



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ

ГОРОДА МОСКВЫ

Государственное бюджетное профессиональное
образовательное учреждение города Москвы

«Колледж малого бизнеса № 4»

(ГБПОУ КМБ № 4)

Отчёт по лабораторной работе №2

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент: Межибор Ярослав Евгеньевич

Группа: ИПО-21.24

Проверил: Рыбаков Александр Сергеевич

Москва, 2025 г.

Цель:

На практике разобраться, как работают основные визуальные компоненты в C#, научиться оживлять интерфейс через обработку разных событий и компоновать элементы так, чтобы программа выглядела аккуратно.

Задачи:

- Сделать форму, которая реагирует на пользователя (кнопка, убегающая от курсора).
- Поработать с ListBox и ComboBox - научиться программно добавлять в них строки, удалять выбранные пункты и сортировать их.
- Добавить в приложение панель инструментов (ToolStrip) для быстрых команд и строку состояния (StatusStrip) для вывода служебной информации.
- Научиться использовать контейнеры (GroupBox, Panel, TabControl), чтобы логично разделять разные блоки настроек на форме.
- Настроить текстовые поля (TextBox, MaskedTextBox) и сделать так, чтобы программа ругалась (через ErrorProvider), если пользователь вводит в поле для цифр буквы.

Ход работы:

Упражнение 1:

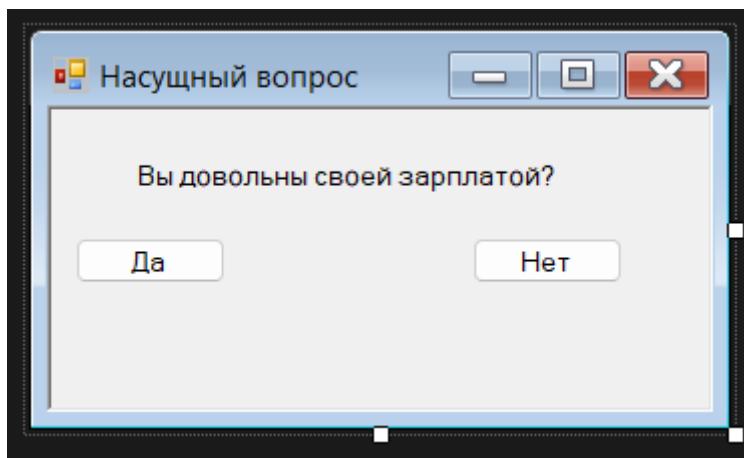


Рис. 1. Создание простой формы с двумя кнопками

```
private void btnyes_Click(object sender, EventArgs e)
{
    MessageBox.Show("Мы и не сомневались, что Вы так думаете!");
}
```

Рис. 2. Код в обработчике события кнопки Да

```

Ссылка: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1
private void btnno_MouseMove(object sender, MouseEventArgs e)
{
    btnno.Top -= e.Y;
    btnno.Left += e.X;
    if (btnno.Top < -10 || btnno.Top > 100)
        btnno.Top = 60;
    if (btnno.Left < -80 || btnno.Left > 250)
        btnno.Left = 120;
}

```

Рис. 3. Код в обработчике события *MouseMove* кнопки *Hem*

Упражнение 2:

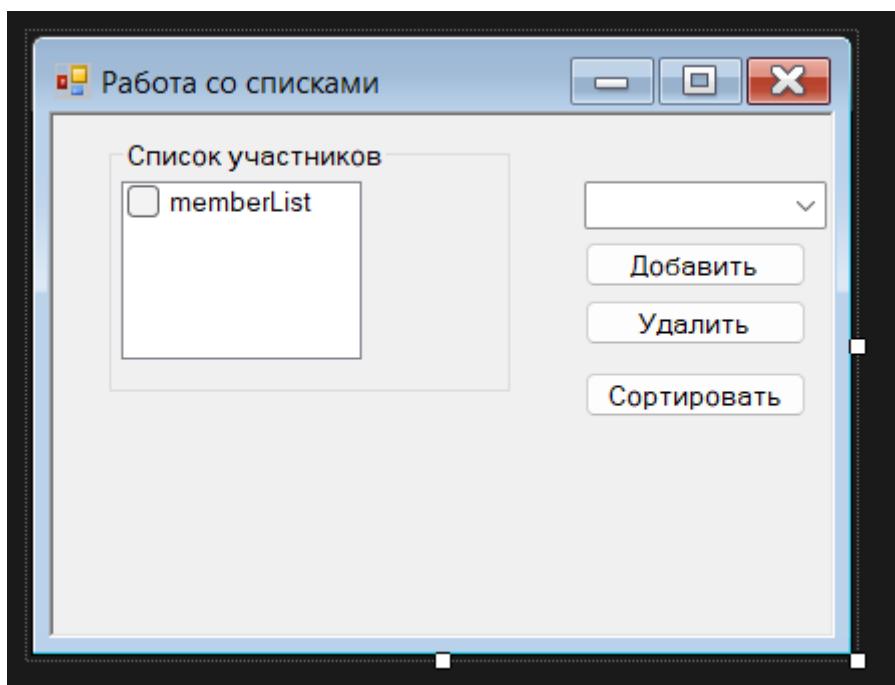


Рис. 4. Создание формы и элементов на ней

```

Ссылка: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1
private void buttonAdd_Click(object sender, EventArgs e)
{
    if (peopleList.Text.Length != 0)
    {
        memberList.Items.Add(peopleList.Text);
    }
    else MessageBox.Show("Выберите элемент из списка или введите новый");
}

```

Рис. 5. Создание кода в обработчике событий кнопки *Добавить*

```
Ссылок: 1 | Язык: C# | Дата: 13 дн. назад | Автор: 1, изменение: 1  
private void buttonDelete_Click(object sender, EventArgs e)  
{  
    while (memberList.CheckedIndices.Count > 0)  
        memberList.Items.RemoveAt(memberList.CheckedIndices[0]);  
}  
}
```

Рис. 6. Создание кода в обработчике событий кнопки Удалить

```
Ссылок: 1 | Язык: C# | Дата: 13 дн. назад | Автор: 1, изменение: 1  
private void buttonSort_Click(object sender, EventArgs e)  
{  
    memberList.Sorted = true;  
}
```

Рис. 7. Создание кода в обработчике событий кнопки Сортировать

Упражнение 3:

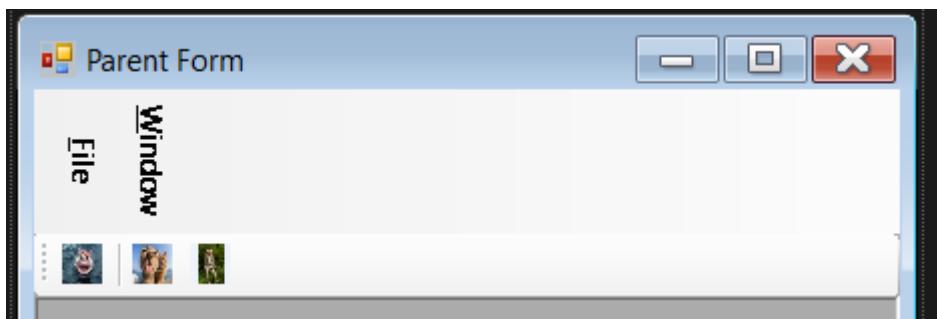


Рис. 8. Создание картинок на элементах панели управления

Ссылок: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void toolStrip1_ItemClicked(object sender, ToolStripItemClickedEventArgs e)
{
    switch (e.ClickedItem.Tag.ToString())
    {
        case "NewDoc":
            Form5 newChild = new Form5();
            newChild.MdiParent = this;
            newChild.Show();
            newChild.Text = newChild.Text + " " +
                ++openDocuments;
            break;
        case "Cascade":
            this.LayoutMdi(System.Windows.Forms.MdiLayout.Cascade);
            break;
        case "Title":
            this.LayoutMdi(
                System.Windows.Forms.MdiLayout.TileHorizontal);
            break;
    }
}
```

Рис. 9. Создание кода обработчика события Click для элемента управления

Упражнение 4:

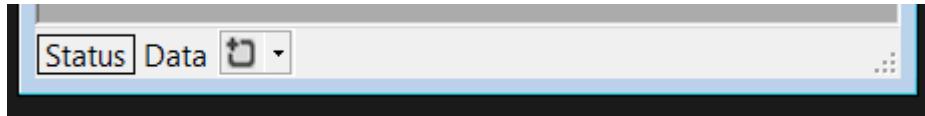


Рис. 10. Создание строки состояния

Ссылок: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void spWin_Click(object sender, EventArgs e)
{
    spWin.Text = "Windows is cascade";
    spWin.Text = "Windows is horizontal";
}
```

Рис. 11. Создание кода для обработки события Click для кнопки Status

Упражнение 5:

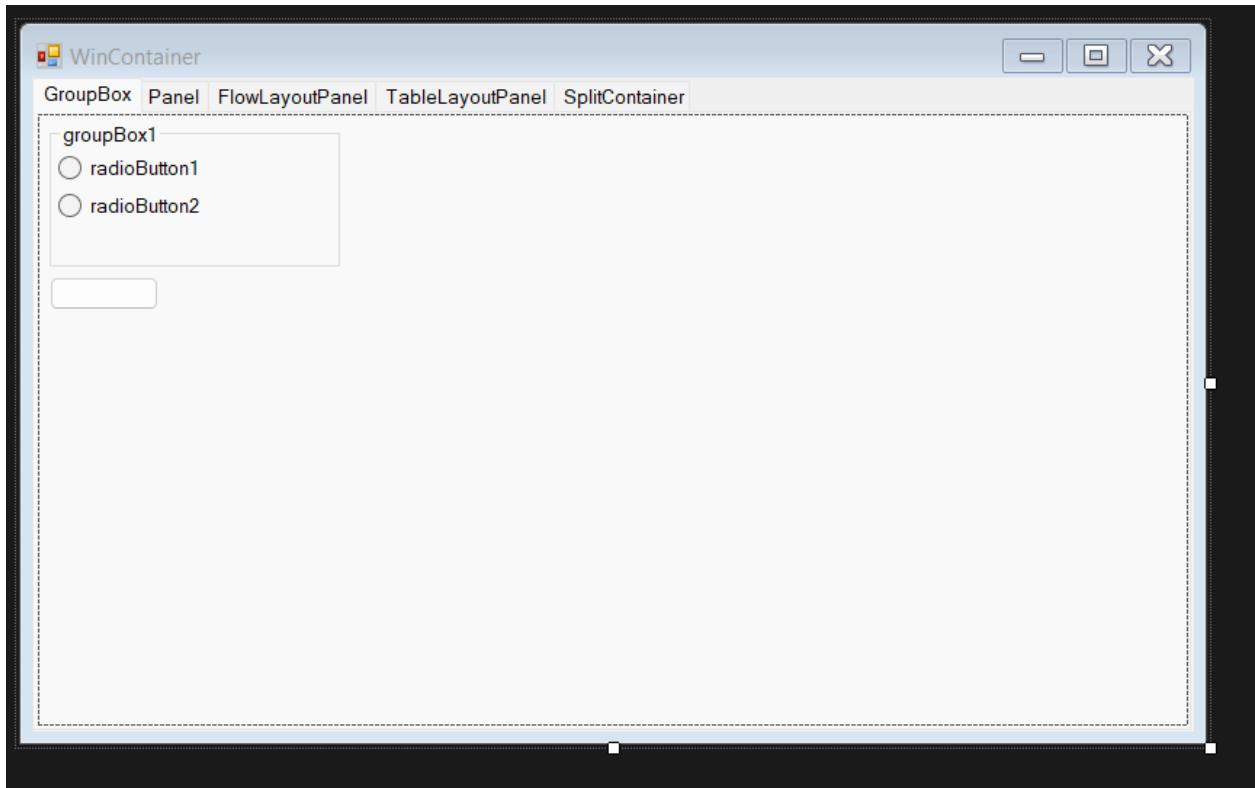


Рис. 12. Создание формы с вкладками и элементами

```
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton1.Checked == true)
        this.but.Text = "First";
    else if (radioButton2.Checked == true)
        this.but.Text = "Second";
}
```

Рис. 13. Создание кода для обработчика события кнопки RadioButton

Ссылок: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void button5_Click(object sender, EventArgs e)
{
    flowLayoutPanel1.SetFlowBreak(button6, true);
}
```

Рис. 14. Создание кода в обработчике события Click для кнопки

Ссылка: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void button9_Click(object sender, EventArgs e)
{
    Button aButton = new Button();
    tableLayoutPanel1.Controls.Add(aButton, 1, 1);
}
```

Рис. 15. Создание кода в обработчике события Click для кнопки

Ссылка: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void button10_Click(object sender, EventArgs e)
{
    if (splitContainer1.FixedPanel == FixedPanel.Panel1)
        splitContainer1.FixedPanel = FixedPanel.None;
    else
        splitContainer1.FixedPanel = FixedPanel.Panel1;
}
```

Рис. 16. Создание кода в обработчике события Click для кнопки

Ссылка: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void button11_Click(object sender, EventArgs e)
{
    splitContainer1.IsSplitterFixed = !(splitContainer1.IsSplitterFixed);
}
```

Рис. 17. Создание кода в обработчике события Click для кнопки

Ссылка: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void button12_Click(object sender, EventArgs e)
{
    splitContainer1.Panel1Collapsed = !(splitContainer1.Panel1Collapsed);
}
```

Рис. 18. Создание кода в обработчике события Click для кнопки

Упражнение 6:

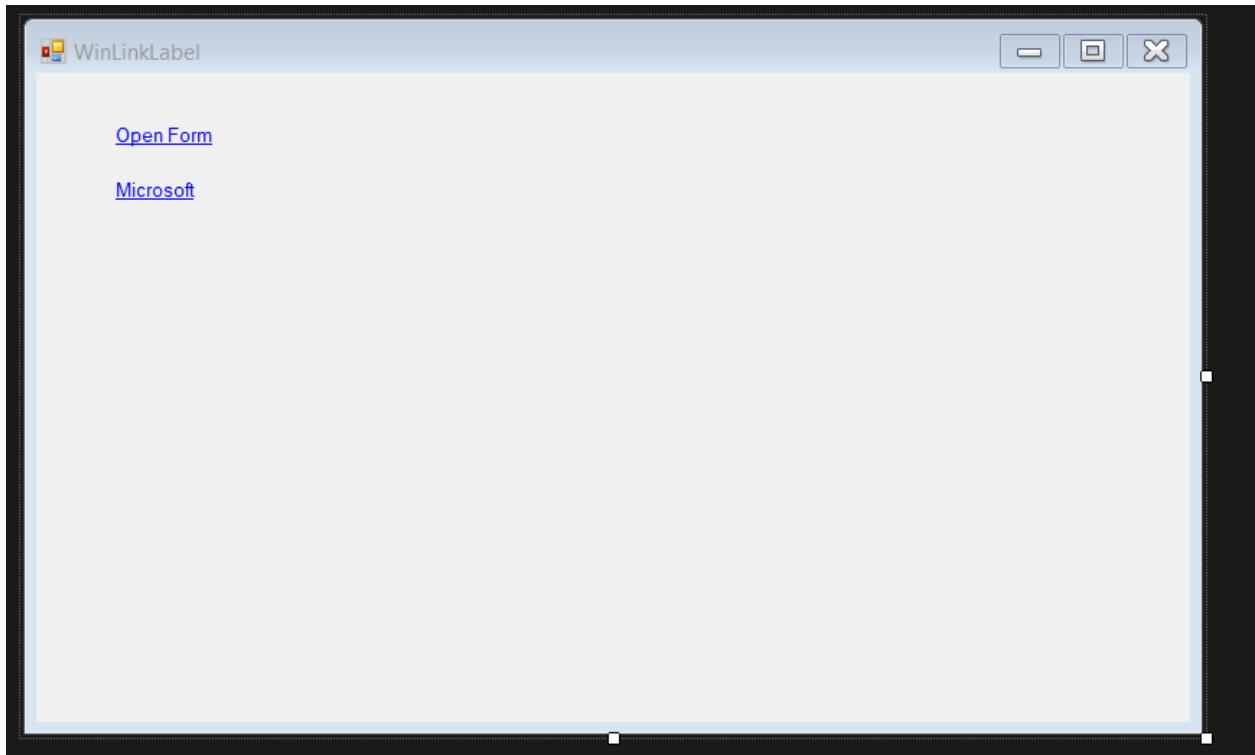


Рис. 19. Создание формы с надписями-ссылками

```
Ссылок: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1
private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    DialogResult aResult;
    Form2 aForm = new Form2();
    aResult = aForm.ShowDialog();
    if (aResult == System.Windows.Forms.DialogResult.OK)
    {
        MessageBox.Show("Your address is " + aForm.textBox2.Text);
        MessageBox.Show("Your phone number is " +
        aForm.maskedTextBox1.Text);

        MessageBox.Show("Your name is " + aForm.textBox1.Text + " " + aForm.textBox2.Text);
    }
    linkLabel1.LinkVisited = true;
}
```

Рис. 20. Создание кода в обработчике события LinkClicked для надписи-ссылки

```
Ссылок: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1
private void linkLabel2_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    System.Diagnostics.Process.Start("www.limtu.com");
    linkLabel2.LinkVisited = true;
}
```

Рис. 21. Создание кода в обработчике события LinkClicked для другой надписи-ссылки, ведущий на сайт

Упражнение 7:

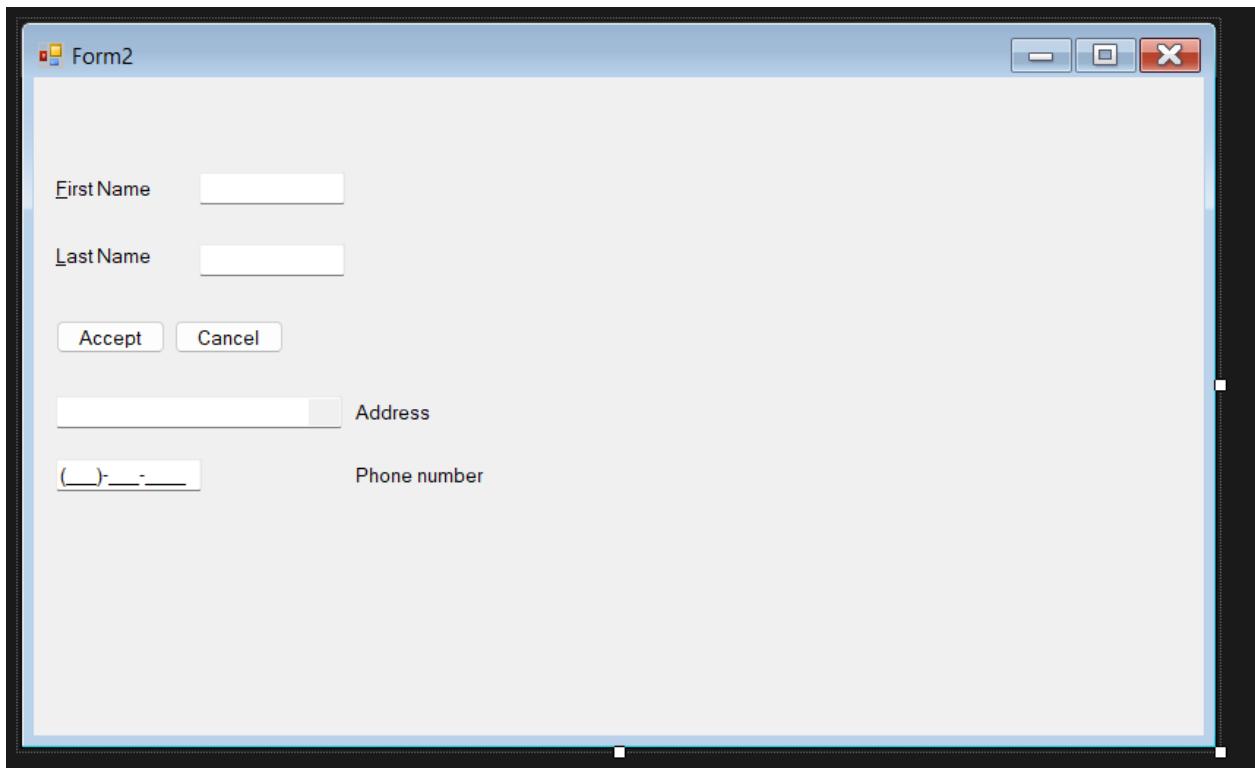


Рис. 22. Создание формы с элементами для ввода данных, в том числе номера телефона

Упражнение 8:

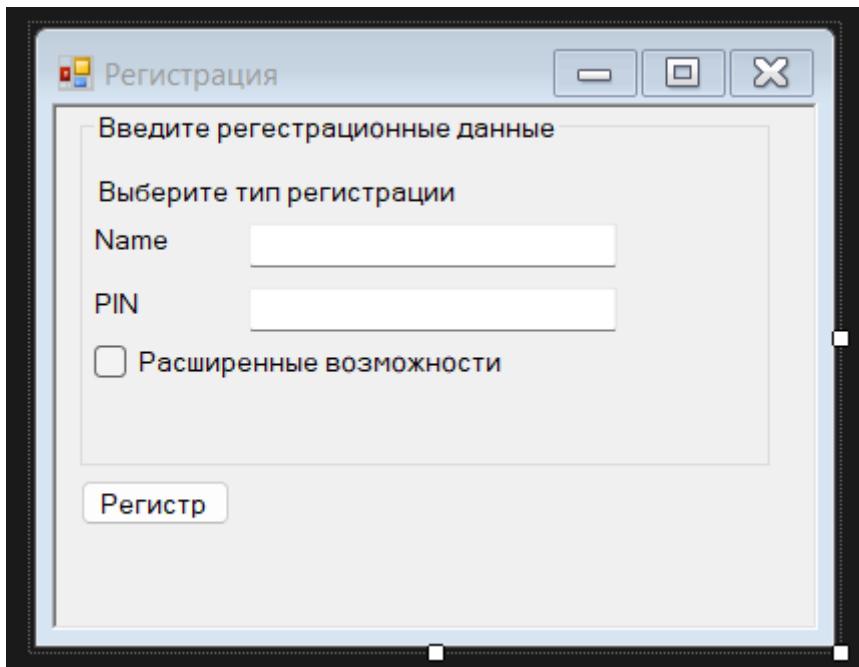


Рис. 23. Создание формы с различными элементами для ввода данных

Ссылок: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked == true)
    {

        Label lbl = new Label();
        lbl.Location = new System.Drawing.Point(16, 96);
        lbl.Size = new System.Drawing.Size(32, 23);
        lbl.Name = "label1";
        lbl.TabIndex = 2;
        lbl.Text = "PIN2";
        groupBox1.Controls.Add(lbl);
        TextBox txt = new TextBox();
        txt.Location = new System.Drawing.Point(96, 96);
        txt.Size = new System.Drawing.Size(184, 20);
        txt.Name = "textboxx";
        txt.TabIndex = 1;
        txt.Text = "";
        groupBox1.Controls.Add(txt);
        txt.KeyPress += new
System.Windows.Forms.KeyPressEventHandler(this.textBox2_KeyPress);

    }
    else
    {
    }
}
```

Рис. 24. Создание кода в обработчике события *CheckedChanged* для строки с местом для галочки

Упражнение 9:

Ссылок: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (char.IsDigit(e.KeyChar))
    {
        e.Handled = true;
        MessageBox.Show("Поле Name не может содержать цифры");
        errorProvider1.SetError(textBox1, "Must be letter");
    }
}
```

Рис. 25. Создание кода для текстового поля 1

Ссылок: 2 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsDigit(e.KeyChar))
    {
        e.Handled = true;
        MessageBox.Show("Поле PIN не может содержать буквы");
    }
}
```

Рис. 26. Создание кода для текстового поля 2

Ссылок: 1 | Yarik, 13 дн. назад | Автор: 1, изменение: 1

```
private void textBox2_Validating(object sender, CancelEventArgs e)
{
    if (textBox2.Text == "")
    {
        e.Cancel = false;
    }
    else
    {
        try
        {
            double.Parse(textBox2.Text);
            e.Cancel = false;
        }
        catch
        {
            e.Cancel = true;
            MessageBox.Show("Поле PIN не может содержать буквы");
        }
    }
}
```

Рис. 27. Создание обработчика события Validating

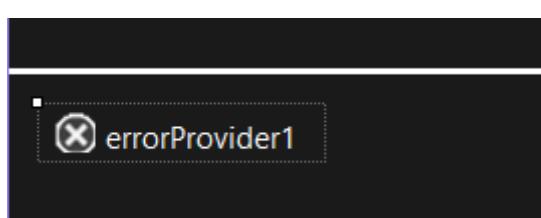


Рис. 28. Создание элемента ErrorProvider

Вывод:

В этой лабе я я понял принцип работы коллекций (Items) у списков и то, как события (типа MouseMove или KeyPress) позволяют полностью контролировать поведение пользователя. Ещё стало понятно, что интерфейс нужно сразу разбивать на логические блоки с помощью контейнеров, иначе при изменении размеров окна всё превращается в кашу.