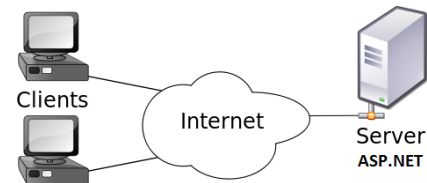


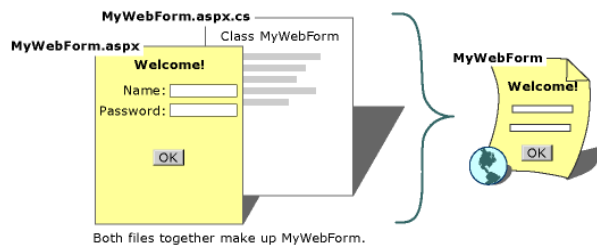
Ižanga į ASP.NET

ASP.NET yra šiuolaikinė profesionali interneto puslapių kūrimo technologija sujungianti Microsoft ASP interneto puslapių ir C# programavimo kalbas. Remiantis dabartinėmis interneto tendencijomis daugelis programinės įrangos sprendimų keliai iš stacionarių kompiuterių (desktop) į internetinį lygmenį, tad labai aktualu mokėti dirbti su šiomis technologijomis.

Bendru atveju ASP.NET yra serveryje veikianti technologija, kurios suformuotus HTML puslapius interneto tinklais gali pasiekti naršyklę naudojantys vartotojai.



ASP.NET praplečia HTML puslapių standartą savo `<asp>` žymėmis ir papildomomis galimybėmis. Šias žymes analogiškai kaip ir kitose (pvz. PHP) kalbose interpretuoja procesorius ir vykdo jose parašytą kodą. Su šia technologija pereita prie visiškai objektinės C# kalbos, kuri populiori ir lanksti, tačiau taip pat atgaliniu būdu suderinama su ASP JavaScript atmaina.



Su ASP.NET pristatytas esminis technologinis principas – „Code Behind“. Pagal jį visi internetiniai puslapiai traktuojami kaip vartotojui priimtinos duomenų formos, kurias galima išskaidyti į du sluoksnius – HTML žymių/dizaino (.aspx failą) bei C# kodą „behind“ (.cs failą). Skirtumas dar toks, jog svetainės kodas gali būti sukompiliuotas į bibliotekas

– Windows DLL (*angl.* dynamic link library) ir vykdomas tiesiogiai CPU komandų lygiu (*native* aplinkoje), kas turėtų suteikti daugiau našumo.

ASP.NET yra karkasas, turintis standartinius formų komponentus – mygtukai, įvedimo laukai ir t.t., automatinį validavimą, kurie palengvina ir pagreitina kūrimo procesą. Tokie dalykai kaip prisijungimo (*angl.* login) formos ir kita – jau realizuota, telieka įdėti komponentą į Web puslapį. ASP taip pat palaiko šablonus (*angl.* templates), kurie padeda realizuoti tėvo/vaiko (master/child) puslapių struktūrą bei leidžia įtraukti vieną šabloną į kitą ir pan.

Norint kurti ASP.NET puslapius net nebūtina gerai žinoti HTML (tačiau rekomenduotina), kadangi daugelis HTML žymių tiesiog pakeičiamos standartine ASP žyme, kuri atitinka komponentą (pvz.: `<asp:TextBox>` žymė pakeičia HTML `<input type="text">` ir pan.). Serveris, apdorojęs puslapį, automatiškai pakeičia ASP žymes į atitinkamas HTML, kurias jau gali suprasti interneto naršyklė.

Puslapį galima įsivaizduoti kaip vieną didelę HTML formą, kadangi dažniausiai visi komponentai yra sudedami tarp HTML puslapio `<form>` žymių. Taip siekiama nuo visiškai būsenos neišsaugojančio (*angl.* stateless) HTML pereiti prie būseną išsaugojančio, nuoseklaus „su atmintimi“ puslapio koncepcijos.

Sukurtus sprendimus-puslapius dažniausiai naudoja verslo sektorius (*angl.* enterprise), todėl ASP .NET talpinimo (*angl.* hosting) paslaugų įmonių Lietuvoje nėra tiek daug, o pačios talpinimo paslaugos yra pakankamai brangios vien dėl to, jog kainuoja Windows licenzijos. Užsienyje nemokamų ir gerų talpyklų taip pat sunku surasti. Susikurti savo serverį galima, jeigu prieinama Windows Server OS, kuri

kartu jau integruoja ir IIS (Internet Information Services), reikalingą ASP.NET veikimui. Instaliavus belieka tik pasirinkti ir nustatyti, jog norite sukurti Web serverį.

Dirbančių su ASP .NET Lietuvoje yra pakankamai nemažai ir dažniausiai, ASP .NET labiau naudojamas didesnio verslo ar enterprise lygio svetainėms kurti, tačiau niekas neriboja taikyti šią technologiją mežesniems iššūkiams.

1. Priedas

ASP.NET Web puslapių kūrimas Microsoft Visual C# 2012 programavimo aplinkoje

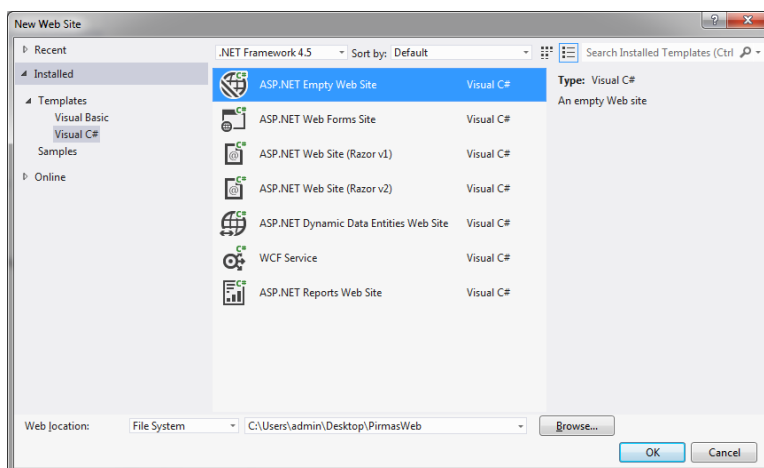
Susipažinsite su langų kūrimu Microsoft Visual C# programavimo aplinkoje:

- ASP.NET Web puslapio projekto sukūrimu;
- Komponentais ir jų savybių keitimu;
- Duomenų įvedimu klaviatūra ir rezultatų į ekraną išvedimu.

1.1. Projektas ir komponentai

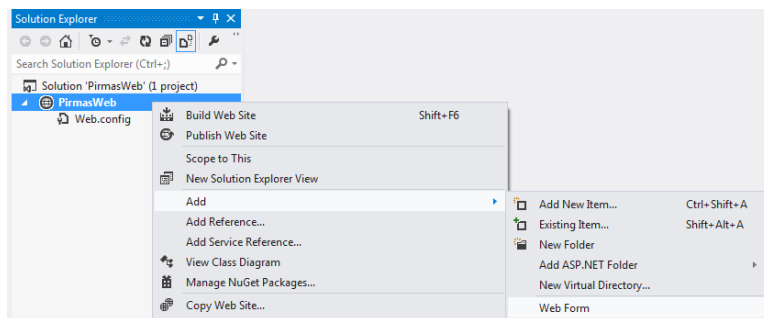
U Pirmas žingsnis.

- Meniu FILE → New → Web Site pasirinkite tuščio (empty) ASP.NET Web puslapio projektą bei jo saugojimo vietos vardą (pvz.: **PirmasWeb**).



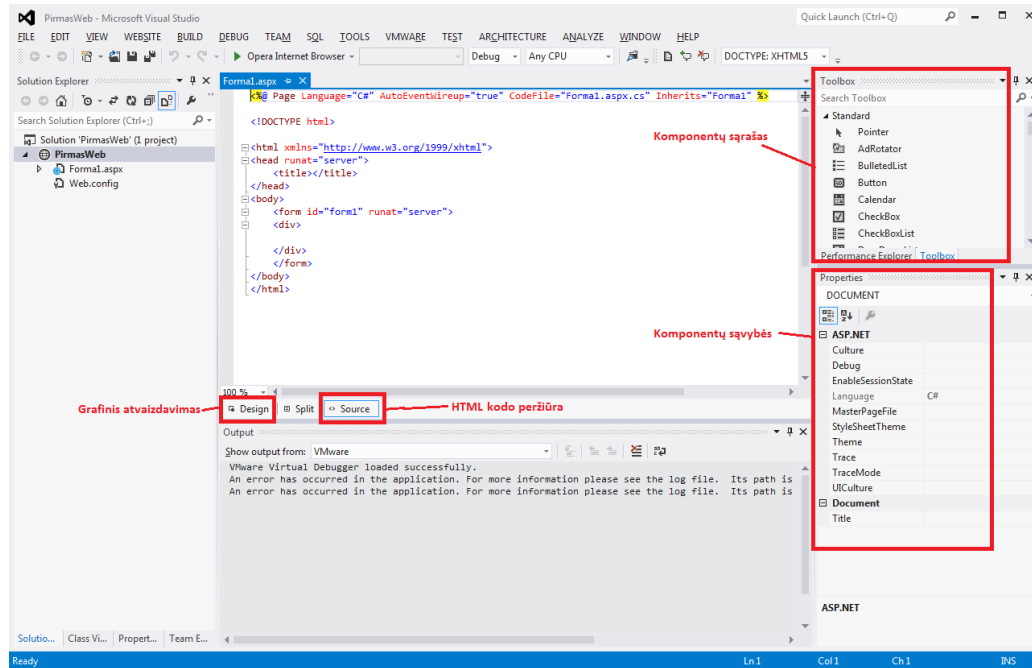
U Antras žingsnis.

- Paspaudę dešinę pelės mygtuką ant projekto pasirenkame Add → Web Form bei įvedame formos pavadinimą (pvz.: **Forma1**).



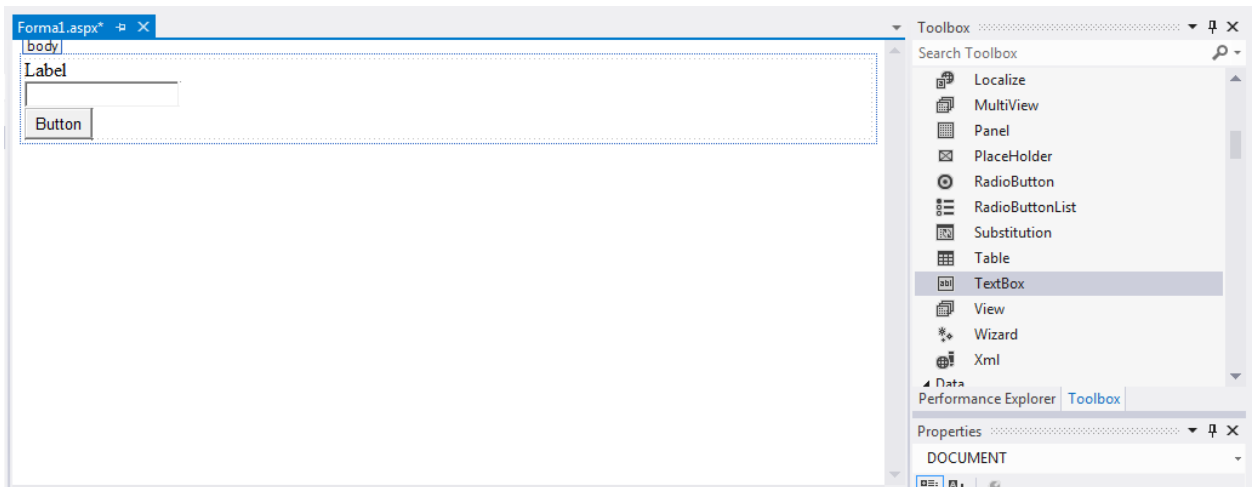
Trečias žingsnis.

- Atsidariusiame lange matome puslapio HTML žymes, įrankių-komponentų sąrašą bei komponentų savybių sąrašą. Paspauskite Design mygtuką ir peržiūrėkite preliminarų puslapio vaizdą.



Ketvirtas žingsnis.

- Matomas puslapis kol kas tuščias. Komponentų sąraše suraskite Label komponentą ir nutempkite jį pele į puslapį. Tą patį padarykite su TextBox bei su Button komponentais. Sudėkite juos skirtingose eilutėse kaip pavaizduota toliau.



Pastaba: Atkreipkite dėmesį, jog HTML kodo lange (Source) puslapio formos HTML žymės buvo papildytos dviem naujais įrašais. Visada naujus komponentus dėkite tik į formos vidų, t.y. tarp `<form>` ir `</form>` žymių.



```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Form1.aspx.cs" Inherits="Form1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body>
<form id="form1" runat="server">
<div>

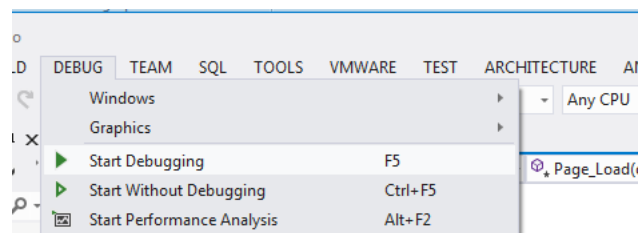
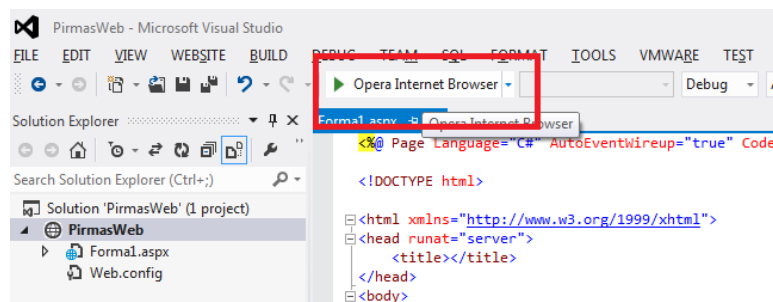
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
<br />
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<br />
<asp:Button ID="Button1" runat="server" Text="Button" />

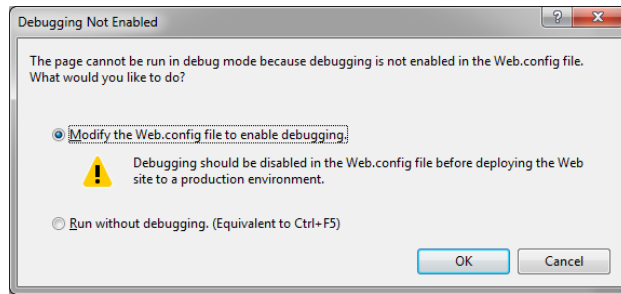
</div>
</form>
</body>
</html>

```

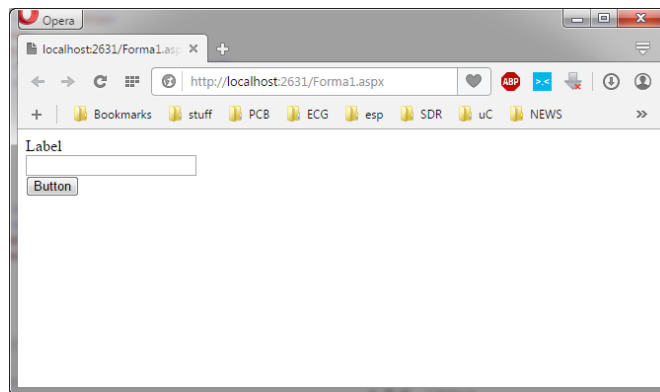
Penktas žingsnis.

- Iš meniu juostos pasirinkite puslapio paleidimo Debug mygtuką arba meniu `DEBUG` → `Start Debugging`. Stebėkite, kaip bus paleidžiama Interneto naršyklė ir pateikiamas kuriamo puslapio vaizdas.



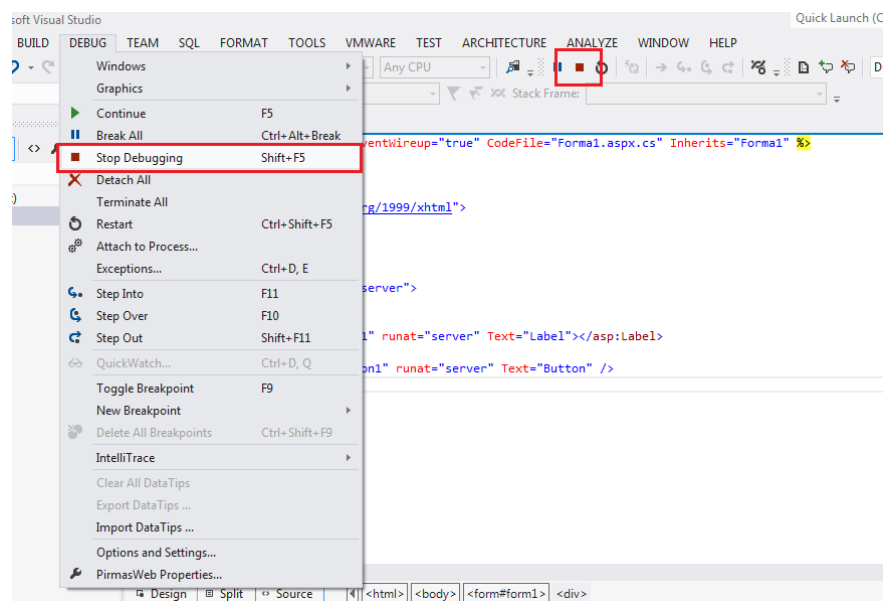


Pastaba: Jeigu pasirodo nurodytas derinimo pranešimas, tai pasirenkame pirmą variantą ir spaudžiame OK.



Šeštas žingsnis.

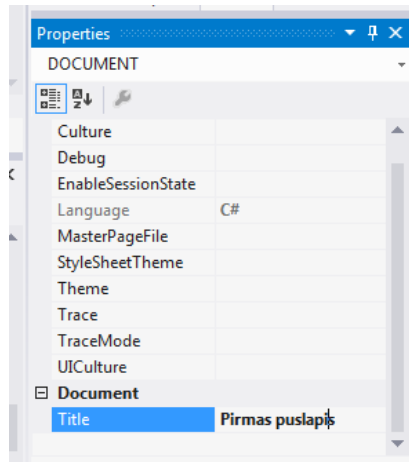
- Kol kas mygtukas naršyklėje nereaguoja į paspaudimus. Uždarykite ją, grįžkite į Visual Studio ir užbaikite derinimo režimą paspaudami ant meniu esančio raudono stačiakampio stabdymo mygtuko arba per **DEBUG** → **Stop Debugging**.



1.2. Komponentų ir puslapio savybių keitimas

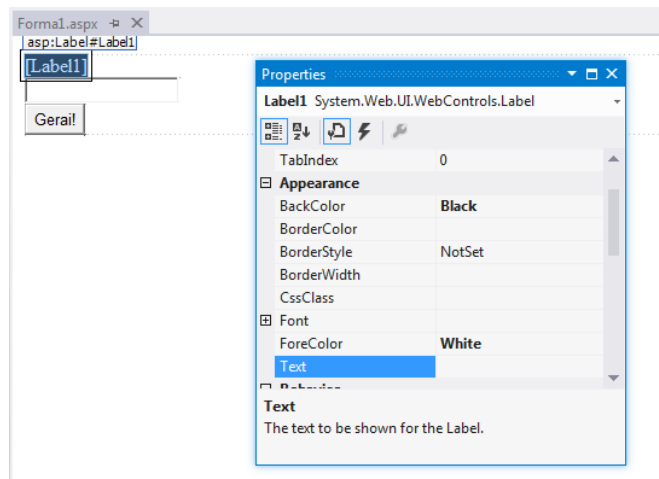
➤ Pirmas žingsnis.

- Įeikite į puslapio Design peržiūrą ir pele paspauskite ant balto puslapio fono. Puslapio savybėse Properties pakeiskite puslapio antraštės pavadinimą Title (pvz.: **Pirmas puslapis**).



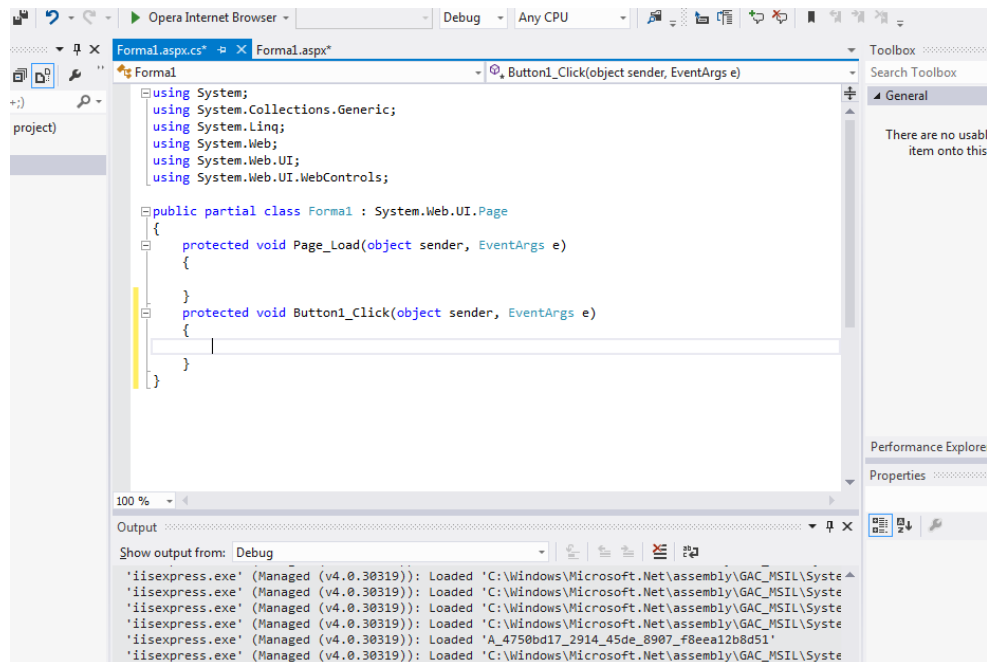
➤ Antras žingsnis.

- Pele puslapyje pažymėkite Label komponentą, nustatykite tekstui baltą spalvą (ForeColor), o fonui – juodą (BackColor), išvalykite parašytą tekstą (Text). Mygtukui nustatykite patvirtinimo tekstą (pvz.: **Gerai!**).



➤ Trečias žingsnis.

- Design peržiūroje du kartus pele spragtelėkite ant mygtuko. Atsidariusiame lange matome (Code-Behind) automatiškai sugeneruotą puslapio C# kodą.



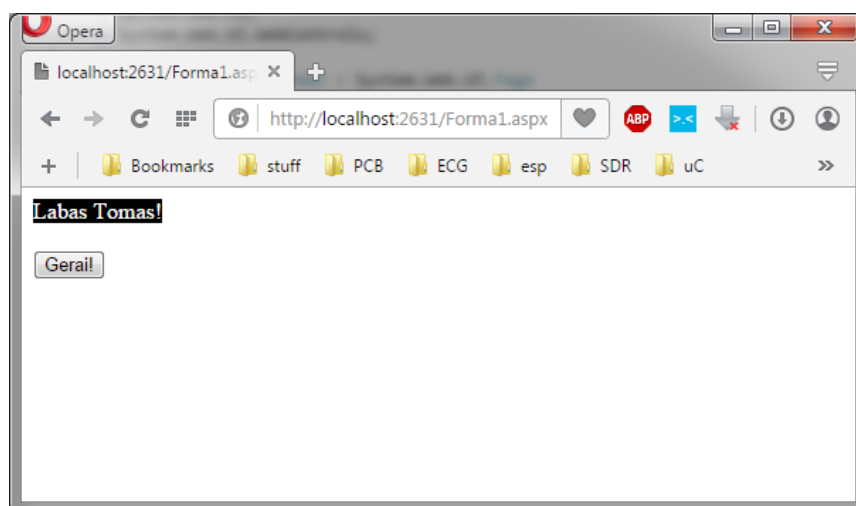
• Ketvirtas žingsnis.

- Button1_Click metodą kviečiamas Web puslapyje paspaudus jame esantį mygtuką, todėl pakeičiame šio metodo realizacijos kodą į pateiktą žemiau:

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = "Labas " + TextBox1.Text + "!";
    TextBox1.Visible = false;
}
```

• Penktas žingsnis.

- Išsaugome ir paleidžiame Web puslapį išbandymui Debug mygtuku ar per DEBUG → Start Debugging. Naršyklėje į teksto lauką įrašykite savo vardą ir paspauskite mygtuką.



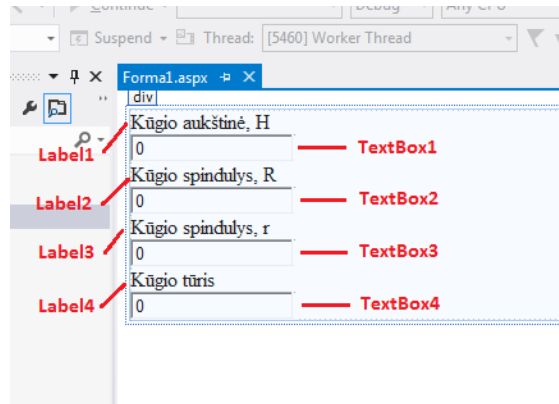
1.3. Reikšmių įvedimas, mygtuko veiksmo programavimas

❶ Pirmas žingsnis.

- Sukurkite naują Web puslapio projektą per FILE → New → Web Site. Pavadinkite nauju vardu (pvz.: **AntrasWeb**). Projektui pridėkite naują formą su Add → Web Form bei įveskite jos pavadinimą (pvz.: **Forma1**).

❷ Antras žingsnis.

- Puslapio Design režime išdėliokite komponentus kaip pavaizduota žemiau:




❸ Trečias žingsnis.

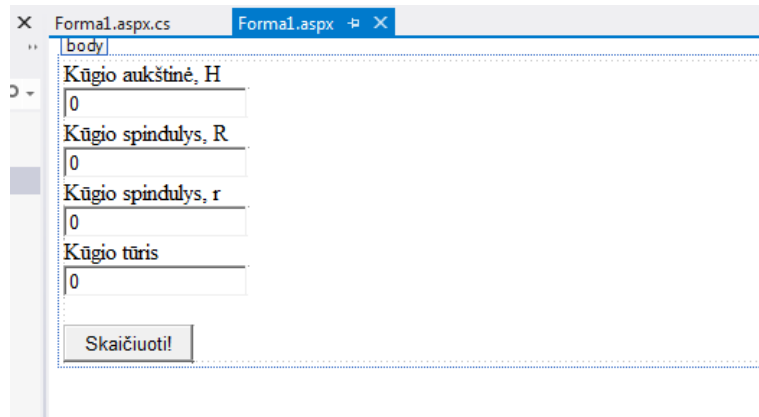
- Atsidarykite HTML žymių Source režimą ir po paskutinio tekstinio lauko TextBox4 įveskite žymės mygtuko aprašymui:

```
<br />  
<br />  
<asp:Button ID="Button1" runat="server" Text="Skaičiuoti!" />
```

Pastaba: Formos vaizdo patikrinimui galite pereiti į Design peržiūros režimą. Laukiamas rezultatas pateikiamas toliau.



```
<asp:Label ID="Label3" runat="server" Text="Kūgio spindulys, r"></asp:Label>  
<br />  
<asp:TextBox ID="TextBox3" runat="server">0</asp:TextBox>  
<br />  
<asp:Label ID="Label4" runat="server" Text="Kūgio tūris"></asp:Label>  
<br />  
<asp:TextBox ID="TextBox4" runat="server">0</asp:TextBox>  
<br />  
<br />  
<asp:Button ID="Button1" runat="server" Text="Skaičiuoti!" />
```



🔹 Ketvirtas žingsnis.

- Du kartus paspauskite ant mygtuko ir pereikite į kodo rašymo langą. Užpildykite mygtuko paspaudimo metodą kūgio tūrio skaičiavimo tekstu:

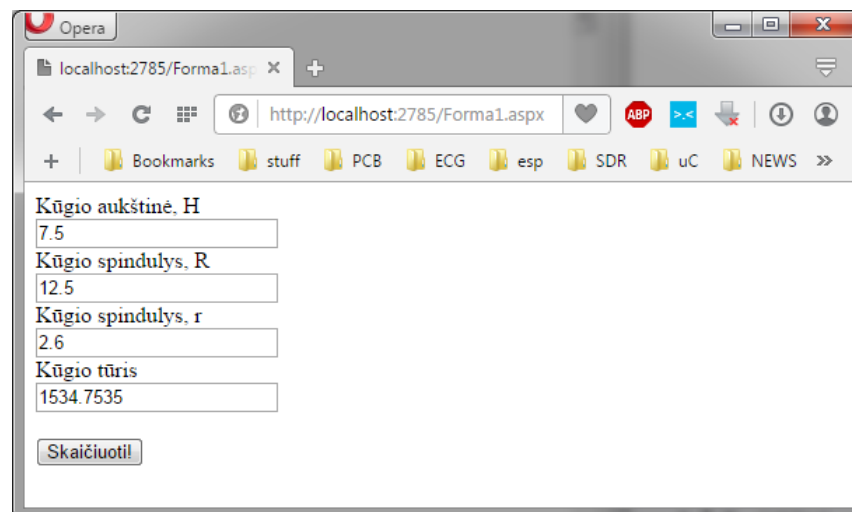
```
protected void Button1_Click(object sender, EventArgs e)
{
    double H = double.Parse(TextBox1.Text, CultureInfo.InvariantCulture);
    double R = double.Parse(TextBox2.Text, CultureInfo.InvariantCulture);
    double r = double.Parse(TextBox3.Text, CultureInfo.InvariantCulture);
    double V = (1.0 / 3) * 3.14 * H * (R * R + R * r + r * r);
    TextBox4.Text = V.ToString();
}
```

Pastaba: Bibliotekų srityje taip pat įterpkite žemiau pateiktą eilutę:

```
using System.Globalization;
```

🔹 Penktas žingsnis.

- Išbandykite kodo veikimą paleidę puslapį naršyklėje per meniu: DEBUG → Start Debugging. Pasitikrinkite gautą rezultatą.



2. Sesija ir reikšmių validavimas

ASP.NET Web puslapių kūrimas Microsoft Visual C# 2012 programavimo aplinkoje

Susipažinsite su sudėtingesniais darbo scenarijais kuriant interneto puslapius Microsoft Visual C# programavimo aplinkoje:

- Sesijos panaudojimu naršymo būsenai saugoti;
- Duomenų įvedimu iš serveryje esančių failų;
- Komponentų reikšmių automatiniu validavimu;

2.1. Sesijos naudojimas būsenai saugoti

Serverio ir kliento komunikavimas vyksta užklausos-atsakymo (angl. *request-response*) principu. Toks komunikavimo būdas yra geras tol, kol neprireikia atlikti sudėtingesnių veiksmų (pvz.: sekti pirkinių krepšelio užpildymą, apmokėjimą ar net prisijungimą), nes nė viena pusė neišsipareigoja saugoti jokios darbinės-bendravimo informacijos. Šiai problemai spręsti HTML yra sukurtas *Sesijos* mechanizmas, kuris įgalina laikyti papildomą, su klientu susijusią, informaciją serverio atmintyje. Tam, kad serveris galėtų atskirti, kurie duomenys yra konkretaus kliento, kliento pusėje – naršyklėje, yra naudojami slapukai (angl. *cookies*), kuriuose įrašomas šifruotas kliento ir serverio duomenis susiejantis ID. Kiekvienos užklausos metu šis ID yra siunčiamas/grąžinamas į serverį, tokiu būdu serveris gali atpažinti tą patį klientą bei į ASP. NET puslapio programą paduoti tik jam skirtą informaciją. Sesijos valdymui ASP. NET turi globalų kintamąjį *Session*, kuriame galima talpinti kitų kintamų reikšmes puslapio veikimui užtikrinti. *Session* kintamasis yra maišos lentelės (angl. *hash table*) tipo, tad jame galima padėti ir surasti duomenis pagal *string* raktą (angl. *key*), pavyzdžiui: `Session["raktas"] = kintamojoReikšmė`, o pasiimti kreipiantis tuo pačiu raktu: `kintamasis = (int)Session["raktas"]`. Rakto pavadinimas turi būti netuščias tekstas.

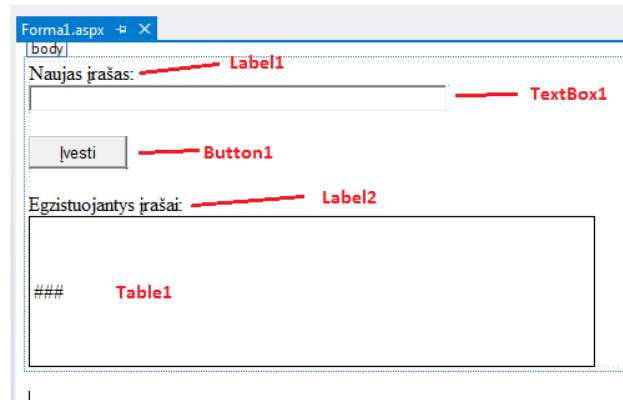
Pastaba. Kadangi *Session* kintamasis neapriboja laikomų duomenų tipo, tai pasiimant reikšmes iš jo, grąžinamus duomenis būtina vėl paversti į konkretų reikiamą tipą naudojant tipo-keitimo (angl. *type-cast*) operatorių (t.y. `tipas x = (tipas)Session["raktas"]`).

🔵Pirmas žingsnis.

- Sukurkite naują Web puslapio projektą per FILE → New → Web Site. Pavadinkite nauju vardu (pvz.: **SesijaWeb**). Projektui pridėkite naują formą su Add → Web Form bei įveskite jos pavadinimą (pvz.: **Form1**).

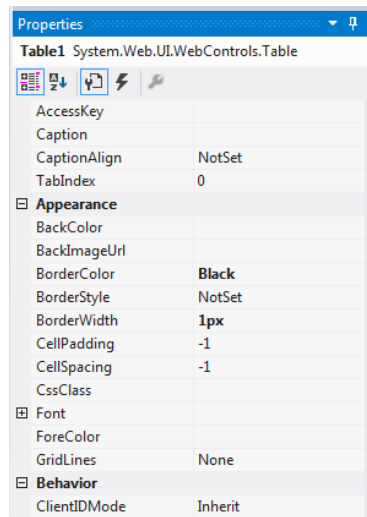
🔵Antras žingsnis.

- Puslapio Design peržiūroje įdėkite teksto elementus (Label), tekstinį lauką (TextBox), mygtuką (Button) bei lentelę (Table) kaip parodyta toliau:



U Trečias žingsnis.

- Lentelei nustatykite juodą (BorderColor) ir 1px storio (BorderWidth) rėmelį.



U Ketvirtas žingsnis.

- Dukart spragtelėkite ant mygtuko Design režime. Atsidarius kodo langui į Button1_Click metodą vedame lentelės papildymo eilutę tekstą.

```
protected void Button1_Click(object sender, EventArgs e)
{
    issaugotasTekstas = TextBox1.Text;
    IterptiIrasa(issaugotasTekstas);
}

private void IterptiIrasa(string tekstas)
{
    TableCell cell = new TableCell();
    cell.Text = tekstas;

    TableRow row = new TableRow();
    row.Cells.Add(cell);

    Table1.Rows.Add(row);
}
```

Penktas žingsnis.

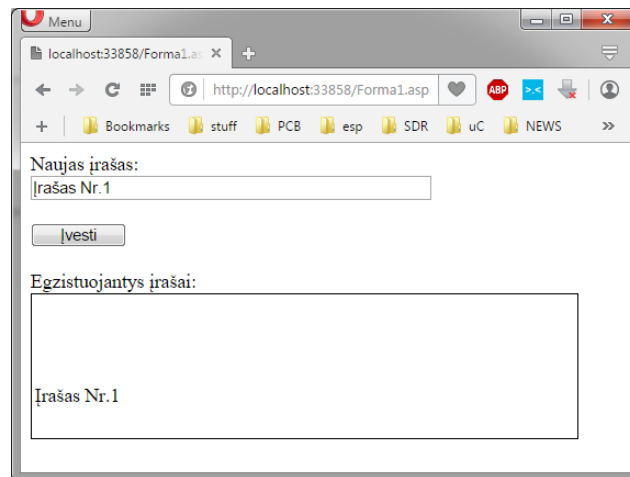
- Virš `Page_Load` metodo aprašykite naują `string` kintamąjį pavadinimu `issaugotasTekstas` - jame bus išsaugoma tekstiniam laukelyje įvesta reikšmė. Kiekvieną kartą naršyklėje užsikrovus puslapiui esamą reikšmę įrašysime į lentelę.

```
private String issaugotasTekstas;

protected void Page_Load(object sender, EventArgs e)
{
    IterptiIrasa(issaugotasTekstas);
}
```

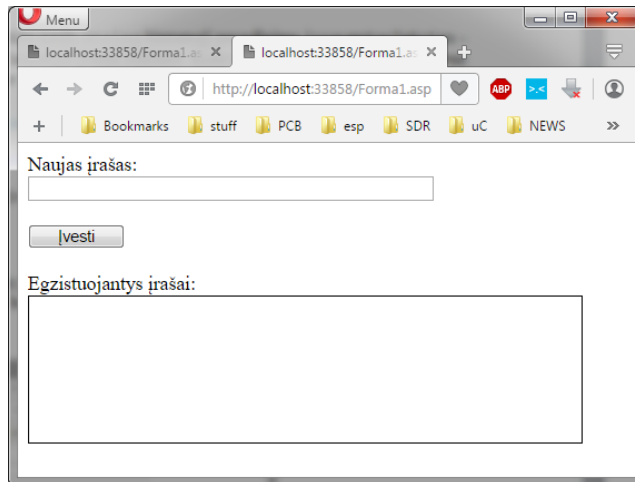
Šeštas žingsnis.

- Išbandykite kodo veikimą paleidę puslapį naršyklėje per meniu: `DEBUG` → `Start Debugging`. Įrašykite tekstą į teksto lauką bei paspauskite įvedimo mygtuką. Pasitikrinkite, ar lentelėje atsirado įrašas.



Septintas žingsnis.

- Nusikopijuokite naršyklėje adreso juostoje matomą adresą, toliau – atsidarykite kitą skirtuką (Tab) ir nukeliate tuo pačiu nusikopijuotu adresu (t.y. atidarykite puslapį dar kartą). Pastebėkite, jog įrašyta reikšmė neišliko puslapį atidarius pakartotinai. To priežastis – kaskart atidarant puslapį iš naujo serveryje sukuriamas naujas puslapio objektas, o anksčiau įvesta informacija į kintamąjį `issaugotasTekstas` neišlieka.



⓪ Aštuntas žingsnis.

- Grįžkite į Visual Studio aplinką bei sustabdykite serverį per meniu: **DEBUG** → **Stop Debugging**. Atsidarykite C# kodo langą ir pakoreguokite kodą taip, jog dominanti reikšmė būtų išsaugoma *Session* saugykloje, o puslapiui užsikrovus – atkuriamą iš jos (pagal raktą „tekstas“), kaip parodyta toliau:

```
protected void Page_Load(object sender, EventArgs e)
{
    // paimame išsaugota reikšmę iš sesijos kintamojo
    issaugotasTekstas = (string)Session["tekstas"];

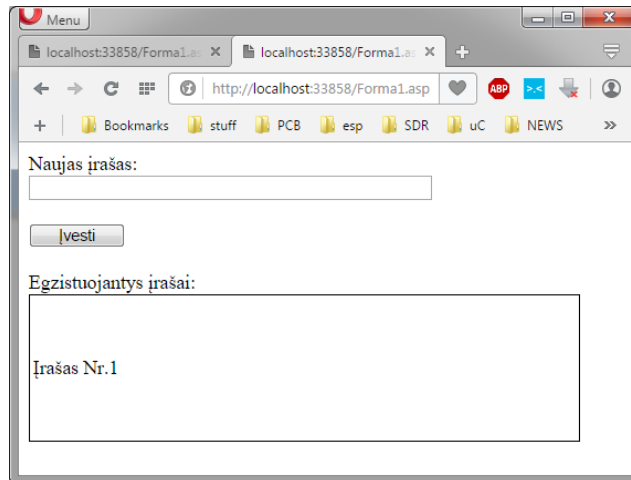
    IterptiIrasa(issaugotasTekstas);
}

protected void Button1_Click(object sender, EventArgs e)
{
    issaugotasTekstas = TextBox1.Text;
    IterptiIrasa(issaugotasTekstas);

    // išsaugome reikšmę sesijos kintamajame
    Session["tekstas"] = issaugotasTekstas;
}
```

⓪ Devintas žingsnis.

- Išbandykite kodo veikimą paleidę puslapį naršyklėje per meniu: **DEBUG** → **Start Debugging**. Įrašykite tekstą į teksto lauką bei paspauskite įvedimo mygtuką. Pasitikrinkite, ar lentelėje atsirado įrašas. Atidarykite tą patį puslapį kitame skirtuke (**Tab**). Pastebėkite – priešingai, negu anksčiau, įvesta reikšmė buvo išsaugota ir yra iš karto matoma vos tik puslapis parodomas.



Pastaba. Session saugykla skirta laikyti mažus duomenų kiekius, kurie pažymėti `Serializable` atributu (t.y. visi baziniai C# tipai ir išvestiniai objektai). Didelės apimties duomenims saugoti naudojamos SQL ir No-SQL duomenų bazės (angl. *databases*).

2.2. Duomenų įvedimas iš XML ir kitų failų

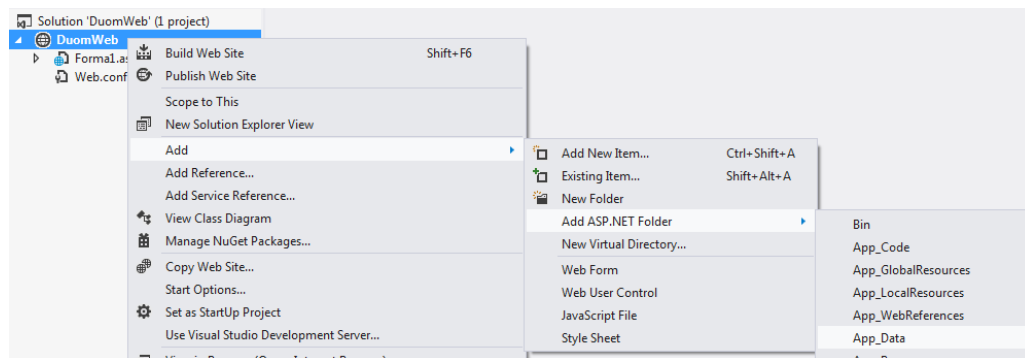
ASP .NET serveryje (pagal nutylėjimą) puslapio duomenų failus įprasta ir rekomenduojama laikyti App_Data sub-kataloge. Norint prieiti prie kataloge esančių duomenų failų naudojant santykinį dokumento kelią, reikia naudoti `Server.MapPath` funkciją, kuri grąžina absoliutų kelią sistemoje iki reikiamo failo.

❶ Pirmas žingsnis.

- Sukurkite naują Web puslapio projektą per `FILE` → `New` → `Web Site`. Pavadinkite nauju vardu (pvz.: **DuomWeb**). Projektui pridėkite naują formą su `Add` → `Web Form` bei įveskite jos pavadinimą (pvz.: **Forma1**).

❷ Antras žingsnis.

- Į projektą įdėkite standartinį ASP.NET duomenų katalogą `App_Data`.



❸ Trečias žingsnis.

- Kataloge `App_Data` sukurkite **Prekes.txt** failą su tekstu, kaip pateikta toliau:

```
Pieštukas 0.1
Trintukas 0.1
A4-popierius 2.31
Rašikliai 0.45
Tušas 1.57
Spalvotas-popierius 3.22
Guašas 2.90
```

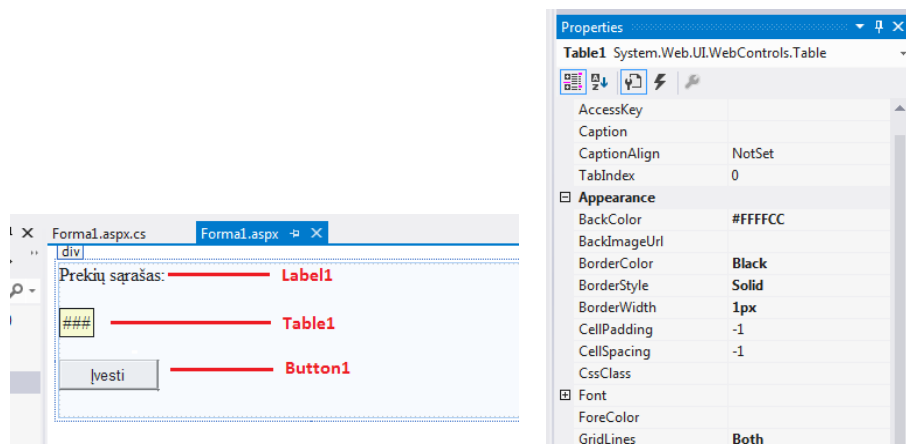
❹ Ketvirtas žingsnis.

- Kataloge `App_Data` sukurkite XML failą **Prekes.xml** su turiniu, kaip pateikta toliau:

```
<parduotuve>
  <preke pavadinimas="Pieštukas" kaina="0.1"/>
  <preke pavadinimas="Trintukas" kaina="0.1"/>
  <preke pavadinimas="A4-popierius" kaina="2.31"/>
  <preke pavadinimas="Rašikliai" kaina="0.45"/>
  <preke pavadinimas="Tušas" kaina="1.57"/>
  <preke pavadinimas="Spalvotas-popierius" kaina="3.22"/>
  <preke pavadinimas="Guašas" kaina="2.90"/>
</parduotuve>
```


Penktas žingsnis.

- Sudėliokite formos komponentus - Label, Table bei Button kaip pavaizduota toliau. Table komponentui nustatykite savybes: BackColor - #FFFFCC, BorderColor - Black, BorderStyle - Solid, BorderWidth - 1px, GridLines - Both, o mygtukui tekstą - „Ivesti“:



Šeštasis žingsnis.

- Kodo lange parašykite mygtuko metodo Button1_Click kodą, kuris perskaito tekstinį duomenų failą ir parodo puslapyje dokumento turinį lentelės pavidalu kaip pavaizduota toliau:

```
protected void Page_Load(object sender, EventArgs e)
{
    TableRow row = new TableRow();

    TableCell pavadinimas = new TableCell();
    pavadinimas.Text = "<b>Pavadinimas</b>";
    row.Cells.Add(pavadinimas);

    TableCell kaina = new TableCell();
    kaina.Text = "<b>Kaina, Eur</b>";
    row.Cells.Add(kaina);

    Table1.Rows.Add(row);
}

protected void Button1_Click(object sender, EventArgs e)
{
    string[] visosEilutes =
File.ReadAllLines(Server.MapPath("App_Data/Prekes.txt"));

    foreach (string eilute in visosEilutes)
    {
        string[] dalys = eilute.Split(' ');

        TableRow row = new TableRow();

        TableCell pavadinimas = new TableCell();
        pavadinimas.Text = dalys[0];
```

```

        TableCell kaina = new TableCell();
        kaina.Text = dalys[1];

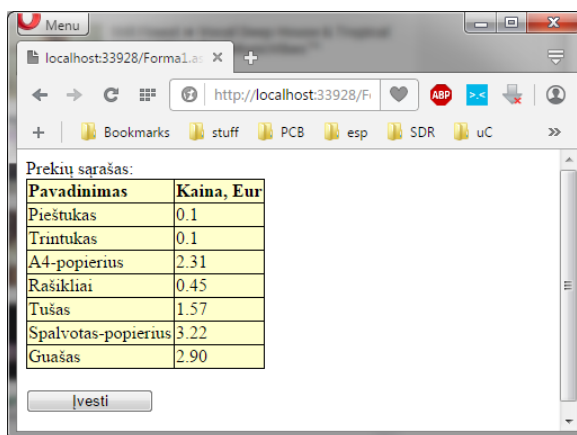
        row.Cells.Add(pavadinimas);
        row.Cells.Add(kaina);

        Table1.Rows.Add(row);
    }
}

```

Septintas žingsnis.

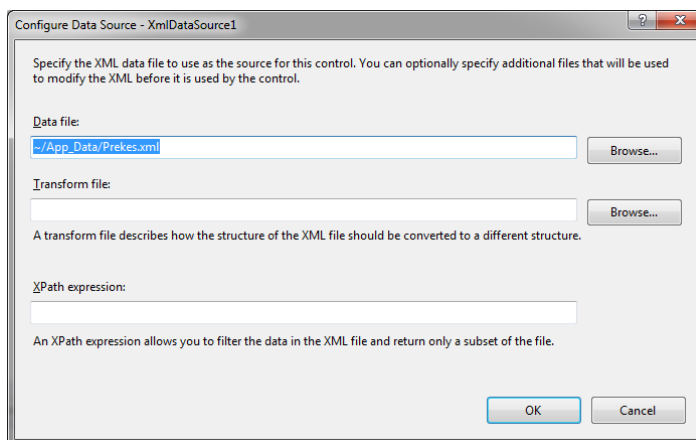
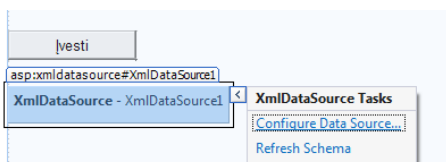
- Išbandykite kodo veikimą paleidę puslapį naršyklėje per meniu: DEBUG → Start Debugging. Paspauskite mygtuką – palyginkite gautą rezultatą su pavyzdžiu toliau.



Aštuntas žingsnis.

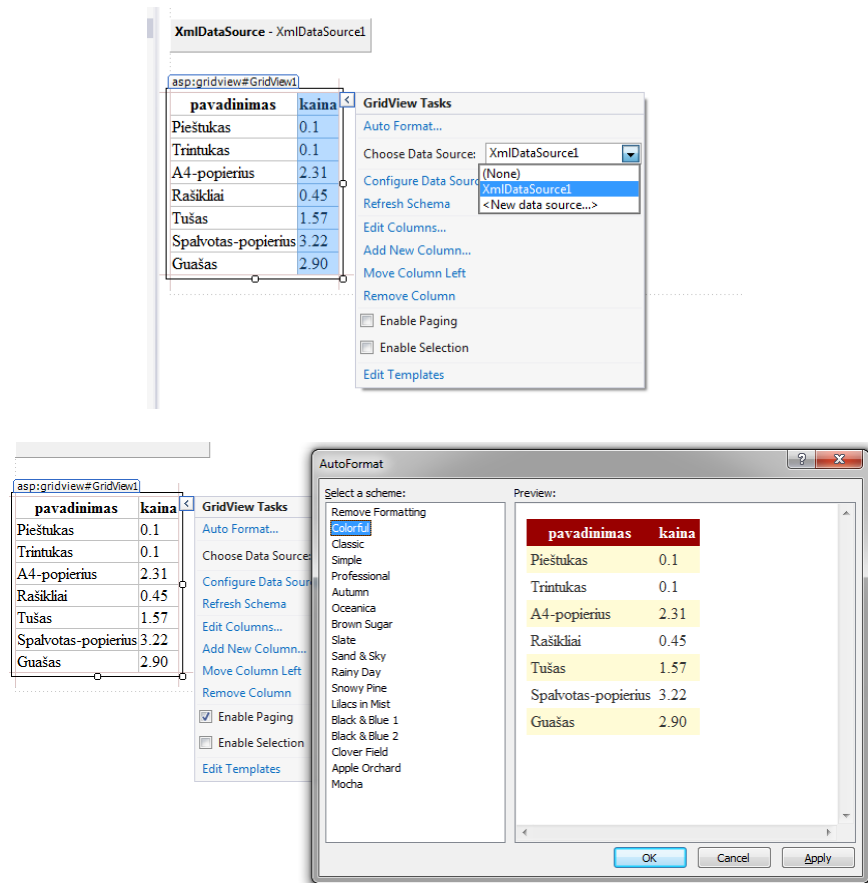
- Duomenys taip pat gali būti įkeliami neparašant nė vienos kodo eilutės. Tam gali būti naudojami įvairūs duomenų šaltinių (angl. *Source*) ir jų peržiūros komponentai. Tokie komponentai dažnai jau turi realizuotas naudingas funkcijas, kaip puslapiavimas, redagavimas ir t.t.

Puslapio Design peržiūros lange po mygtuku įterpkite XmlDataSource komponentą. Pasirinkite Configure Data Source ir atsidariusiame lange Data File lauke su Browse pasirinkite anksčiau sukurtą **Prekes.xml** failą.



Devintas žingsnis.

- Žemiau XmlDataSource komponento įkelkite GridView komponentą. Pasirinkę rodyklę ant šio komponento iš sąrašo nustatykite XmlDataSource1 kaip duomenų šaltinį. Taip pat pakeiskite lentelės sąrašo stilių į **Colorful** pasirinkę „Auto Format...“.



Dešimtas žingsnis.

- Išbandykite kodo veikimą paleidę puslapį naršyklėje per meniu: DEBUG → Start Debugging. Paspauskite mygtuką – palyginkite gautą rezultatą su pavyzdžiu toliau.

Menu

localhost:33928/Forma1.asp

Prekių sąrašas:

Pavadinimas	Kaina, Eur
Pieštukas	0.1
Trintukas	0.1
A4-popierius	2.31
Rašikliai	0.45
Tušas	1.57
Spalvotas-popierius	3.22
Guašas	2.90

pavadinimas	kaina
Pieštukas	0.1
Trintukas	0.1
A4-popierius	2.31
Rašikliai	0.45
Tušas	1.57
Spalvotas-popierius	3.22
Guašas	2.90

2.3. Komponentų reikšmių validavimas

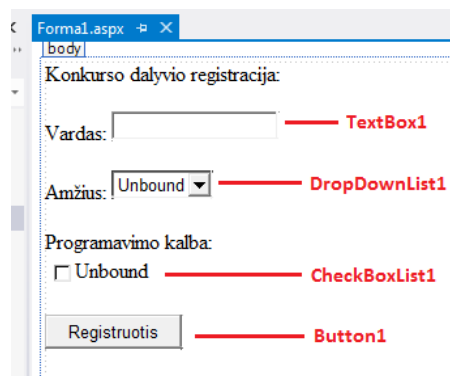
Su grafine aplinka dirbantis žmogus ne visada iš pirmo karto sugeba suvesti reikalingus duomenis teisingai – dažnai pasitaiko klaidų. ASP.NET turi automatinio tikrinimo-validavimo priemones, leidžiančias puslapio programuotojui nustatyti taisykles, pagal kurias į komponentus įvestos reikšmės bus pirmiausia tikrinamos kliento pusėje prieš siunčiant jas į serverį. Serverio pusėje validavimas visada dar kartą turi būti atliekamas siekiant apsisaugoti nuo situacijų, kai kliento pusėje validavimas nesuveikia ar būna tyčia piktybiškai išjungtas. Šios priemonės yra Visual Studio IDE `Toolbox` lange, skiltyje `Validation`.

🕒Pirmas žingsnis.

- Sukurkite naują Web puslapio projektą per `FILE` → `New` → `Web Site`. Pavadinkite nauju vardu (pvz.: **ValidWeb**). Projektui pridėkite naują formą su `Add` → `Web Form` bei įveskite jos pavadinimą (pvz.: **Forma1**).

🕒Antras žingsnis.

- Į Web puslapio formą sudėkite toliau paveiksle matomus komponentus (`TextBox`, `DropDownList`, `CheckBoxList`, `Button`) bei surašykite tekstą.



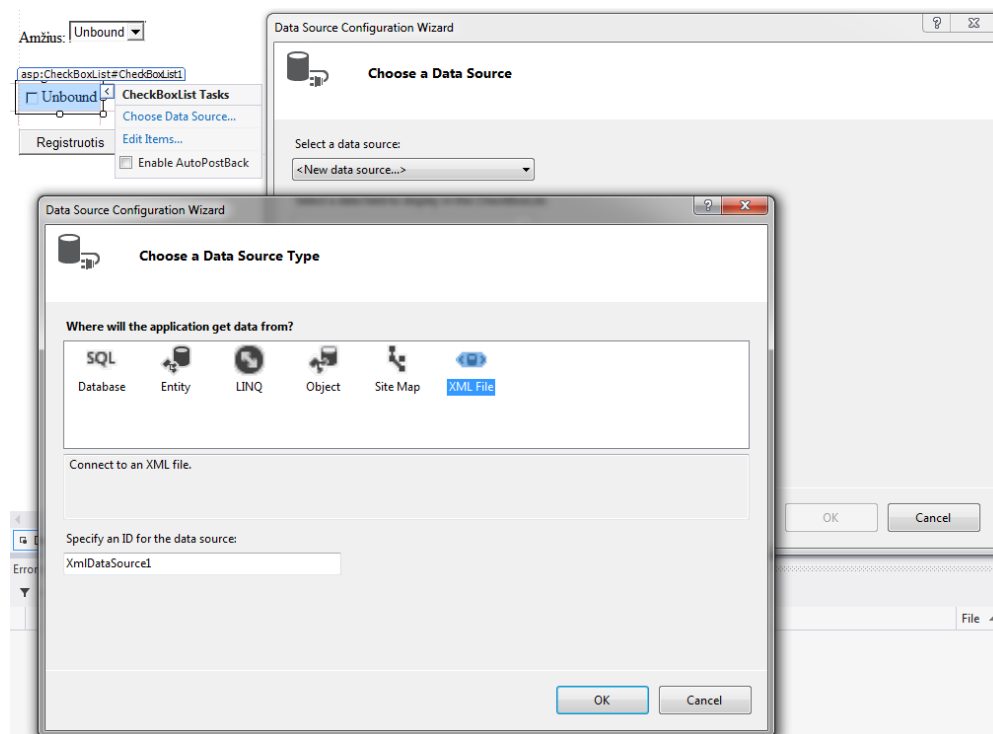
🕒Trečias žingsnis.

- Sukurkite projekte **App_Data** katalogą bei jame naują XML failą pavadinimu **kalbos.xml**. Užpildykite XML failą toliau duotu tekstu:

```
<?xml version="1.0" encoding="utf-8" ?>
<kalbos>
  <kalba pavadinimas="C#"/>
  <kalba pavadinimas="C++"/>
  <kalba pavadinimas="Paskalis"/>
  <kalba pavadinimas="COBOL"/>
  <kalba pavadinimas="Fortran"/>
  <kalba pavadinimas="Asembleris"/>
</kalbos>
```

🕒Ketvirtas žingsnis.

- Pasirinkę **CheckBoxList** nueikite ant šalia esančios rodyklės ir paspauskite **Choose Data Source**. Atsidariusiame lange pasirinkite **<New Data Source>**, toliau kitame lange **XML file** ir spauskite **OK**. Nurodykite anksčiau sukurtą XML failą.



Po šio žingsnio puslapyje žemiau `CheckBoxList` komponento turi atsirasti `XmlDataSource` komponentas.

Penktas žingsnis.

- Atsidarykite puslapio kodą ir puslapio užkrovimo metu `Page_Load` metode užpildykite `DropDownList` komponentą reikšmėmis:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (DropDownList1.Items.Count == 0)
    {
        DropDownList1.Items.Add("-");
        for (int i = 14; i <= 25; i++)
        {
            DropDownList1.Items.Add(i.ToString());
        }
    }
}
```

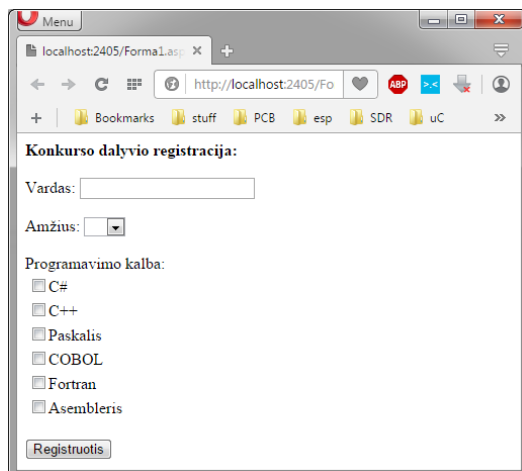
Šeštasis žingsnis.

- Atsidarykite projekto `Web.config` failą ir jame tarp `<configuration>` XML žymių įdėkite toliau parodytą raktą-nustatymą:

```
<configuration>
  <!-- ... -->
  <appSettings>
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />
  </appSettings>
  <!-- ... -->
</configuration>
```

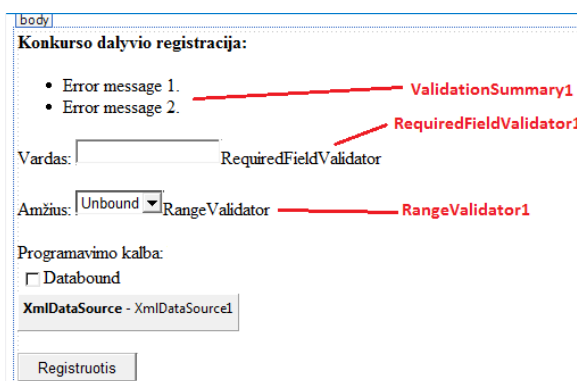
Septintas žingsnis.

- Išbandykite kodo veikimą paleidę puslapį naršyklėje per meniu: DEBUG → Start Debugging. Palyginkite gautą rezultatą su pavyzdžiu toliau. Patikrinkite, ar DropDownList komponentas buvo užpildytas duomenimis.



Aštuntas žingsnis.

- Iš Toolbox Validation srities prie formos komponentų įkelkite naujus komponentus: ValidationSummary, RequiredFieldValidator, RangeValidator kaip parodyta paveiksle toliau.



Devintas žingsnis.

- Visiems validavimo komponentams nustatykite savybes kaip parodyta lentelėje toliau.

Komponentas	RequiredFieldValidator1	RangeValidator1	ValidationSummary1
Savybė			
ErrorMessage	Vardas yra privalomas	Neteisinga metų reikšmė	
Text	*	*	
ControlToValidate	TextBox1	DropDownList1	
ForeColor	Red	Red	Red
MinimumValue		14	
MaximumValue		25	
Type		Integer	

Dešimtas žingsnis.

- Išbandykite veikimą paleidę puslapį naršyklėje per meniu: DEBUG → Start Debugging. Pradžioje nieko neįveskite į laukelius ir bandykite patvirtinti formą paspausdami mygtuką. Palyginkite gautą rezultatą su pavyzdžiu toliau.

Menu

localhost:2405/Forma1.aspx

http://localhost:2405/Forma1.aspx

Bookmarks stuff PCB esp SDR uC NEWS

Konkurso dalyvio registracija:

- Vardas yra privalomas
- Neteisinga metų reikšmė

Vardas: *

Amžius: *

Programavimo kalba:

- ☐ C#
- ☐ C++
- ☐ Paskalis
- ☐ COBOL
- ☐ Fortran
- ☐ Asembleris

Registruotis

Savarankiška užduotis.

Užbaikite kurti konkurso dalyvių registracijos formą leidžiančią įvesti dalyvio vardą ir pavardę, mokyklos pavadinimą, nurodyti amžių bei pažymėti, kokias programavimo kalbas dalyvis moka.

- Panaudokite sesiją asmenų duomenims saugoti.
- Formos apačioje pateikite numeruotą visų jau užregistruotų dalyvių lentelę-sąrašą bei bendrą jų skaičių.
- Įdėkite mygtuką visiems įvestiems sąrašo įrašams pašalinti.
- Įvedimo laukams panaudokite automatinio validavimo komponentus.
- Realizuokite įvestų reikšmių validavimą serverio pusėje (kode).