

Šī laboratorijas darba mērķis ir iepazīties ar izstrādes vidē pieejamajiem rīkiem, kas nodrošina tīmekļa lietojumu vizuālu izstrādi. Tiek pieņemts, ka studentiem ir HTML/XML valodu un HTTP protokola pamatzināšanas.

### 7.1 IESKATS TEHNOLOĢIJĀ ASP.NET

Kā zināms, HTTP protokols nav ar pastāvīgu savienojumu starp serveri un klientu, t.i. katrs URL resursa pieprasījums tiek uzskatīts par pilnīgi neatkarīgu no iepriekšējiem un nākošajiem. Tas nozīmē, ka tīmekļa lietojumprogrammu stāvokļa pārvaldību (saistības nodrošināšanu starp URL resursu pieprasījumiem), izstrādātājiem ir jārealizē pastāvīgi, kas nav triviāls uzdevums. Tāpat HTTP protokols neparedz iespēju aktualizēt tikai daļu no pieprasītā resursa - vienmēr ir jāpārlādē viss pieprasītais resurss.

Lai vienkāršotu tīmekļa lietojumprogrammu izstrādi, tiek piedāvāts liels skaits tīmekļa tehnoloģiju, kas nodrošina papildus abstrakcijas slāni virs protokola HTTP, lai lietojumprogrammas izstrādātājiem piedāvātu stāvokļa pārvaldību, kā arī minimizētu HTML valodas nepastarpinātas pielietošanas nepieciešamību. Vienu no šādiem risinājumiem piedāvā arī tehnoloģija ASP.NET.

Tehnoloģija ASP.NET ir cieši saistīta ar .NET ietvaru, un nodrošina objektorientētu tīmekļa lietojumprogrammu izstrādes ietvaru, piedāvājot t.s. servera puses vizuālos un nevizuālos komponentus, kuri abstrahē HTML/JavaScript valodu .NET klasēs, īpašībās un metodēs. Tomēr, lai pilnvērtīgi spētu pielietot šo tehnoloģiju, izstrādātājam tomēr ir jāpārzina HTML/XML un JavaScript valodu pamatprincipi.

Iepazīsimies ar tehnoloģiju ASP.NET, pakāpeniski izstrādājot tīmekļa lietojumprogrammu.

### 7.2 UZDEVUMS PASNIEDZĒJA VADĪBĀ

Papildināt 5. vai 6. laboratorijas darbā izveidoto noslēguma darbu lietojumprogrammu, lai nodrošinātu iespēju studentiem patstāvīgi pieteikties noslēguma darba tēmām, izmantojot globālo tīmekli.

Šim nolūkam, nodrošināt iespēju apskatīt pieejamo tēmu un vadītāju sarakstu noteiktam studiju līmenim, un katrai tēmai norādīt pašlaik pieteikušos studentu skaitu. Lai students pieteiktos tēmai, viņš to vispirms izvēlas no tēmu saraksta, pēc tam aizpilda datus par sevi (vārds, uzvārds) un, apstiprinot ievadīto informāciju, dati tiek reģistrēti datu bāzē, bet studentam tiek atgriezta informācija par reģistrācijas sekmību:

← → ↻ 🏠 localhost:1208/TopicSelection.aspx

## Noslēguma darbu tēmas rezervēšana

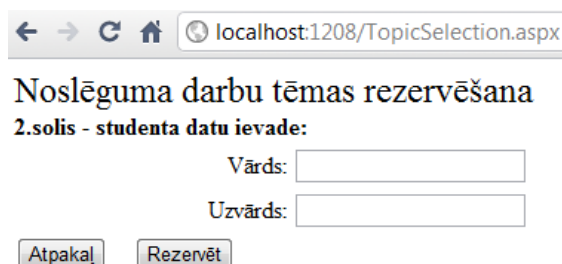
**1.solis - studiju līmeņa un tēmas izvēle:**

Studiju līmenis:

Vadītājs	Tēma	Rezervāciju skaits	
Alksnis, Gundars	Kategoriju teorija datorzinātnēs	1	<input type="button" value="Izvēlēties"/>
Ņikiforova, Oksana	Datu bāzu vadība Delphi vidē	1	<input type="button" value="Izvēlēties"/>
Ņikiforova, Oksana	Klases un objekti sistēmas izstrāde	0	<input type="button" value="Izvēlēties"/>
Rusakovs, Pāvels	Algoritmi informācijas aizsardzībai	1	<input type="button" value="Izvēlēties"/>
Rusakovs, Pāvels	Datu bāzu publicēšana globālajā tīmeklī	0	<input type="button" value="Izvēlēties"/>

Lai turpinātu, izvēlieties sev vēlamo studiju līmeni un noslēguma darba tēmu.

Pēc tēmas izvēles un pogas **Turpināt** nospiešanas, studentam jānorāda vārdu un uzvārdu, kā arī jānodrošina iespēju atgriezties pie tēmu saraksta, lai mainītu savu izvēli:



Nospiežot pogu **Rezervēt**, jānotiek pārbaudei, vai vārda un uzvārda lauki ir aizpildīti – ja kāds no tiem ir tukšs, jāizvada brīdinošs ziņojums un rezervēšana jāatceļ.

Šī uzdevuma tīmekļa lietojuma realizācijai tiks izmantota izstrādes vidē atbalstītā tehnoloģija ASP.NET 3.5. Šim nolūkam:

- Risinājumam **QualificationWorks** pievienot jaunu valodas C# apakš kategorijas **Web** projektu ar sagatavi **ASP.NET Web Application**, nosaucot to par **QualifWorksWebApp**.
- Projekta **QualifWorksWebApp** atsaucēs norādīt projektu **DataModel**.
- No projekta izdzēst artefaktu **Default.aspx**.
- Projektam **QualifWorksWebApp** pievienot jaunu artefaktu ar sagatavi **Site.Master** (jeb **Master Page** Visual Studio 2008 versijā), kuru nosaukt par **Site.Master**. Šī sagatave nodrošinās, ka tīmekļa lietojumprogrammai būs saskaņota saskarne, kas paliks nemainīga, mainoties tikai saturam, kura vietu definē ASP.NET tags `<asp:ContentPlaceholder>`:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs"
Inherits="QualifWorksWebApp.Site" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
  <asp:ContentPlaceholder ID="head" runat="server">
</asp:ContentPlaceholder>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:ContentPlaceholder ID="ContentPlaceholder1" runat="server">
</asp:ContentPlaceholder>
    </div>
  </form>
</body>
</html>
```

Pēc noklusēšanas šajā failā ir definēti divi ASP.NET tagi `<asp:ContentPlaceholder>` – viens sekcijai `<head>`, otrs – sekcijai `<body>`.

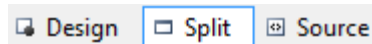
Izstrādes vides ASP.NET lapu redaktors jebkuru ASPX lapu, kas definē HTTP URL resursu, var atspoguļot trīs režīmos:

**Design** – ir redzams tikai lapas grafiskais dizains;

**Split** – ir redzamas gan lapas ASP.NET iezīmes (*markup*), gan grafiskais dizains;

**Source** – ir redzamas tikai ASP.NET lapas iezīmes.

Pārslēgšanos starp šiem režīmiem tiek veikta ar redaktora loga apakšā esošajām pogām:



- Projektam **QualifWorksWebApp** pievienot jaunu artefaktu ar sagatavi **Web Form using Master Page** (jeb **Web Content Form** Visual Studio 2008 versijā), kuru nosaukt par **TopicSelection.aspx**.

---

**Padoms:** *To pašu var palīdzēt, artefaktam **Site.Master** pārslēgt ASP.NET lapu redaktoru **Split** režīmā un grafiskā dizaina apgabalā (loga apakšējā daļā) atvērt konteksta izvēlni un izvēlēties punktu **Add Content Page**. Pēc šīm darbībām nepieciešams pārsaukt (refaktorēt) noklusēto satura lapas nosaukumu no **WebForm1.aspx** uz **TopicSelection.aspx**.*

---

Ar šo komandu projektam tiek pievienota jauna satura lapa, kas pārlūkprogrammā tiks integrēta **Site.Master** tagu `<asp:ContentPlaceHolder>` vietā, bet izstrādes vidē tiek atvērta jaunpievienotā satura lapa ar šādu pirmkodu:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true"
CodeBehind="TopicSelection.aspx.cs" Inherits="QualifWorksWebApp.TopicSelection" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
</asp:Content>
```

Ievietojot ASP.NET tagus vienā no tagiem `<asp:Content>`, pie ASPX lapas parādīšanas pārlūkprogrammā, tie tiks ievietoti atbilstošās **Site.Master** vietās.

Pašlaik URL resursā **TopicSelection.aspx** nav ievietots neviens komponents, tomēr to jau var aplūkot pārlūkprogrammā. Izvēlētās tīmekļa lapas atvēršanu pārlūkprogrammā veic projekta pārlūkā no lapas konteksta izvēlnes izvēloties punktu **View in Browser**.

- Pārbaudīt ASPX labas atvēršanu, izvēloties punktu **View in Browser**. Automātiski tiks palaists izstrādes vides tīmekļa servera process **WebDev.WebServer.exe**, atvērta noklusētā tīmekļa pārlūkprogramma, kuras adreses laukā tiek ierakstīta izstrādes vides tīmekļa servera URL adrese un izvēlētās tīmekļa lapas nosaukums.

---

**Padoms:** *Ja izpildot komandu **View in Browser**, pārlūkprogrammas vietā tiek izvadīts kļūdas ziņojums, izstrādes vides tīmekļa serveri var palaist arī no komandrindas, piemēram, Visual Studio 2008 versijai:*

```
"C:\Program Files\Common Files\Microsoft Shared\DevServer\9.0\WebDev.WebServer.exe"
/port:8888 /path:"C:\Work\Lab7\QualifWorksWebApp"
```

*Bet Visual Studio 2010 versijai:*

```
"C:\Program Files\Common Files\microsoft shared\DevServer\10.0\webDev.WebServer40.exe"
/port:8888 /path:"C:\Work\Lab7\QualifWorksWebApp"
```

*Pēc šīs komandas palaišanas, ir jāatver tīmekļa pārlūkprogramma, un tās adresu laukā jāieraksta:*

```
http://localhost:8888/TopicSelection.aspx
```

---

Kā jau iepriekš tika minēts, tehnoloģija ASP.NET piedāvā objektorientētu programmēšanas modeli - kad lietotājs pārlūkprogrammā ieraksta URL adresi ASPX resursam, serverī šis resurss tiek nokompilēts .NET klasē un izveido tās instanci. Šī iemesla dēļ ASPX resursa pirmā ielāde ir ar nelielu aizturi - katra nākamā griešanās pie šī paša resursa, jau tiks apstrādāta nokompilētā klasē un tiks apstrādāta ātrāk.

Faktiski ASP.NET tīmekļa lietojumprogrammu programmēšanu var sadalīt divos posmos – vizuālās prezentācijas projektēšana ar ASP.NET servera puses komponentiem, un šo komponentu notikumu apstrādātāju programmēšanas ar C# pirmkodu, piemēram, reaģējot uz lapas ielādes un lietotāja darbībām ar kontrolēm.

Lai par to pārliecinātos, ASPX lapas **TopicSelection.aspx** redaktora loga konteksta izvēlē izvēlēties punktu **View Code**, kā rezultātā tiks atvērts lapai piesaistītās klases pirmkods:

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace QualifWorksWebApp
{
    public partial class TopicSelection : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
    }
}

```

Redzams, ka klase **TopicSelection** ir daļēja (**partial**), bet par bāzes klasi tiek norādīta klase **System.Web.UI.Page**, kas automātiski nodrošina funkcionalitāti, kas ir kopēja visām ASPX lapām.

Klasē definētā metode **Page\_Load()** ļauj realizēt notikuma **Page\_Load** funkcionalitāti, kuru nepieciešams izpildīt pie katras šī URL resursa pieprasīšanas.

- Papildināt šīs metodes realizāciju ar šādām pirmkoda rindām:

```

protected void Page_Load(object sender, EventArgs e)
{
    //šī rinda nepieciešama tikai tad, ja dati tiek glabāti
    // SQL Compact datu bāzē
    //(Pretējā gadījumā tiks izvadīta kļūda: SQL Server Compact is not
    // intended for ASP.NET development.)
    AppDomain.CurrentDomain.SetData("SQLServerCompactEditionUnderWebHosting"
    , true);
}

```

Līdzīgi, kā grafiskās lietotāja saskarnes gadījumā (izmantojot **Windows Forms** kontroles), arī ASP.NET piedāvā komponentu modeli, kurus uz ASPX lapu var pārvilkt no komponentu loga **Toolbox**. Šajā logā galvenokārt ir definētas t.s. servera kontroles un komponenti. Tas nozīmē, ka šiem komponentiem ir piesaistītas atbilstošas .NET klases, īpašības un metodes, kuras var izsaukt šo komponentu instancēm.

Kaut arī šie komponenti piedāvā objektu modeli, tomēr to 'izpildes' rezultāts lietotāja pārlūkprogrammā tiek atspoguļots HTML un JavaScript valodu formātā - lietotāja pārlūkprogrammā šo komponentu instances netiek veidotas un izpildītas.

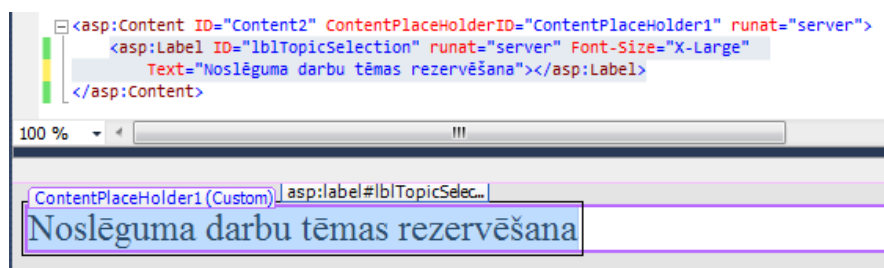
Komponentu logā piedāvātie komponenti nav pilnībā identiski tiem, kuri tiek izmantoti grafiskās lietotāja saskarnes projektēšanai - lai gan komponentu logā ir minētas kontroles, kuru nosaukumi ir identiski ar grafiskās lietotāja saskarnes kontrolēm (piem., **Label** un **TextBox**), to īpašības faktiski nepārklājas. ASP.NET komponentiem ir mazāks īpašību, metožu un iespējamo notikumu skaits. Tas ir skaidrojams ar HTML valodas un HTTP protokola īpatnībām un ierobežojumiem. Piemēram, grafiskās lietotāja saskarnes kontroļu instanču nosaukumi tika definēti atribūtā (**Name**), bet ASP.NET servera kontrolēm šo lomu spēlē atribūts (**ID**).

Iepazīšanos ar ASP.NET servera kontroļu un komponentu īpašībām veiksīm pakāpeniski izstrādājot prasīto tīmekļa lietojumprogrammu:

- Atvērt ASP.NET lapas **TopicSelection.aspx** redaktoru.
- Redaktora loga grafiskajā daļā no rīku loga uzvilkt kategorijas **Standard** kontroli **Label**, uzstādīt īpašības:

Īpašība:	Vērtība:
(ID)	lblTopicSelection
Text	Noslēguma darbu tēmas rezervēšana
Font.Size	X-Large

Ievērot, kā lapas iezīmes daļā tiek aktualizēts tās ASPX saturs, un kā tajā tiek atspoguļotas izmaiņas īpašības:



- No rīku loga, zem tikko uzvietotās kontroles **Label**, uzvilkt kontroli **MultiView**. Pēc pielietošanas mērķa, šī kontrole atgādina Windows Forms kontroli **TabControl**, nodrošinot, ka pārlūkprogrammā vienlaicīgi tiek atspoguļota tikai viena no kontroles **View** (līdzība ar **TabPage**) instancēm.

- Uzvietotajai **MultiView** kontrolei uzstādīt īpašības:

Īpašība:	Vērtība:
(ID)	mvTopicSelection
ActiveViewIndex	0

- Uz kontroles **mvTopicSelection** uzvietot kontroli **View** un uzstādīt tās īpašības (**ID**) vērtību uz **vwTopicSelection**.

- Kontrolē **vwTopicSelection** ievietot jaunu kontroli **Label**, uzstādīt īpašības:

Īpašība:	Vērtība:
(ID)	lblFirstStep
Text	1.solis - studiju līmeņa un tēmas izvēle:
Font.Bold	True

- Jaunā rindā zem kontroles **lblFirstStep** ievietot jaunu kontroli **Label**, uzstādīt īpašības:

Īpašība:	Vērtība:
(ID)	lblQualifLevel
Text	Studiju līmenis:

- Studiju līmeņa izvēle tiks realizēta ar komponenta **DropDownList** palīdzību, kas datus ņems no projekta **DataModel** klases **QualifLevel**. Tomēr, lai to varētu izdarīt, nepieciešams papildināt klasi **QualifLevel** ar noklusēto konstruktoru un nestatisku metodi **GetData()**:

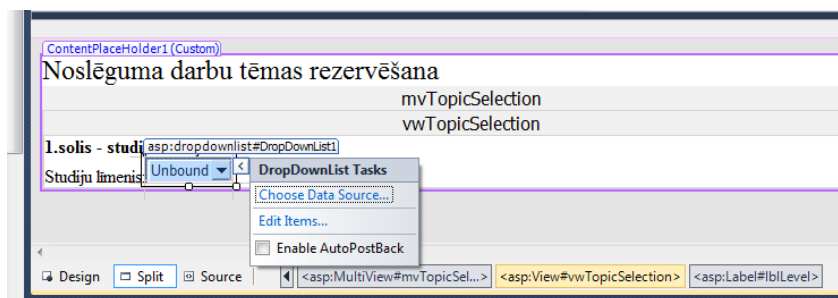
```
public QualifLevel() { }
public ArrayList GetData() { return Levels; }
```

Pēc veiktajām izmaiņām klasē, lai tās stātos spēkā, nokompilēt risinājumu.

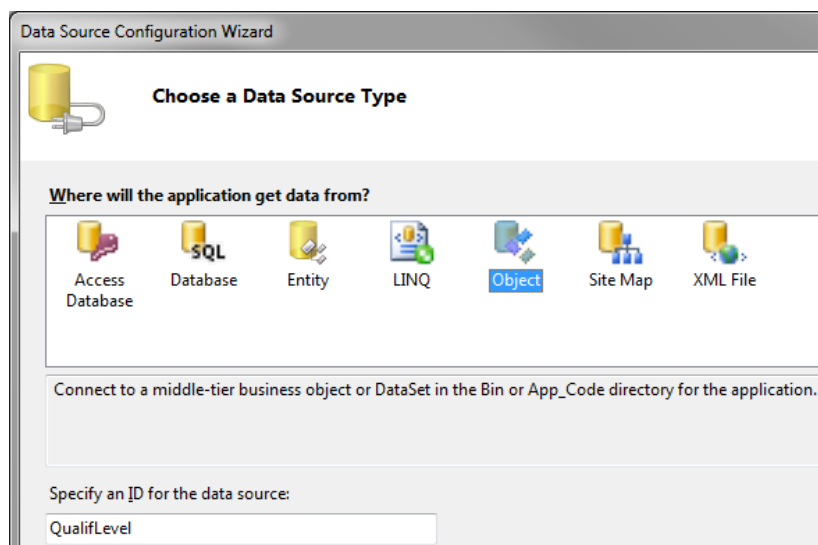
- Atvērt artefakta **TopicSelection.aspx** redaktoru un aiz kontroles **lblQualifLevel** uzvietot kontroli **DropDownList**, un uzstādīt īpašības:

Īpašība:	Vērtība:
(ID)	ddlQualifLevel
AutoPostBack	True

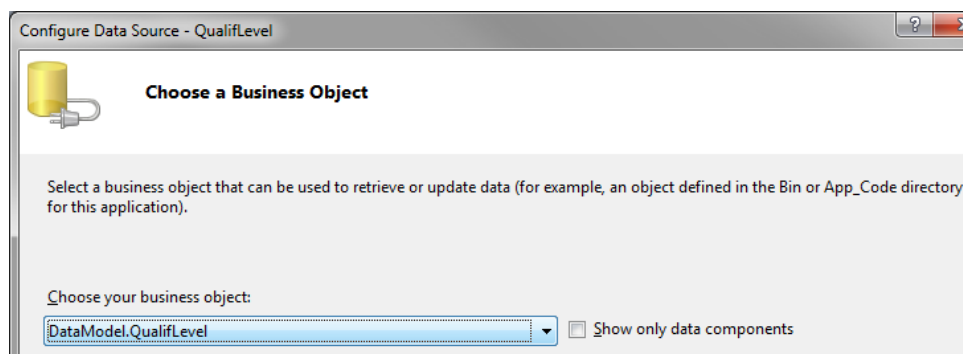
- Lai komponentam **ddlQualifLevel** par datu avotu piesaistītu klasi **QualifLevel**, atvērt šī komponenta izvēlni un izvēlēties punktu **Choose Data Source...**:



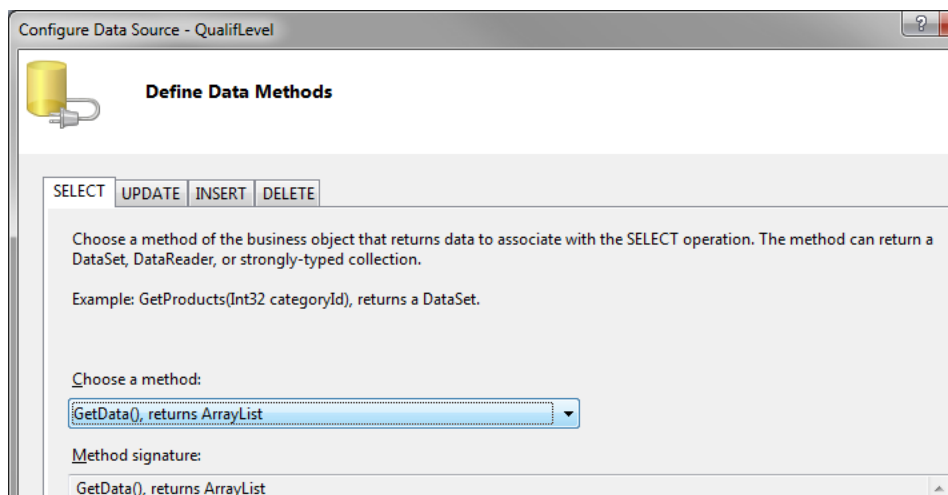
- Atvērsies datu avota izvēles vednis, kura pirmajā logā par datu avotu tipu izvēlēties **Object**, un nosaukt par **QualifLevel**:



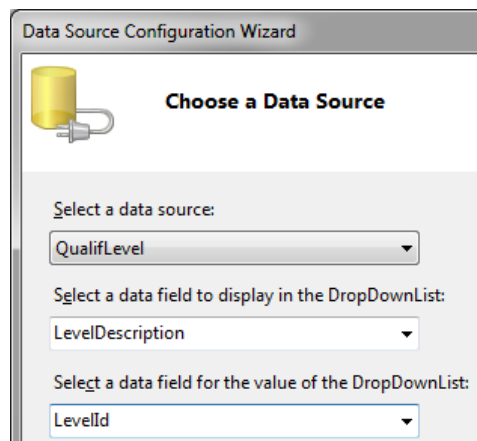
- Nākošajā dialogā, atslēgt izvēles rūtiņu **Show only data components** un kombinētajā sarakstā izvēlēties **DataModel.QualifLevel**:



- Nākošajā dialogā par datu izguves metodi izvēlēties **GetData()**:



- Nospieš pogu Finish, un datu avota konfigurācijas dialogā aizpildīt klases **QualifLevel** īpašības, kas atbilst rādāmajiem un vērtības laukiem (zemākos divus laukus aizpildīt ar roku):



Pirms turpināt ar datu par pieejamajām noslēguma darbu tēmām un to vadītājiem atspoguļošanu, projekta **DataModel** diagrammā **DataModelDataSet.xsd** nepieciešams izveidot jaunu tabulas adapteri, kas nodrošinātu nepieciešamo datu izgūšanu.

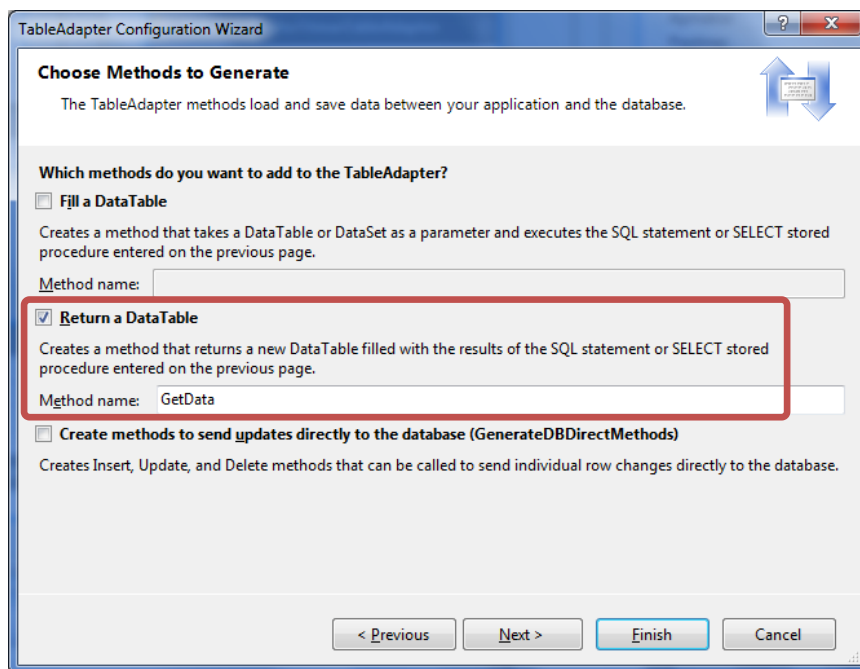
- Pastāvīgi izveidot tabulas adapteri **TopicSelection**, par pamatu izmantojot šādu SQL izteiksmi:

```
SELECT qw.SupervisorID
      , s.Surname + N', ' + s.Name AS Supervisor
      , qw.TopicID
      , qw.Topic
      , CASE WHEN u.Cnt IS NULL THEN 0 ELSE u.Cnt END AS ReservationCnt
FROM   QualifWorks AS qw
       INNER JOIN Supervisors AS s ON qw.SupervisorID = s.SupervisorID
       LEFT OUTER JOIN (SELECT TopicID, COUNT(TopicID) AS Cnt FROM Students GROUP BY
TopicID) AS u ON qw.TopicID = u.TopicID
WHERE  qw.Level <= @QualifLevel
ORDER BY Supervisor, Topic
```

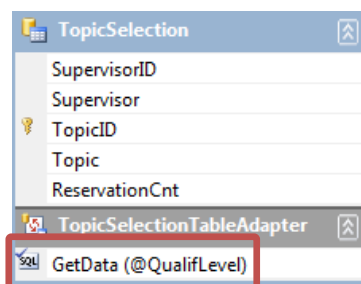
Šis vaicājums, izmantojot parametru **@QualifLevel**, atgriež datus par noslēguma darbu vadītājiem, tēmām un pašreiz pieteikušos studentu skaitu (tiek pieņemts, ka tēmai var nebūt pieteicies neviens students, vai arī vienai un tai pašai tēmai var būt pieteikušies vairāki studenti).



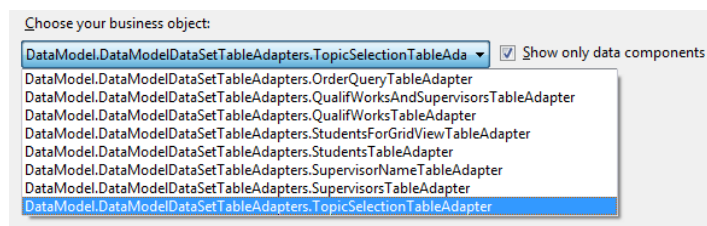
- Tabulu adaptera veidošanas vednī norādīt, ka jāģenerē tikai metode **GetData()**, kas atgriež datus kā objektu **DataTable**:



Pēc šī vedņa aizvēšanas un tabulas adaptera pārdēvēšanas, tas tiks atspoguļots diagrammā:

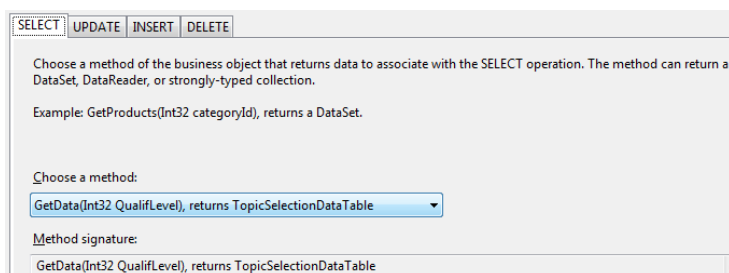


- Lai jaunizveidotajam tabulas adapterim noģenerētos pirmkoda klases, nokompilēt risinājumu.
- Atvērt ASP.NET lapas **TopicSelection.aspx** redaktoru.
- Kontrolē **vwTopicSelection** ievietot kategorijas **Data** kontroli **GridView**, nosaukt to par **gvTopicSelection**.
- Kā datu avotu piesaistīt tabulas adapteri **TopicSelectionTableAdapter**.
- Šim nolūkam, kontroles **gvTopicSelection** izvēlnē izvēlēties punktu **Choose Data Source: <New Data Source...>**. Atvērsies datu avota konfigurācijas vednis, kura pirmajā dialogā par datu avota veidu jāizvēlas **Object**, bet laukā **Specify an ID for the data source** jānorāda **TopicSelectionDataSource**.
- Nospiežot **Next>** jāizvēlas tikko izveidotā tabulas adaptera klase:

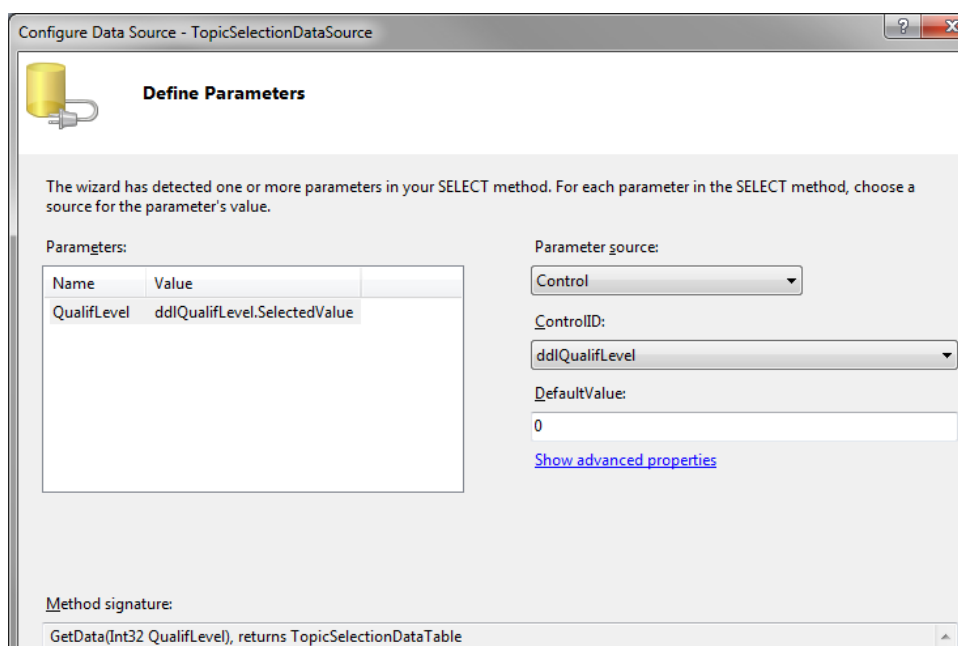




- Nākošajā dialogā pārliecināties, ka datu avota SELECT metodei ir piesaistīta klases metode **GetData()**:



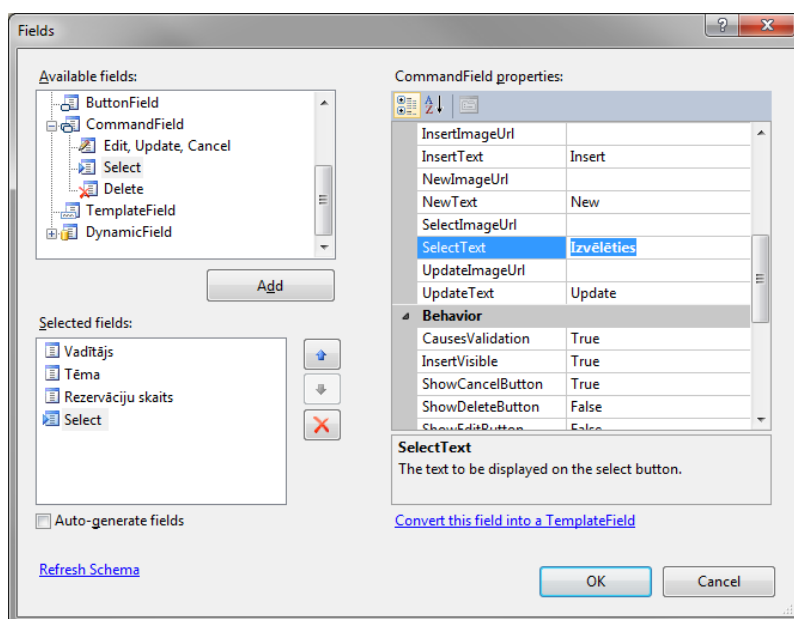
- Nākošajā dialogā ir jāsaista metodes **GetData()** parametrs ar kontroles **ddlQualifLevel** vērtību:



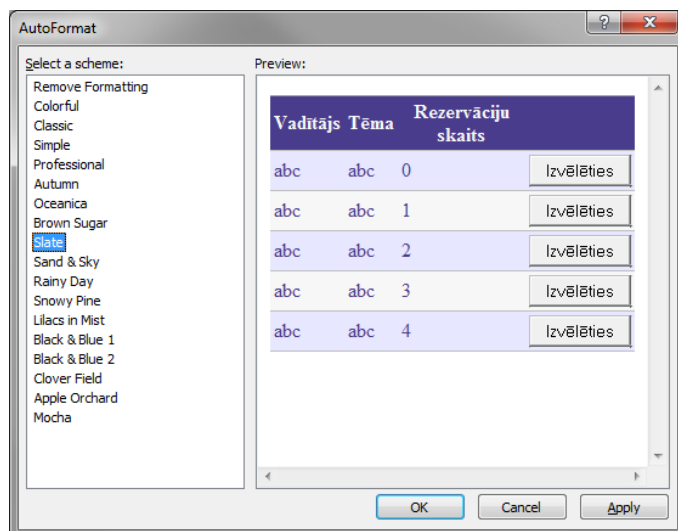
Pabeidzot datu konfigurēšanas vedni, kontrolē automātiski tiks ievietotas piesaistītā datu avota tabulas adaptera kolonnas.

- Atvērt kontroles kolonnu redaktora logu un izdzēst kolonnas **SupervisorID** un **TopicID**, bet kolonnu **Supervisor**, **Topic** un **ReservationCnt** atribūtam **Header Text** attiecīgi uzstādīt vērtības "Vadītājs", "Tēma" un "Rezervāciju skaits".

- No saraksta **Available fields:** pievienot virsotni **CommandField | Select:**



- Jaunpievienotajai kolonnai **Select**, atribūta **SelectText** īpašību nomainīt uz "Izvēlēties", bet īpašību **ButtonType** uz **Button**.
- Atgriezties izstādes vidē un kontroles izvēlnē izvēlēties punktu **Auto Format...** nomainīt kontroles **gvTopicSelection** formatējumu uz kādu no formatējuma shēmu sarakstā pieejamo, piemēram, **Slate**:



Aizverot dialogu, automātiski izvēlēta formatējuma shēma tiks piemērota kontrolei **gvTopicSelection**.

Uzstādīt sagatavi, kas tiks izvadīta, ja kontroles **gvTopicSelection** datu avota nosacījumam neatbilst neviena tēma:

- Kontroles **gvTopicSelection** konteksta izvēlnē izvēlēties punktu **Edit Template | EmptyDataTemplate**, un laukā kas parādīsies kontroles vietā, norādīt tekstu: "Izvēlētajam studiju līmenim netiek piedāvāta neviena tēma."
- Izvēlēties konteksta izvēlnes punktu **End Template Editing**, pēc kā atkal tiks parādītas kontroles **gvTopicSelection** kolonnas.

- Nokompilēt risinājumu lai pārlicinātos, ka tajā nav kļūdas. Projekta pārlūkā uz lapas **TopicSelection.aspx** virsotnes konteksta izvēlnes izvēlēties punktu **View in Browser** un pārlicināties, ka pārlūkprogrammā var izvēlēties studiju līmeni, un atbilstoši tam, tabulas veidā tiek izvadīts vaicājuma rezultāts par izvēlētam studiju līmenim pieejamām noslēguma darbu tēmām. Tabulas pēdējā kolonnā tiek attēlotas pogas **Izvēlēties**, kuras nospiežot tiek izvēlēta atbilstošā tabulas rinda.
- Kontrolē **mvTopicSelection** ievietot vēl vienu kontroli **View** un nosaukt to par **vwStudentData**.

Kontrolē **vwStudentData** ievietot kontroli **Label** un uzstādīt īpašības:

Īpašība:	Vērtība:
(ID)	lbl2solis
Text	2.solis - studenta datu ievade:
Font.Bold	True

No kontroļu kategorijas **HTML**, kurā ir pieejami standarta HTML elementi (nejaukt ar tāda paša nosaukuma ASP.NET servera kontrolēm kategorijā **Standard!**) ievietot kontroli **Table**, un izmantojot konteksta izvēlnes komandas **Delete | Delete Columns** un **Delete Rows**, izdzēst pēdējo kolonnu un pēdējo rindu, lai paliktu divas rindas un divas kolonnas.

Tabulas šūnās no kontroļu kategorijas **Standard** ievietot divas kontroles **Label** un divas kontroles **TextBox**, izvietojot tās pa tabulas šūnām:

Īpašība:	Vērtība:
(ID)	lblStudentName
Text	Vārds:

Īpašība:	Vērtība:
(ID)	lblStudentSurname
Text	Uzvārds:

Īpašība:	Vērtība:
(ID)	txtStudentName

Īpašība:	Vērtība:
(ID)	txtStudentSurname

- Izmantot tabulas šūnu satura izlīdzināšanas iespējas, lai izlīdzinātu ievietotās kontroles.
- No kontroļu kategorijas **Validation**, pa labi no abām **TextBox** kontrolēm ievietot kontroles **RequiredFieldValidator**.

Kontroļu grupā **Validation** ir pieejamas ASP.NET servera kontroles, kas ļauj automatizēt lauku vērtību pārbaudes. Kontrolē **RequiredFieldValidator** nodrošina, ka tai piesaistītajām kontrolēm obligāti būs jābūt norādītām vērtībām, pretējā gadījumā tiks izvadīts kontroles atribūtā **ErrorMessage** uzstādītais teksts.

- Abām **RequiredFieldValidator** instancēm uzstādīt īpašības:

Īpašība:	Vērtība:
(ID)	valStudentName
ControlToValidate	txtStudentName
Display	Dynamic
ErrorMessage	Vārds jānorāda obligāti.

Īpašība:	Vērtība:
(ID)	valStudentSurname
ControlToValidate	txtStudentSurname
Display	Dynamic
ErrorMessage	Uzvārds jānorāda obligāti.

Rezultātā kontrolēm jābūt izvietotām šādi:

- Zem kontroles **mvTopicSelection** ievietot trīs kategorijas **Standard** kontroles **Button**, atdalot tās ar atstarpī (&nbsp;), un uzstādīt īpašības:

**Padoms:** Kontroles no loga **Toolbox** var vilkt un atlaist arī ASPX iezīmju daļā - atlaižot peli noteiktā vietā, tās vietā tiks ievietots kontroles tags. Šādi var rīkoties, ja kontroli nav iespējams novietot grafiskajā daļā tieši nepieciešamajā vietā.

Īpašība:	Vērtība:
(ID)	btnBack
CausesValidation	False
Text	Atpakaļ

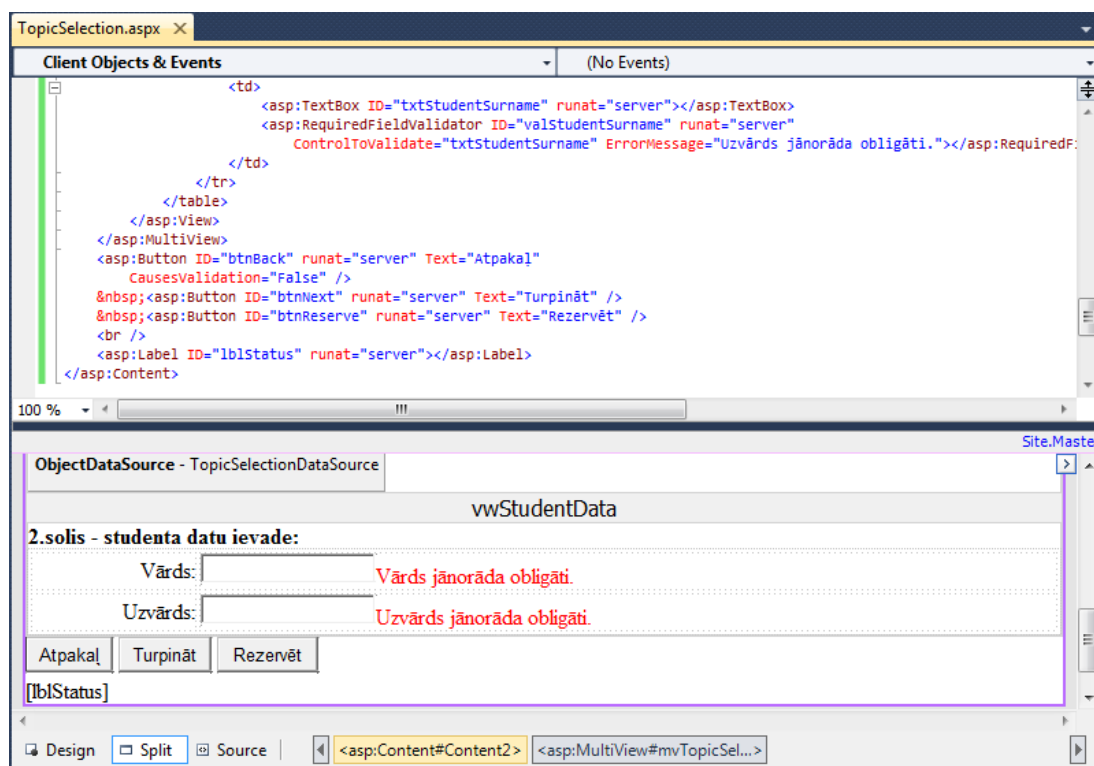
Īpašība:	Vērtība:
(ID)	btnNext
Text	Turpināt

Īpašība:	Vērtība:
(ID)	btnReserve
Text	Rezervēt

Visbeidzot, jaunā rindā zem pogām uzvietot kontroli **Label** un uzstādīt īpašības:

Īpašība:	Vērtība:
(ID)	lblStatus
Text	

Rezultātā, kontroles **vwStudentData** redaktora vizuālajā daļā vajadzētu būt šādam izvietojumam:



Ar šo ir pabeigta lietojumprogrammas vizuālās daļas realizācija - atliek to papildināt ar funkcionalitāti, realizējot pogu **Izvēlēties**, **Atpakaļ**, **Turpināt** un **Rezervēt** nospiešanas notikumu apstrādātājus.

Vispirms URL resursa **TopicSelection.aspx** klasē definēt divas privātas metodes, kuru uzdevums ir inicializēt rādīšanai pārlūka lapā vai nu tēmu saraksta tabulu vai arī studenta datu ievades laukus:

```
namespace QualifWorksWebApp
{
    public partial class TopicSelection : System.Web.UI.Page
    {
        private void SetupView0()
        {
            // uzstāda pārlūka lapā rādāmo View kontroli
            mvTopicSelection.SetActiveView(vwTopicSelection);
            btnNext.Visible = true;
            // turpināt ļaut tikai, ja ir izvēlēta tēma
            btnNext.Enabled = gvTopicSelection.SelectedIndex != -1;
            btnBack.Visible = btnReserve.Visible = false;
        }

        private void SetupView1()
        {
            mvTopicSelection.SetActiveView(vwStudentData);
            btnNext.Visible = false;
            btnBack.Visible = btnReserve.Visible = true;
        }
        //...
    }
}
```

Papildināt metodes **Page\_Load()** realizāciju ar šādu pirmkodu:

```
protected void Page_Load(object sender, EventArgs e)
{
    // ...

    // lapu atverot tiek rādīts tēmu saraksts
    SetupView0();
    if (!btnNext.Enabled)
```

```

        lblStatus.Text = "Lai turpinātu, izvēlieties sev vēlamo studiju līmeni  
un noslēguma darba tēmu.";
    }

```

Lai piesaistītu tēmas izvēles notikumu pogas **Turpināt** aktivizēšanai, iezīmēt kontroli **gvTopicSelection**, īpašību redaktorā pāriet uz notikumu sarakstu un ar peles dubultklikšķi izveidot notikuma **SelectedIndexChanged** apstrādātāju ar šādu realizāciju:

```

protected void gvTopicSelection_SelectedIndexChanged(object sender
, EventArgs e)
{
    btnNext.Enabled = gvTopicSelection.SelectedIndex != -1;
    if (btnNext.Enabled)
        lblStatus.Text = "";
}

```

Ar peles dubultklikšķi uz pogām **Atpakaļ**, **Turpināt** un **Rezervēt**, izveidot notikuma **Click** apstrādātājus ar šādām realizācijām:

```

protected void btnBack_Click(object sender, EventArgs e)
{
    SetupView0();
}

```

```

protected void btnNext_Click(object sender, EventArgs e)
{
    SetupView1();
}

```

```

protected void btnReserve_Click(object sender, EventArgs e)
{
    if (!this.IsValid) // ja lapas lauki nav veiksmīgi validēti
        return;

    try
    {
        // izmantojot noģenerētās klases, pieklūt tabulas Bakalauri datiem
        StudentsTableAdapter taStudents = new StudentsTableAdapter();

        DataModel.DataModelDataSet.StudentsDataTable tblStudents
            = new DataModel.DataModelDataSet.StudentsDataTable();

        // izveido jaunu raksta instanci
        DataModel.DataModelDataSet.StudentsRow newRow
            = tblStudents.NewStudentsRow();

        // uzstāda raksta vērtības
        newRow.Name = txtStudentName.Text;
        newRow.Surname = txtStudentSurname.Text;
        newRow.IDCard = (new Random()).Next(999999999).ToString();
        newRow.Level = Convert.ToInt32(ddlQualifLevel.SelectedValue);
        newRow.Notes = "Rezervēts no tīmekļa.";
        newRow.TopicID = (int)gvTopicSelection.SelectedDataKey["TopicID"];
        // pievieno raksta instanci lokālajai datu kešatmiņai
        tblStudents.AddStudentsRow(newRow);
        // saglabā izmaiņas datu bāzē
        taStudents.Update(tblStudents);
        // aktualizē tēmu saraksta tabulas datus
        gvTopicSelection.DataBind();
        // ziņojums lietotājam par veiksmīgu pievienošanu
        lblStatus.Text = "Tēmas reģistrācija ir veiksmīga!";
        // inicializē laukus sagatavojot jaunas tēmas ievadišanai
        txtStudentName.Text = "";
        txtStudentSurname.Text = "";
        gvTopicSelection.SelectedIndex = -1;
        SetupView0();
    }
    catch (Exception ex)
    {
        lblStatus.Text = ex.Message;
    }
}

```

- Patstāvīgi novērst visas tipu neatpazīšanas kļūdas.

Līdz ar to, sākotnēji izvirzītais darba uzdevums ir izpildīts – nokompilēt risinājumu, lai pārlicinātos, ka kompilēšana ir veiksmīga.

Pārlūkā atvērt tīmekļa lietojuma lapu un pārlicināties, ka:

- Kamēr nav izvēlēta vēlāmā tēma, poga **Turpināt** ir neaktīva;
- Nospiežot pogu **Turpināt**, tiek parādīti lauki vārda un uzvārda ievadei, kā arī poga **Atpakaļ**, ar kuru var atgriezties pie tēmu saraksta;
- Nospiežot pogu **Rezervēt**, tiek pārbaudīts vai lauks **Vārds** un **Uzvārds** ir norādītas vērtības, un ja kāda iztrūkst, tiek izvadīts teksts pa labi no kontroles, kuras validēšana bija neveiksmīga.
- Norādot visu informāciju korekti, nospiežot **Rezervēt** notiek jauna studenta reģistrēšana noslēgumu darbu datubāzē un tīmekļa pārlūkā tiek parādīts tēmu saraksts, ar palielinātu rezervāciju skaitu izvēlētajai tēmai.