

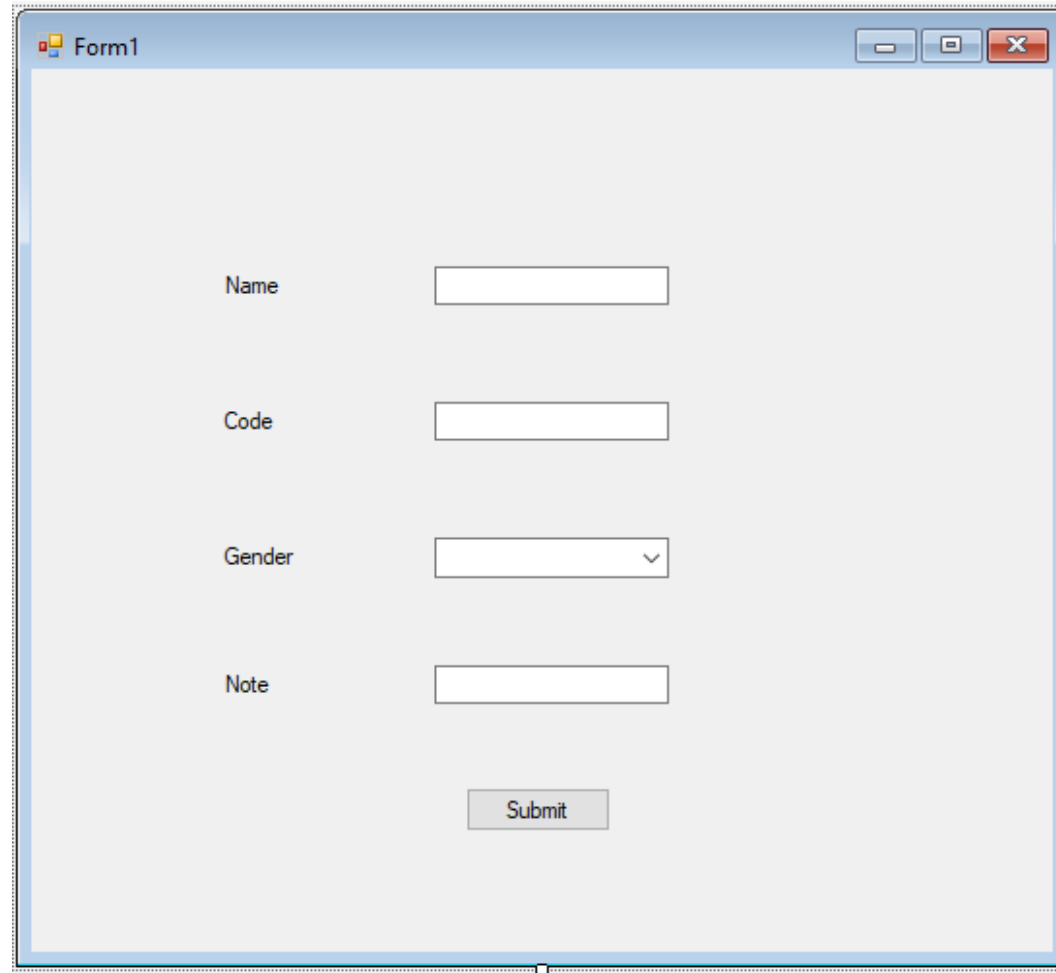


THE LECTURE I I



EXCEPTION HANDLING

ADDING TEXTBOXES



A screenshot of a Windows application window titled "Form1". The window contains a form with four input fields and a submit button. The fields are labeled "Name", "Code", "Gender", and "Note". The "Gender" field is a dropdown menu. The "Submit" button is located at the bottom center of the form.

Name	<input type="text"/>
Code	<input type="text"/>
Gender	<input type="text" value="v"/>
Note	<input type="text"/>

ADDING CONTEXT MENU AND ERRORPROVIDER

The image shows a Windows application window titled "Form1". Inside the window, there is a form with four labels and corresponding input controls: "Name" with a text box, "Code" with a text box, "Gender" with a dropdown menu, and "Note" with a text box. Below these fields is a "Submit" button. The window has a standard Windows title bar with minimize, maximize, and close buttons. At the bottom of the screen, the Windows taskbar is visible, showing three icons: a context menu strip, a color dialog, and an error provider.

Form1

Name

Code

Gender

Note

Submit

contextMenuStrip1 colorDialog1 errorProvider1

CREATING A NEW CLASS

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

namespace MenuEditor
{
    Ссылка: 2
    class Student
    {
        public string name;
        public int code;
        public string gender;
        public int note;
        Ссылка: 0
        public Student()
        {
            this.name = null;
            this.code = 0;
            this.gender = null;
            this.note = 0;
        }
        Ссылка: 0
        public Student(string n, int c, string g, int nt)
        {
            this.name = n;
            this.code = c;
            this.gender = g;
            this.note = nt;
        }
        Ссылка: 0
        public int GetCode()
        {
            return this.code;
        }
        Ссылка: 0
        public void SetCode(int value)
        {
            this.code = value;
        }
    }
}
```

```
public string Name
{
    get { return this.name; }
    set { this.name = value; }
}

Ссылка: 0
public string Gender
{
    get { return this.gender; }
    set { this.gender = value; }
}

Ссылка: 0
public int Note
{
    get { return this.note; }
    set { this.note = value; }
}
```

ADDING ERROR MESSAGES

```
ссылка: 1
private void button1_Click(object sender, EventArgs e)
{
    int formStatus = 0;
    if(tbName.Text == "")
    {
        formStatus++;
        errorProvider1.SetError(tbName, "Enter your name!");
    }
    if(tbCode.Text == "")
    {
        formStatus++;
        errorProvider1.SetError(tbCode, "Enter the code!");
    }
    if(cbGender.Text == "")
    {
        formStatus++;
        errorProvider1.SetError(cbGender, "Enter your gender!");
    }
    if(tbNote.Text == "")
    {
        formStatus++;
        errorProvider1.SetError(tbNote, "Enter the note!");
    }
    if(formStatus == 0)
    {
        try
        {
            Student s = new Student();
            s.Name = tbName.Text;
            s.setCode(Convert.ToInt32(tbCode.Text));
            s.Gender = cbGender.Text;
            s.Note = tbNote.Text;

            throw new Exception("Student has been created");
        }
        catch(FormatException exc)
        {
            MessageBox.Show("The format exception happened", exc.Message);
        }
        finally
        {

```

ENTER THE FORM'S VALIDATION

ССЫЛКА: 1

```
private void tbNote_Validating(object sender, CancelEventArgs e)
{
    try
    {
        string note = tbNote.Text;
        if (note.Length < 10)
            MessageBox.Show("Enter a longer string", "Error", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);
    }
    catch (Exception)
    {
        MessageBox.Show("Invalid data entered", "Error", MessageBoxButtons.OKCancel, MessageBoxIcon.Error);
    }
}
```

ADDING CONTEXT MENU ACTIONS

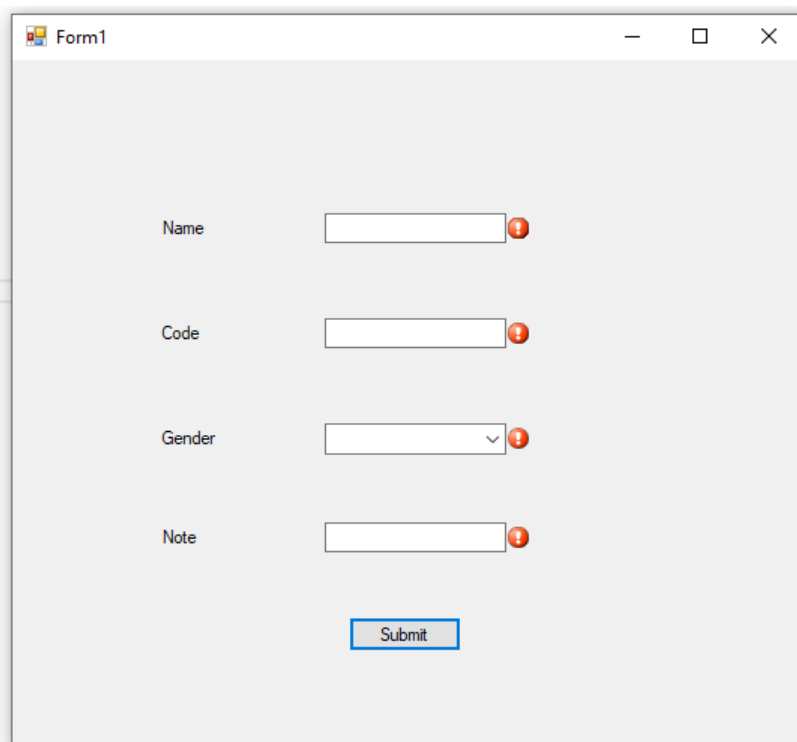
ссылка: 1

```
private void backgroundColorToolStripMenuItem_Click(object sender, EventArgs e)
{
    ColorDialog dlg = new ColorDialog();
    if (dlg.ShowDialog() == DialogResult.OK)
        contextMenuStrip1.SourceControl.BackColor = dlg.Color;
}
```

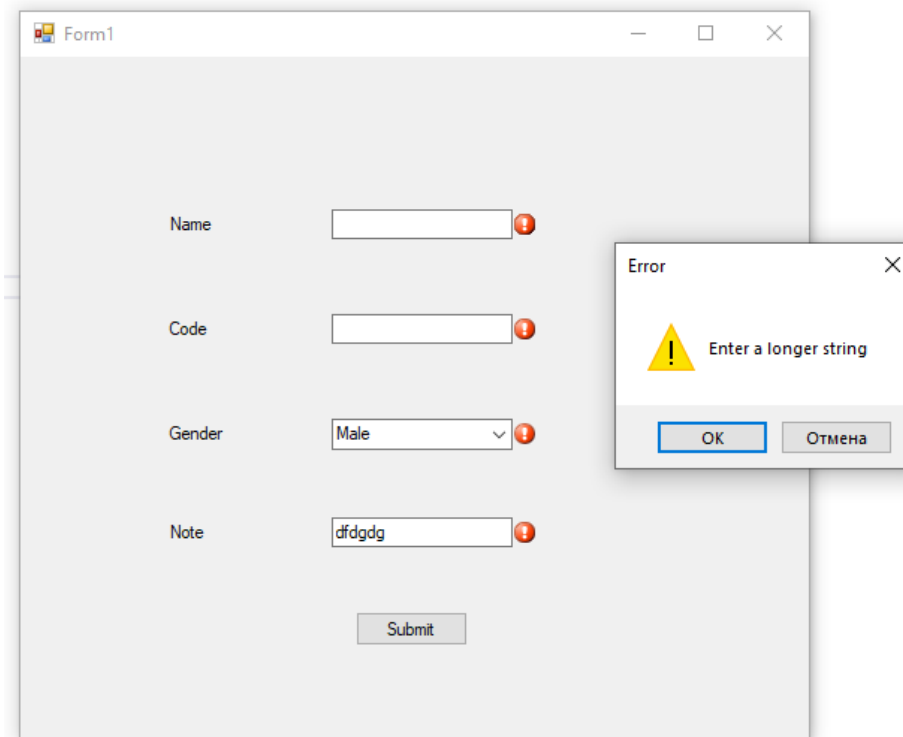
ссылка: 1

```
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

FORM ERRORS



A screenshot of a Windows application window titled "Form1". The window contains four input fields: "Name", "Code", "Gender", and "Note". Each field has a red error icon (a circle with an exclamation mark) to its right. The "Name" and "Code" fields are empty text boxes. The "Gender" field is a dropdown menu with a downward arrow. The "Note" field is an empty text box. Below the input fields is a "Submit" button.



A screenshot of the same "Form1" window, but with an "Error" dialog box open in the foreground. The dialog box has a yellow warning triangle icon and the text "Enter a longer string". It has two buttons: "OK" and "Отмена" (Cancel). In the background, the "Form1" window is visible, showing the same input fields and error icons. The "Submit" button is now disabled (grayed out).