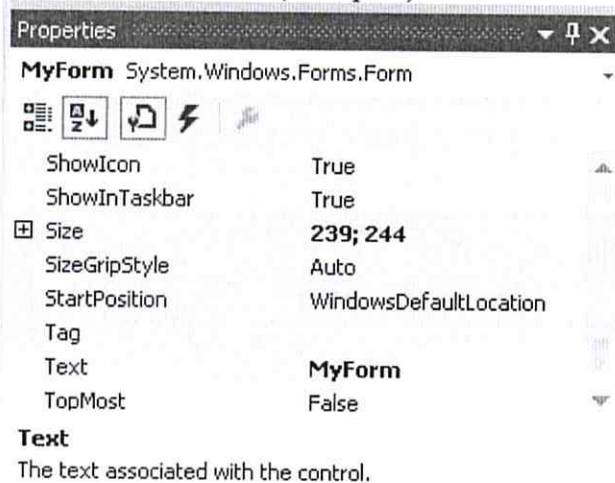
Formos savybių (Properties) keitimas

Pagrindinis veiksmas kurį galima daryti su forma yra jos išmatavimų (aukščio ir pločio) keitimas. Formos išmatavimai keičiami pelės pagalba, nuspaudus kairįjį klavišą, tempiant už šalia formos ribų esančių „auselių“. Tempimo kryptį rodo pasikeitusi pelės rodyklė. Norint pakeisti kitas formos savybes (spalvą, užrašą ir pan.) reikia pasinaudoti savybių (properties) keitimo langu, esančiu apatiniam dešiniajame kampe. Savybių *Properties* lange yra išvardintos visos formos savybės ir jų numatytosios reikšmės, kurias galima keisti (žiūr. pav.).

Pavyzdžiui, reikia pakeisti formos antraštę *MyForm* į *Pirmoji Windows programa*, formą nudažyti geltona spalva, o formos išmatavimus pakeisti tokiais: plotis – 500, aukštis – 350.

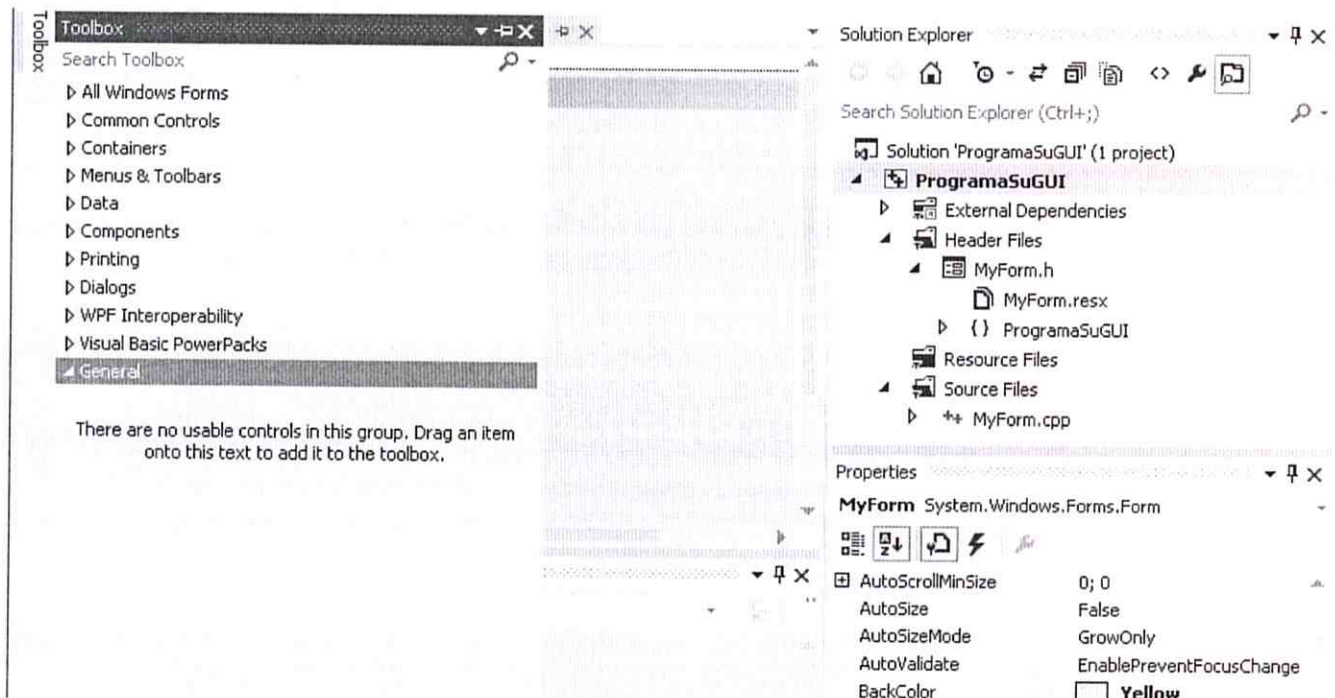
Šiems veiksmams atlikti *Properties* lange reikia atitinkamai pakeisti tris savybes: *Text* (*Pirmoji Windows programa*), *BackColor* (*Yellow*) ir *Size* (*500;350*)

Norint pamatyti kaip atrodys veikianti programa ją reikia išsaugoti (menu komanda *File→Save All*), sukompiliuoti (menu komanda *Build→Build Solution*) ir įvykdyti (menu komanda *Debug→Start Without Debugging*). Tai atlikus ekrane pasirodo „tuščias“ geltonos spalvos langas su antrašte *Pirmoji Windows programa* (žiūr. pav.).

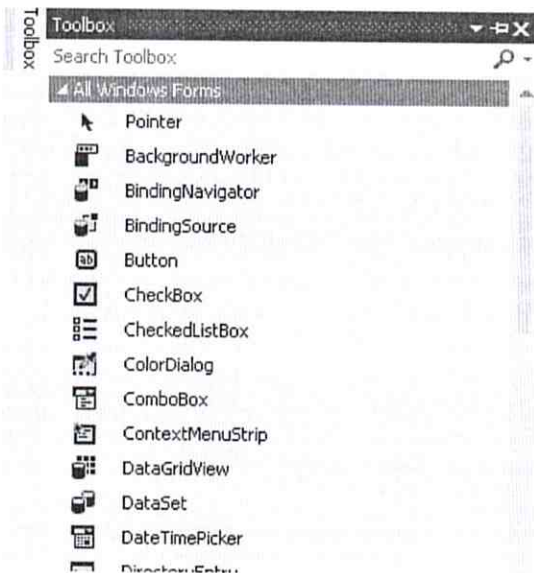


Komponentų įkėlimas į formą

Sukurtą formą galima papildyti reikalingais komponentais. Visus komponentus galima rasti **ToolBox** lange, kurį galima iškviesti meniu komanda **View→ToolBox** (žiūr. pav.).



Šiame **Toolbox** lange komponentai yra sugrupuoti į atskiras grupes pagal paskirtį. Visus komponentus galima pamatyti paspaudus pele ženklelį **+** ties užrašu **All Windows Forms** (žiūr. pav.).



Pavyzdžiui, į formą reikia įkelti du komponentus: **Label** (Etiketė) ir **Button** (Mygtukas), pakeisti pageidaujamas jų savybes ir įvykdyti programą.

Nurodytiems veiksams atlikti reikia **Toolbox** lange surasti šiuos komponentus ir juos po vieną išsirinkus formoje pele parodyti vietą, kurioje juos norime padėti (Pastaba: vėliau jų padėtį formoje bus galima keisti). Kuomet komponentai atsiduria formoje jie yra įvardijami ir sunumeruojami pradedant nuo vienetų: **label1** ir **button1**.

Komponentų savybes, analogiškai kaip ir formos savybes, galima keisti. Tam tikslui reikia pele pažymėti atitinkamą komponentą, paspausti dešinįjį pelės klavišą ir atsiradusiame kontekstiniame meniu iškviesti **Properties** langą. Šiame lange ir bus keičiamos pasirinkto komponento savybės.

Pakeiskite komponento **label1** dviejų savybių reikšmes tokiomis:

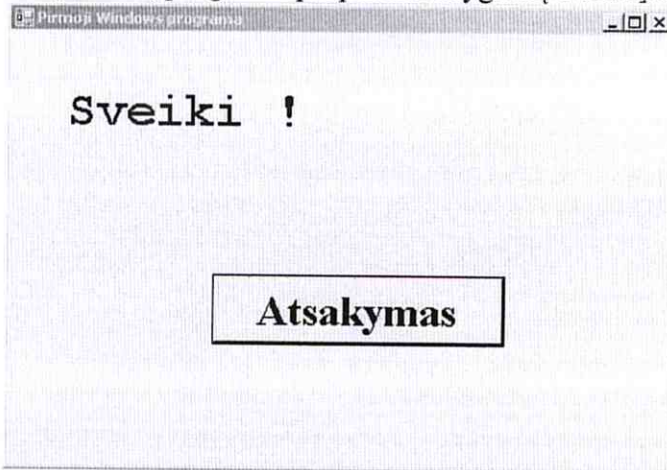
Text (Sveiki!) ir **Font** (Courier New, Bold, 28),

o komponento button1 – tokiomis:

Text (Atsakymas) ir Font (Times New Roman, Bold, 24).

Toliau išsaugokite, sukompiliuokite ir įvykdysite programą. Ekrane pasirodys programos langas (žiūr. pav.). Paspauskite mygtuką Atsakymas. Pastebėsite, kad programos lange niekas nesikeičia. Norint, kad programa atliktų kokį tai veiksmą reikia jį atitinkamai suprogramuoti.

Pavyzdžiui, programa paspaudus mygtuką Atsakymas turi taip pat pasisveikinti.



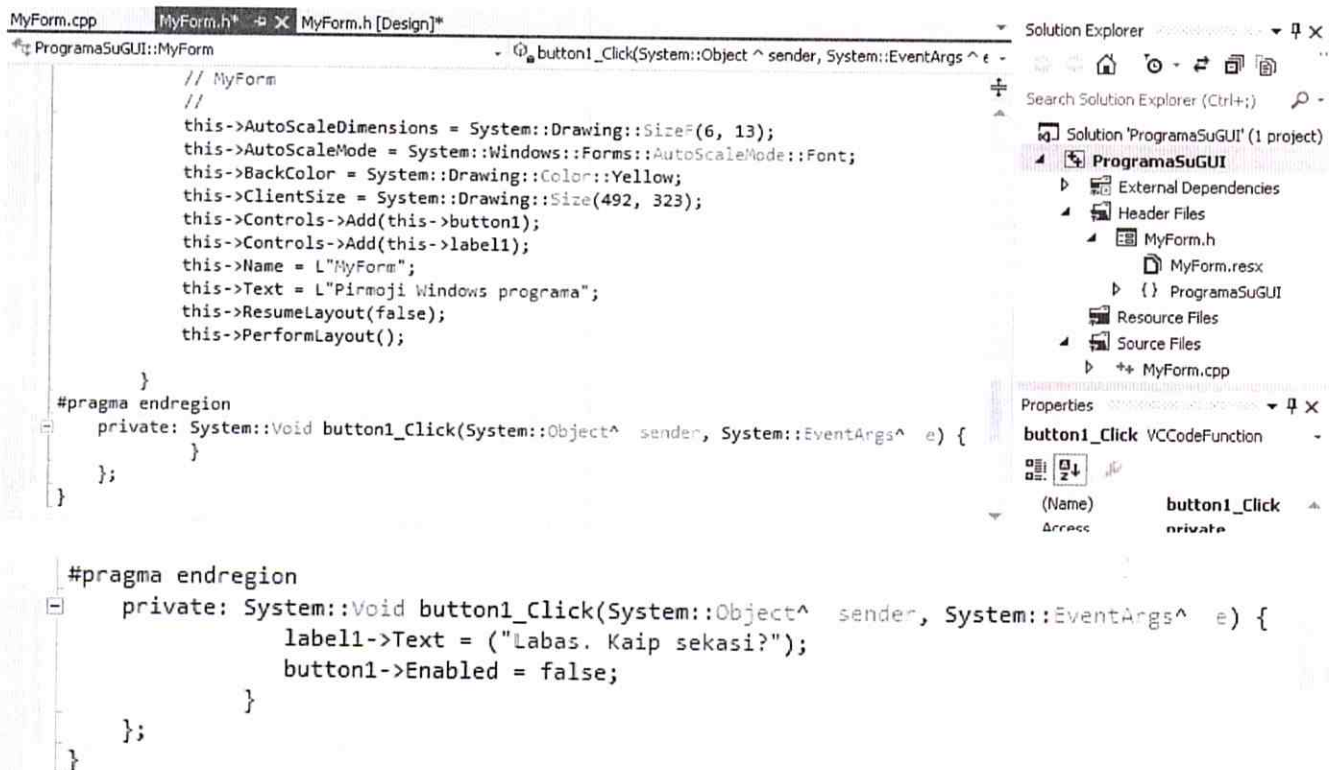
Kaip suprogramuoti mygtuko paspaudimo veiksmą?

Tam reikia du kartus greitai spragtelėti kairiuoju pelės klavišu ant mygtuko Atsakymas. Atlikus šį veiksmą šalia kortelės (lango) MyForm.h [Design] atsiveria programos teksto rašymo kortelė (langas) MyForm.h, kurioje mirksintis žymeklis rodo tą vietą, kurioje turi būti rašomas reikalingas programos tekstas (žiūr. pav.). Čia užrašomi du sakiniai:

```
label1->Text = ("Labas. Kaip sekasi?");
button1->Enabled = false;
```

Pirmasis sakiny pakeičia komponento label1 savybę Text, o antrasis sakiny pakeičia komponento button1 savybę Enabled (false - neleidžia dar kartą paspausti mygtuką) (žiūr. pav.).

Sukompiliavus ir įvykdžius programą ekrane pasirodys programos langas (žiūr. pav.). Paspaudus mygtuką Atsakymas vietoj teksto Sveiki! atsiranda tekstas Labas. Kaip sekasi? ir mygtukas Atsakymas tampa pasyvus, t. y. jis nereguoja į paspaudimus.



Paprastus skaičiavimus atliekanti programa – duomenų įvedimas ir rezultatų išvedimas

Pavyzdžiui, reikia sukurti programą, kuri sveiką skaičių pakeltų kvadratu.

Programos kūrimo veiksmų seka:

1. Sukuriamas naujas projektas, kuris pavadinamas vardu Kvadratas.
2. Į formą įkelkime 5 komponentus: `label1`, `textBox1`, `label2`, `textBox2` ir `button1` (1 pav.).
3. Pakeiskime kai kurias komponentų savybes (lentelė 1). Tai atlikus programos forma turėtų atrodyti panašiai kaip pavaizduota 2 pav.
4. Aprašykime skaičiaus, kurį kelsime kvadratu kintamąjį `x` ir rezultato kintamąjį `y` faile `Form1.h` klasės `Form1` private srityje (3 pav.).
5. Suprogramuokime mygtuko `Skaičiuoti` reakcijos į paspaudimą veiksmą: kairiuoju pelės klavišu du kartus spragtelėkime mygtuką `Skaičiuoti` ir atsivėrusiame `Form1.h` lange nurodytoje vietoje (`button1_Click()` metode) įrašykime tekstą:

```
x = int::Parse(textBox1->Text); /* 1 */  
y = x * x; /* 2 */  
label2->Text = ("Kvadratas"); /* 3 */  
textBox2->Text = y.ToString(); /* 4 */
```

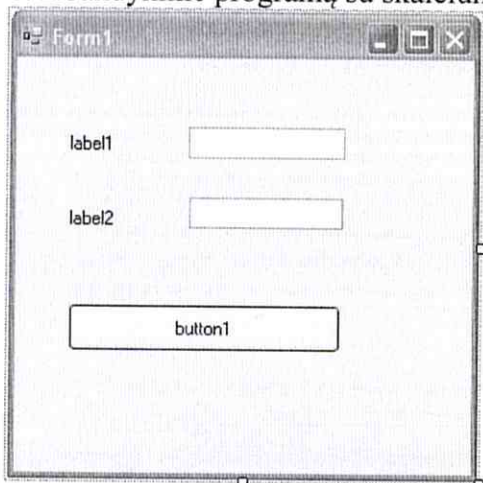
Pirmasis šio programos fragmento sakinyss kintamajam `x` priskiria skaičių užrašytą komponento `textBox1` laukelyje. Antrasis – pakelia skaičių `x` kvadratu ir priskiria kintamajam `y`. Trečiasis – pakeičia komponento `label1` savybę `Text`: Kvadratas. Ketvirtasis sakinyss yra skirtas rezultatui, kintamojo `y` reikšmei, atvaizduoti komponente `textBox2`. Prieš atliekant priskyrimo veiksmą kintamojo `y` reikšmė turi būti konvertuojama į eilutę, nes komponento `textBox2` savybės `Text` tipas yra eilutė.

6. Sukompiliuokime ir įvykdykime programą. Išbandykime ją su skaičiumi 15 (4 pav.).

7. Modifikuokime programą taip, kad ji galėtų kelti kvadratu realų skaičių. Tam reikia pakeisti faile `Form1.h` kintamųjų `x` ir `y` aprašymą iš `int` į `double`. O mygtuko `button1_Click()` metode pirmajame sakinyje vietoj `int` užrašyti `double`, t. y.

```
x = double::Parse(textBox1->Text); /* 1 */
```

8. Išbandykime programą su skaičiumi 1.5. Rezultatas turėtų gautis 2.25.



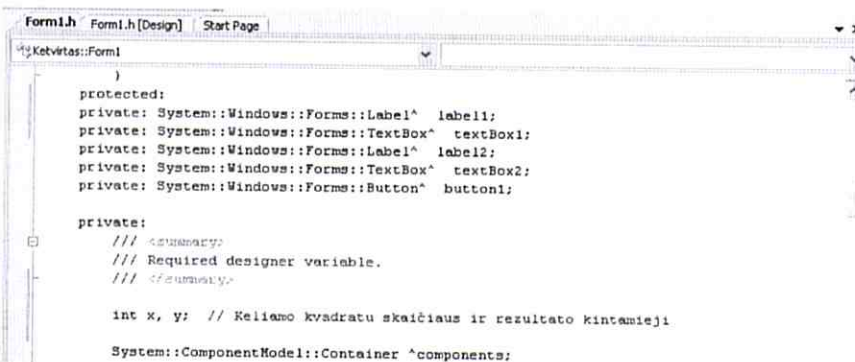
1 pav. Programos formos projektas

1 lentelė. Pakeistos komponentų savybės

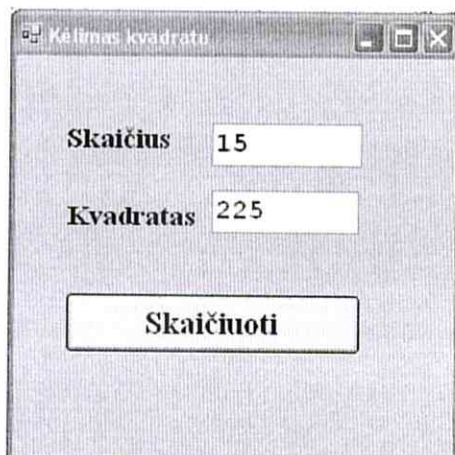
Komponentas	Savybė	Reikšmė
Form1	Text BackColor	Kėlimas kvadratu Gold
label1	Text Font	Skaičius Times New Roman, Bold, 14
textBox1	Text Font ForeColor	0 (Nulis) Courer New, Bold, 14 Blue
label2	Text Font	Rezultatas Times New Roman, Bold, 14
textBox2	Font ForeColor	Courer New, Bold, 14 Red
button1	Text Font	Skaičiuoti Times New Roman, Bold, 16



2 pav. Pakeistas programos formos projektas



3 pav. Failo Form1.h fragmentas su kintamųjų x ir y aprašymu



4 pav. Programa atlikusi skaičiavimus su skaičiumi 15