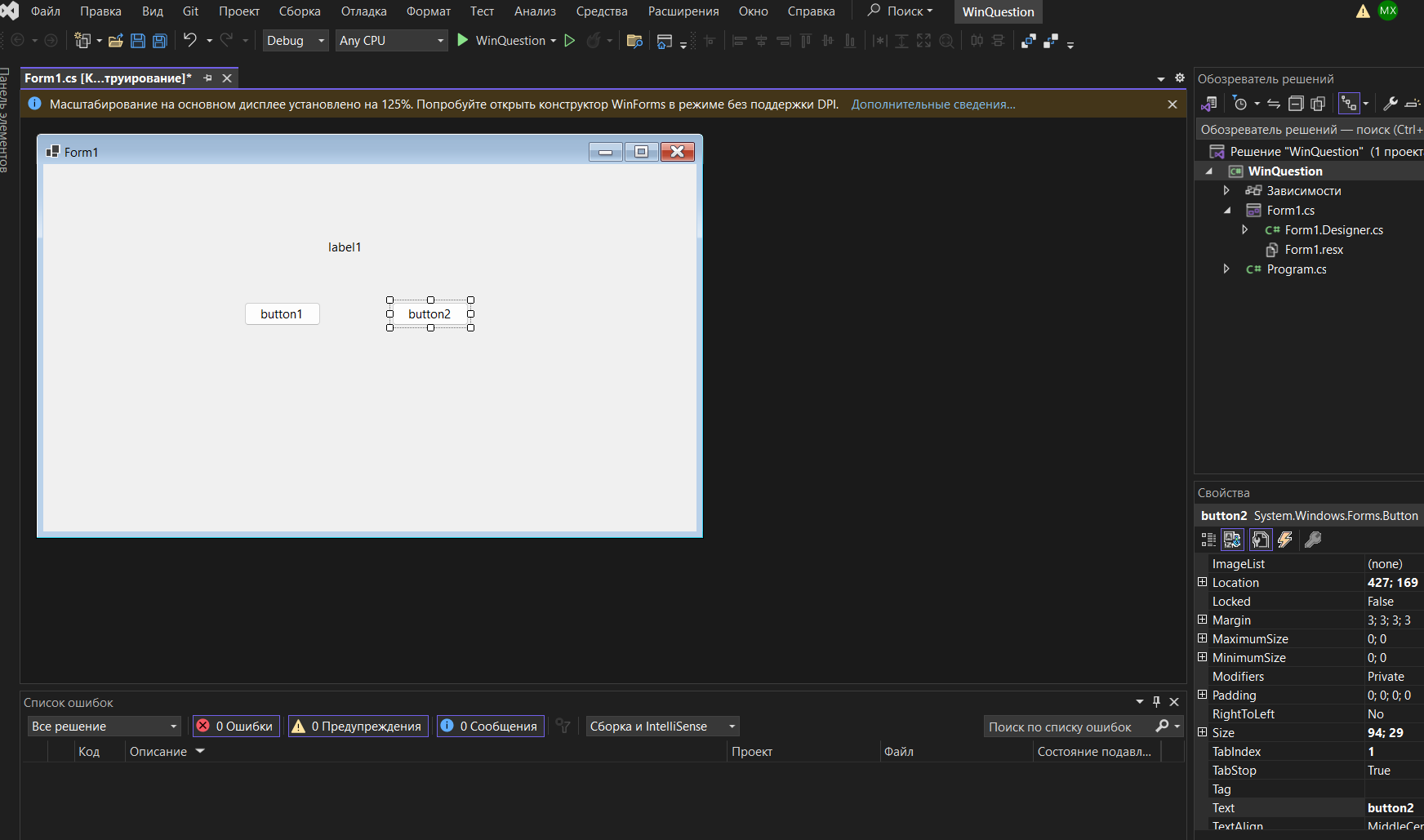
**Лабораторная работа 2. Работа с элементами управления**

Цель работы: Изучение способов использования элементов управления и получение навыков по обработке событий.

**Упражнение 1. Обработка событий Click и MouseMove**

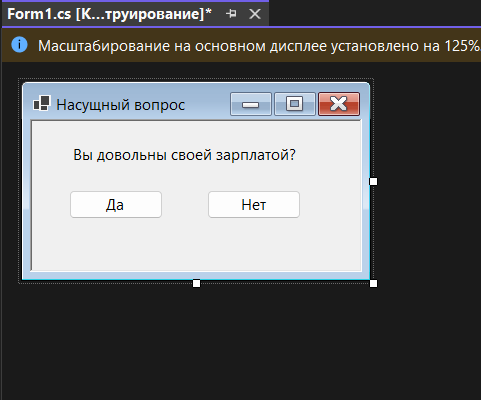
**Размещение на форме элементов управления**



1. Создайте новое Windows приложение. Назовите его WinQuestion.

2. Расположите на форме две кнопки Button и надпись Label,

разместите их по-своему усмотрению.

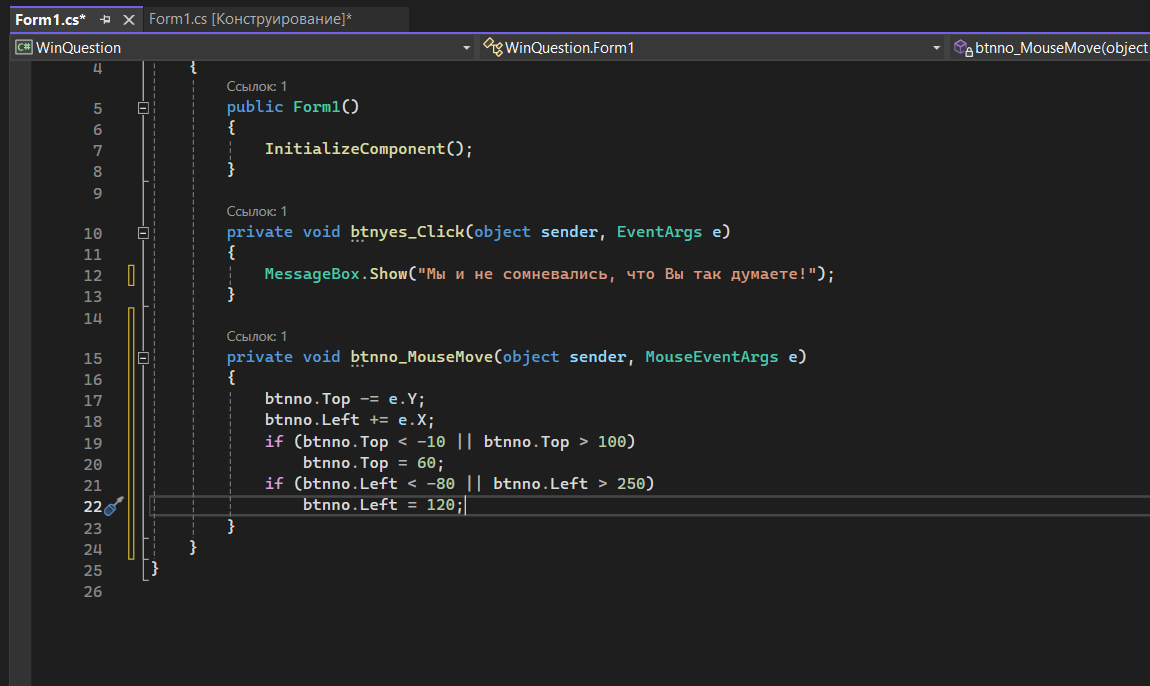


3. Установите следующие свойства элементов управления и формы:



4. Щелкните дважды по кнопке "Да". В обработчике этой кнопки

добавьте следующий код: MessageBox.Show("Мы и не сомневались, что Вы так думаете!");



5. Выделите кнопку "Нет". В окне Properties переключитесь в окно

событий и дважды щелкните в поле MouseMove.

6. В обработчике этого события добавьте код для связывания

движения мыши с координатами кнопки и указания координат, куда

кнопка будет перемещаться:

btnno.Top -= e.Y;

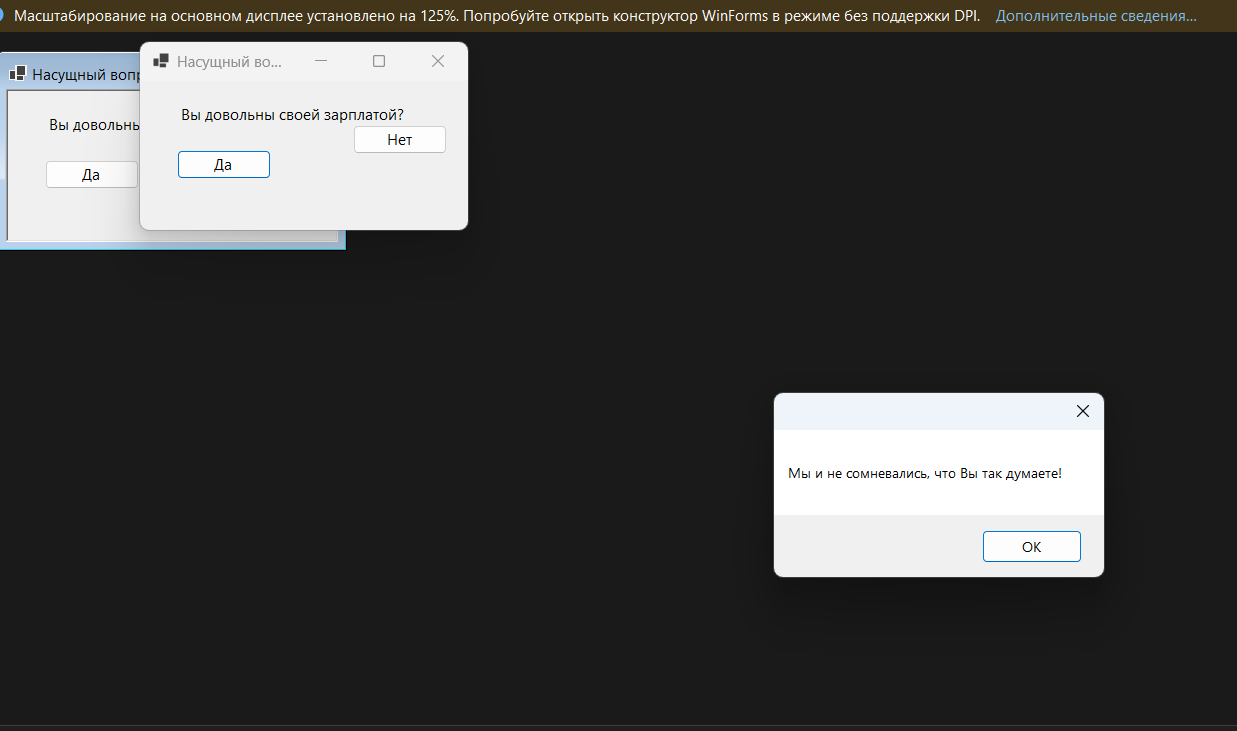
btnno.Left += e.X;

if (btnno.Top < -10 || btnno.Top > 100)

btnno.Top = 60;

if (btnno.Left < -80 || btnno.Left > 250)

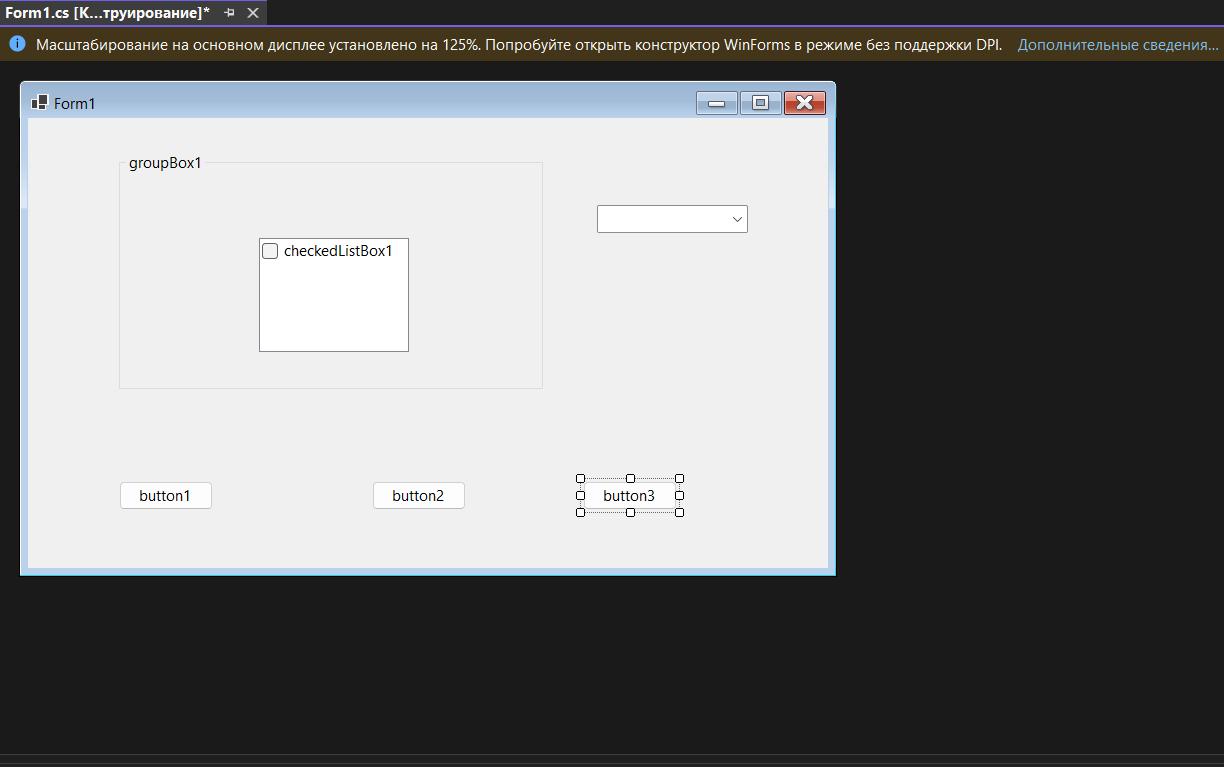
btnno.Left = 120;



7. Запустите приложение и нажмите на каждую из кнопок.

**Упражнение 2. Работа со списками**

**Создание приложения, использующее список**



1. Создайте новый проект Windows Forms, укажите имя TestList.

2. Добавьте на форму следующие элементы управления:

a. GroupBox,

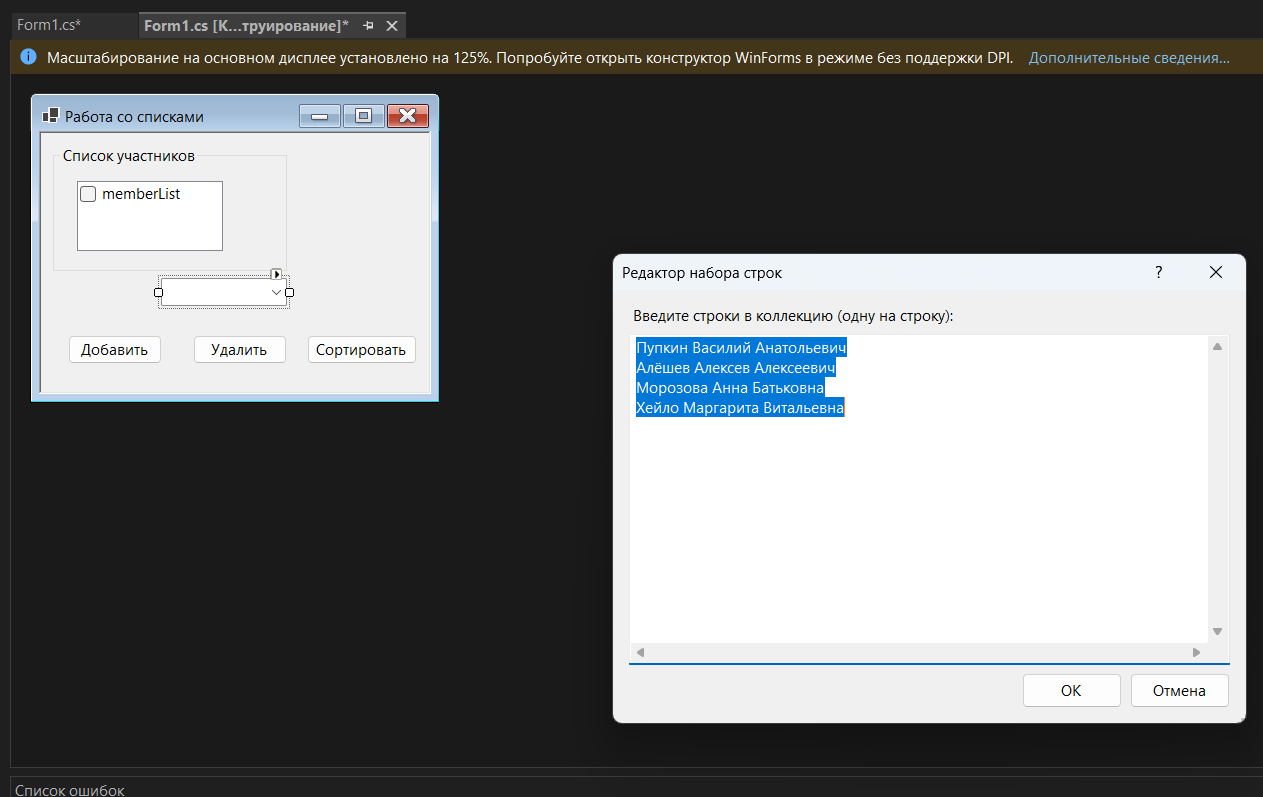
b. CheckedListBox (поместите в GroupBox)

c. ComboBox

d. три элемента Button.

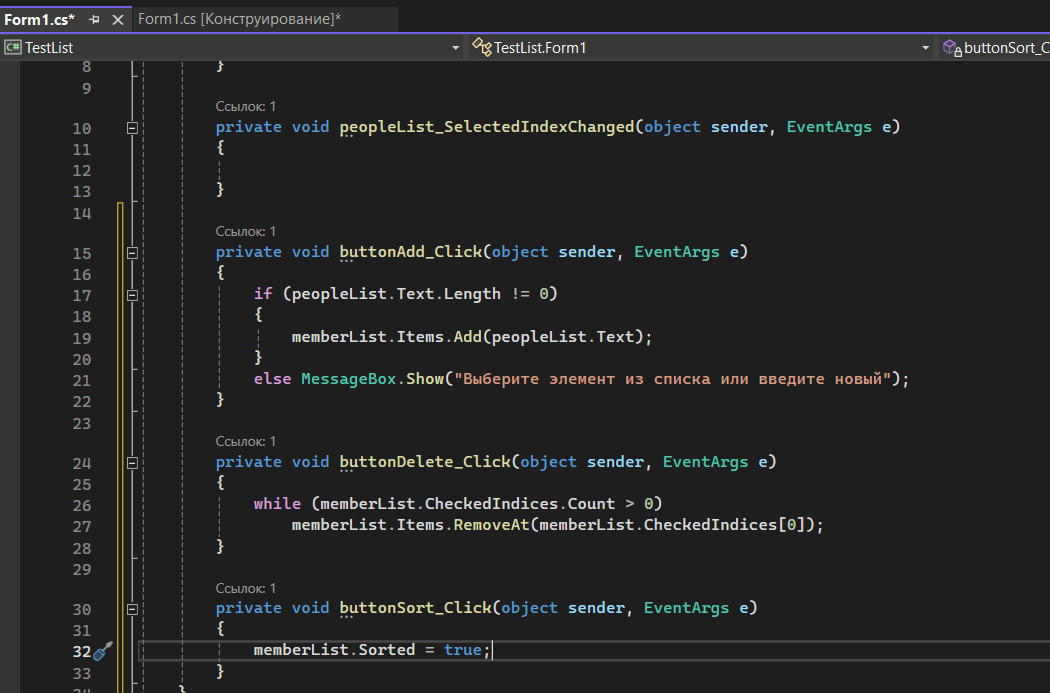


3. Установите следующие свойства формы и элементов управления:



4. Проинициализируйте элемент управления ComboBox списком

предполагаемых участников. Для этого в окне свойств peopleList выберите свойство Items. Откройте окно String Collection Editor, нажав на кнопку с тремя точками в поле Items. Добавьте в окно Ф.И.О. нескольких участников. Нажмите OK.



5. Добавьте обработчики для кнопок Добавить и Удалить, два раза

щелкнув левой кнопкой мыши по каждой из кнопок.

6. В тело обработчика события кнопки Добавить вставьте

следующий код:

if (peopleList.Text.Length != 0)

{

memberList.Items.Add(peopleList.Text);

}

else MessageBox.Show ("Выберите элемент из списка или

введите новый");

7. Для реализации удаления элементов из списка введите код в тело

обработчика события кнопки Удалить:

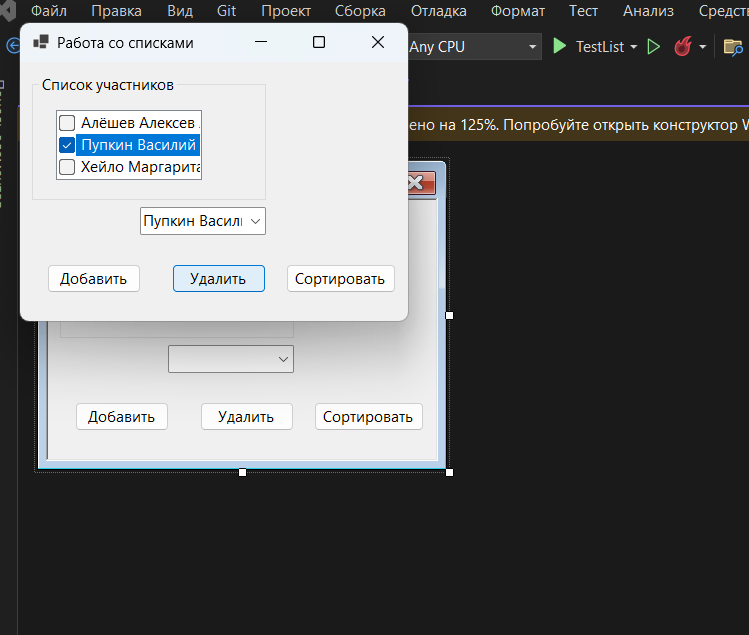
while (memberList.CheckedIndices.Count > 0)

memberList.Items.RemoveAt(memberList.CheckedIndices[0]);

8. Для реализации сортировки элементов введите код в тело

обработчика события кнопки Сортировать:

memberList.Sorted = true;

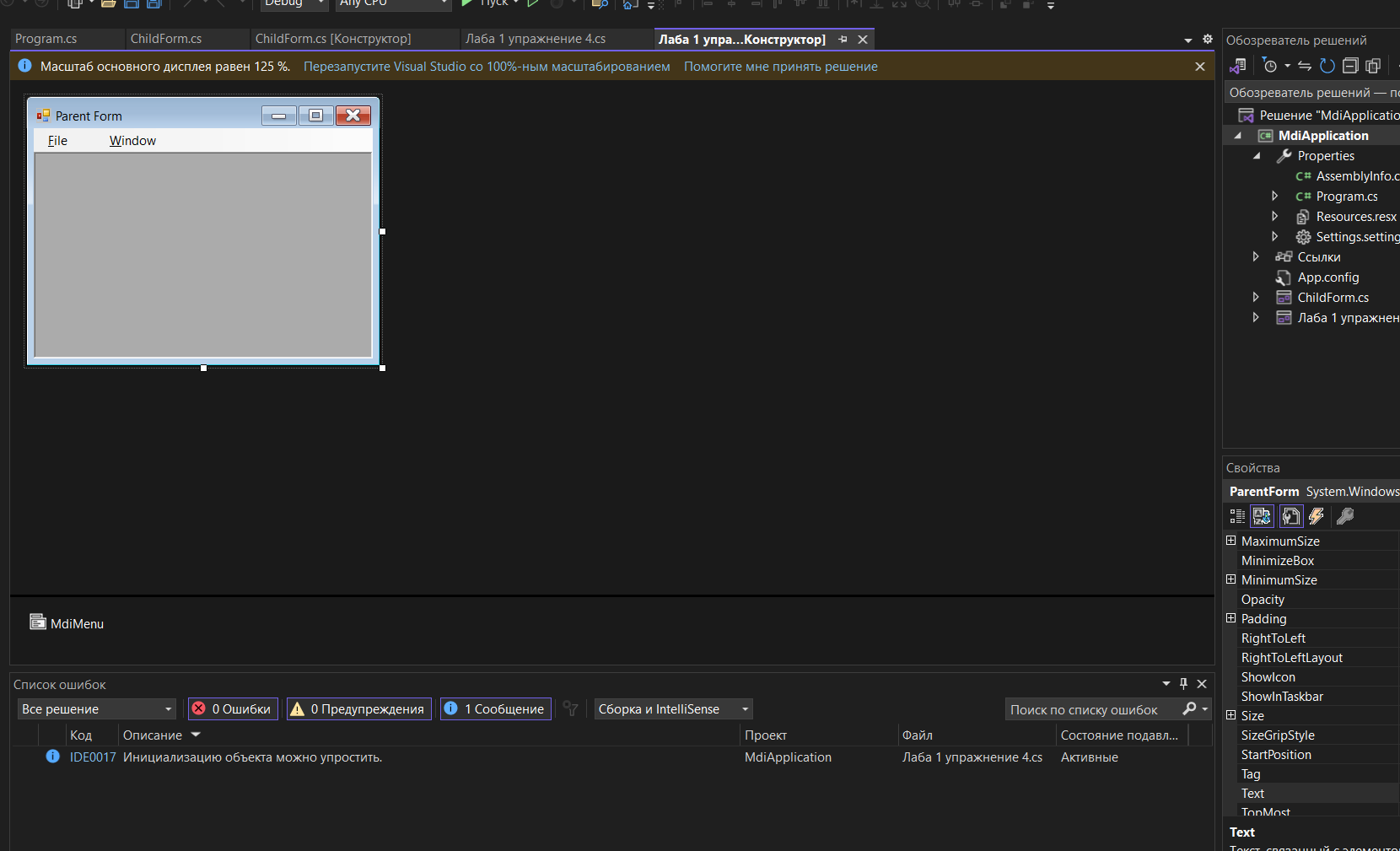


9. Откомпилируйте и запустите приложение. Заполните список

участников, выбирая их из элемента ComboBox. Запишите новые данные в этот элемент и добавьте их в список. Отсортируйте список участников.

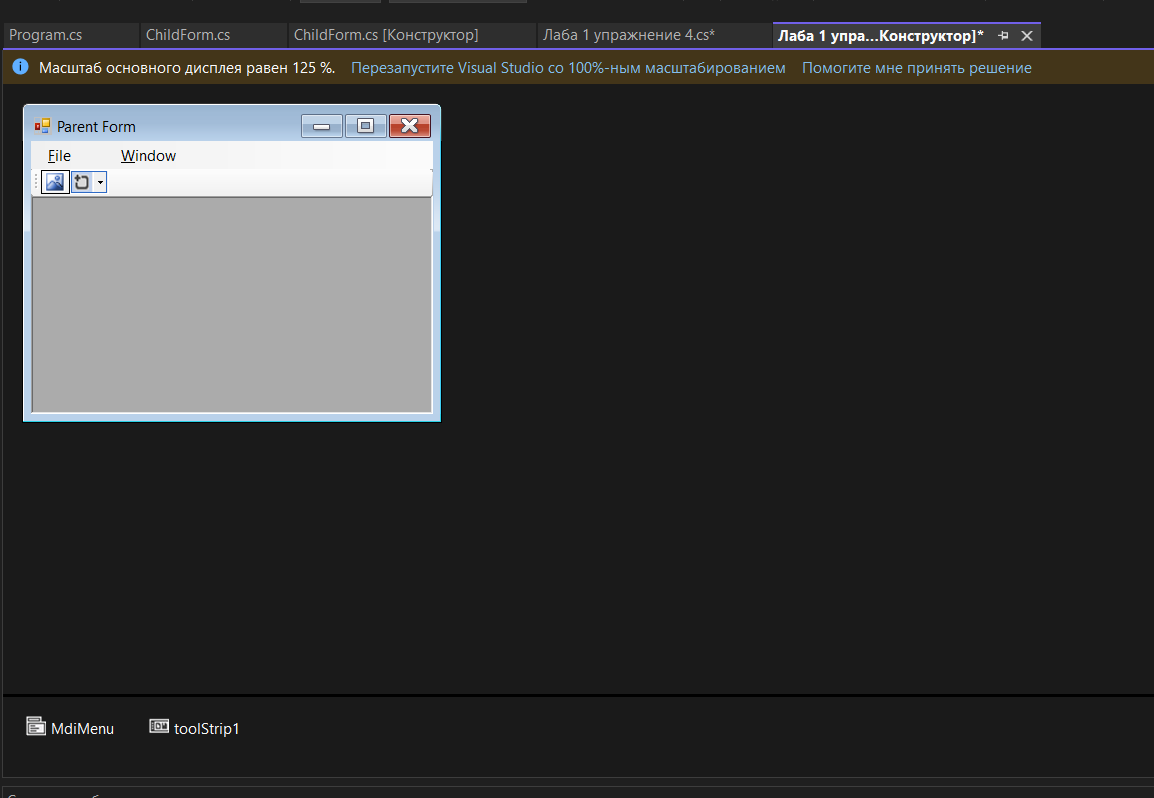
**Упражнение 3. Создание и использование элемента управления ToolStrip**

**Добавление на форму шаблона панели инструментов**



1. Откройте проект MdiApplication.

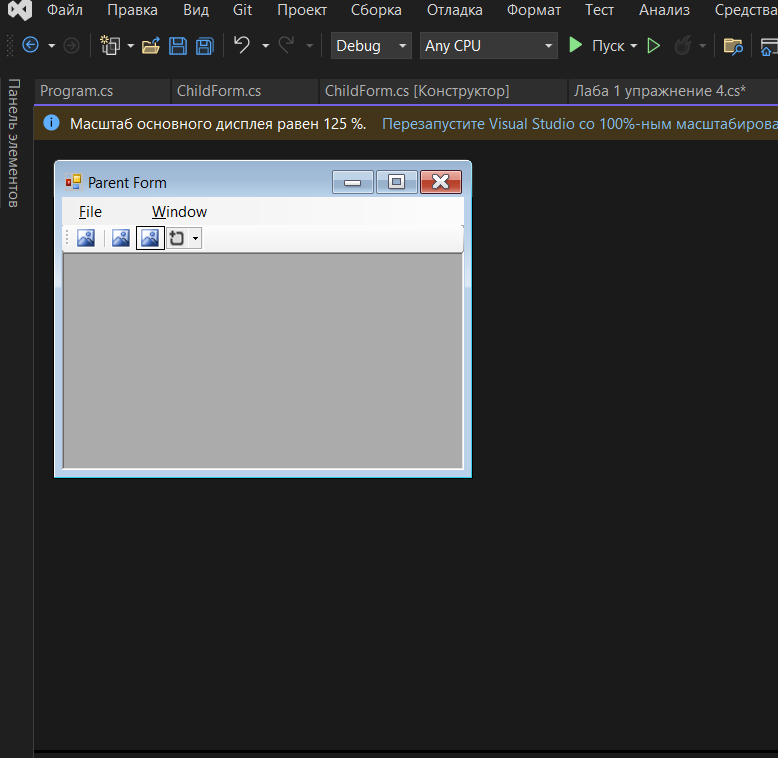
2. Откройте форму ParentForm в режиме конструктора.



3. Добавьте на форму ЭУ ToolStrip.

4. На форме откройте выпадающий список ЭУ ToolStrip и выберите

button – добавится элемент toolStripButton1. В панели инструментов он представлен в виде кнопки с рисунком, обозначающим функцию, которую этот элемент содержит.



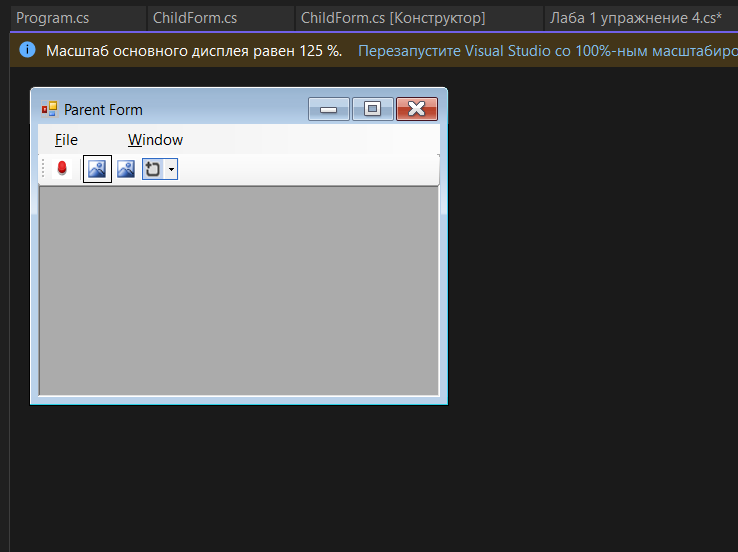
5. Снова откройте выпадающий список ЭУ ToolStrip и выберите

Separator – добавится элемент, который отделяет одни элементы панели инструментов от других.

6. Справа от разделителя добавьте еще две кнопки – элементы toolStripButton2 и toolStripButton3.

7. В итоге вы должны получить три кнопки, отделенные одним разделителем.

**Отображение рисунка на элементах панели управления**



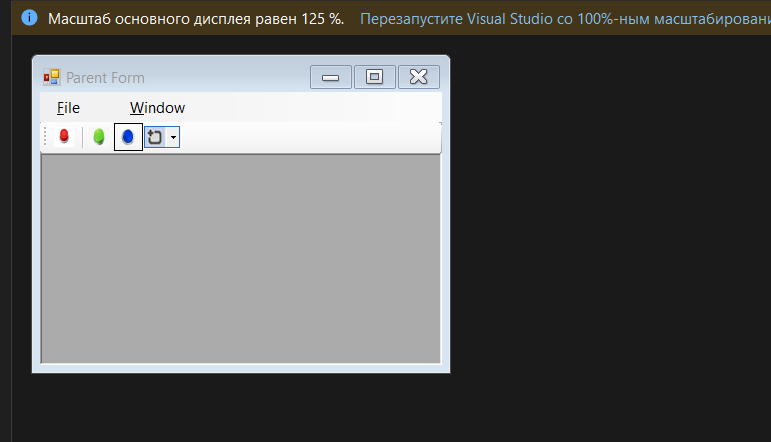
8. Выберите первую кнопку. Убедитесь, что в окне Properties

свойству DisplayStyle задано значение Image.

9. В окне Properties выберите изображение элемента управления,

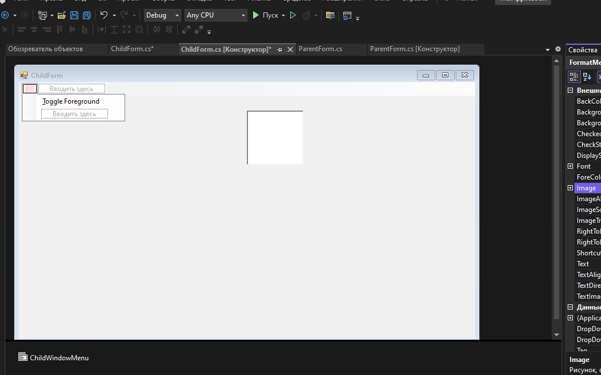
щелкнув свойство Image и выбрав соответствующее изображение или путь к нему в диалоговом окне Select Resource. Если у Вас есть готовые файлы подходящих изображений, то выберите их, в противном случае укажите любой рисунок из папки Мои рисунки. Набор готовых изображений можно найти в графических файлах в папке \Microsoft Visual Studio

9.0\Common7\VS2008ImageLibrary\ .



10. Повторите предыдущие два пункта для остальных кнопок.

**Создание графических изображений кнопок панели инструментов**



11. Если Вас не устраивает вид готовых рисунков, то можно для

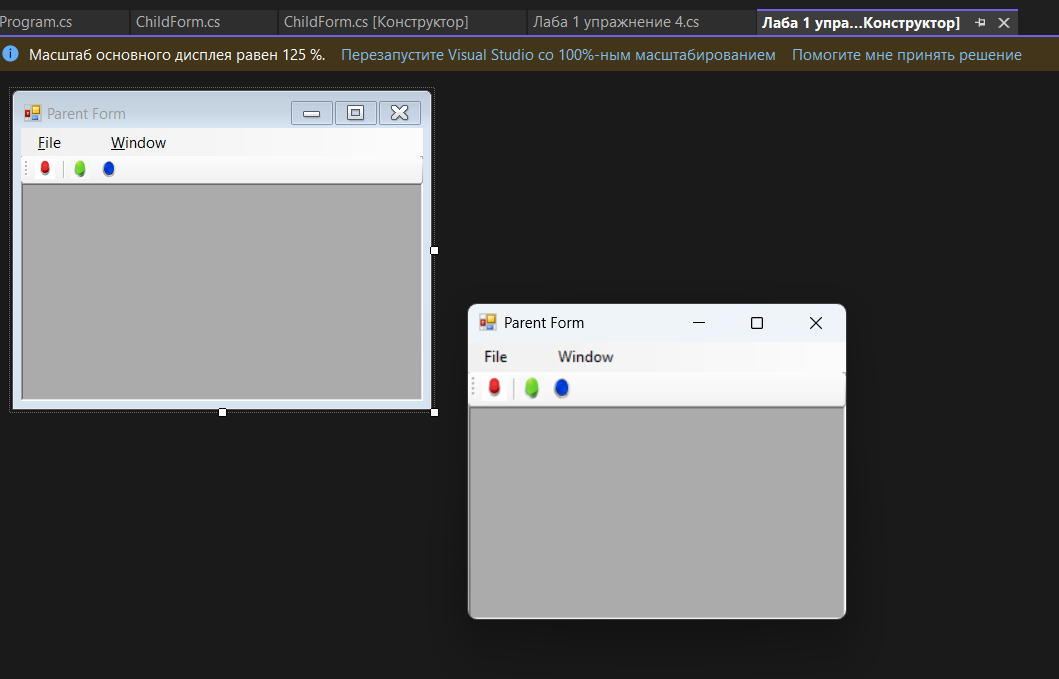
придания кнопкам графических изображений самостоятельно их создать. Для работы с изображениями в среде разработки существует Image Editor. Он позволяет создать изображения с использованием простейших инструментов.

12. Для создания файла с изображением выберите меню

File | New | File... В появившемся окне New File выберите тип файла “Файл точечного рисунка”, нажмите Open. Появится пустое изображение с дополнительной панелью управления. В основном меню появится новый пункт меню Image. Для отображения панели с палитрой компонент выберите в меню пункт Image | Show Colors Window (Показать окно выбора цвета).

13. Создайте по своему усмотрению изображение для кнопки New,

например, в виде белого листа и сохраните изображение в файл с именем Icon New.bmp в каталог с решением.



14. Повторите действия для создания иконок для других кнопок,

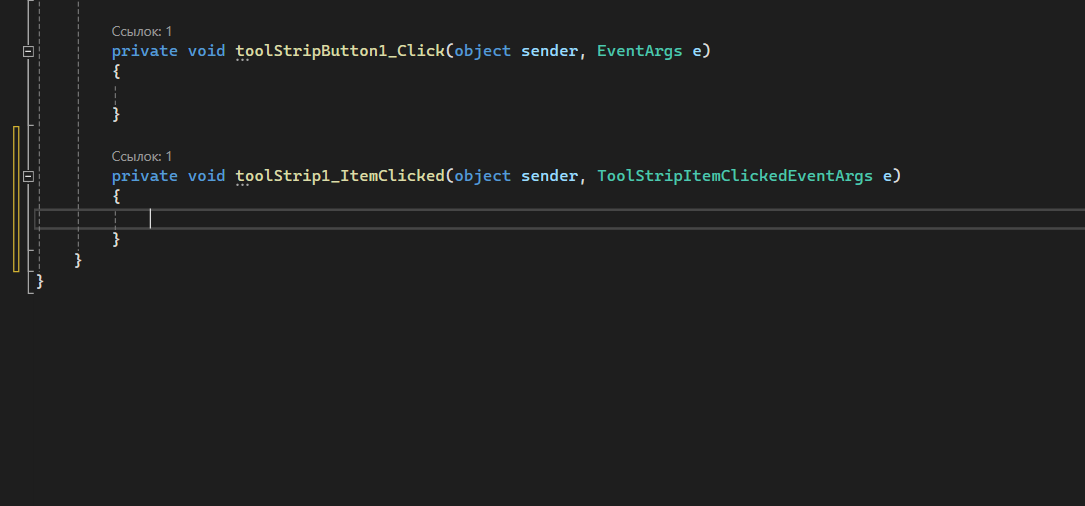
например, в виде нарисованных распылителем букв C и T. Сохраните изображения в каталог с решением в файлы с именами

Icon\_Windows\_Cascade.bmp и Icon\_Windows\_Title.bmp соответственно.

15. Выполните действия для указания новых изображений этим кнопкам.

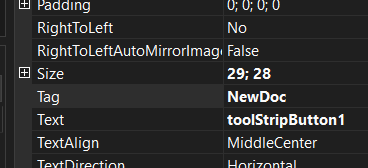
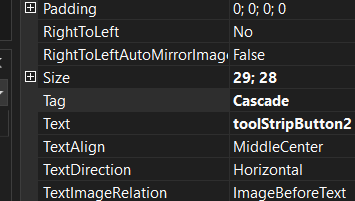
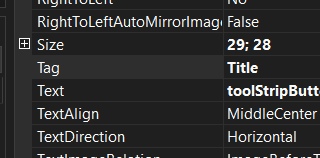
16. Сохраните и запустите проект. Проверьте вид и работоспособность кнопок.

**Добавление обработчиков событий для кнопок**



17. Добавьте обработчик события Click объекта toolStrip1, щелкнув

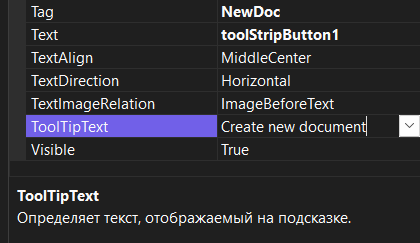
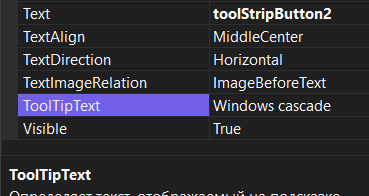
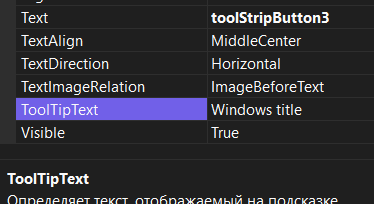
два раза указателем мыши по имени события Click на закладке событий в окне свойств. В программу добавится функция toolStrip1\_ItemClicked как обработчик события, происходящего при нажатии кнопки на панели инструментов.

18. В окне Properties для toolStripButton1 в свойстве Tag запишите

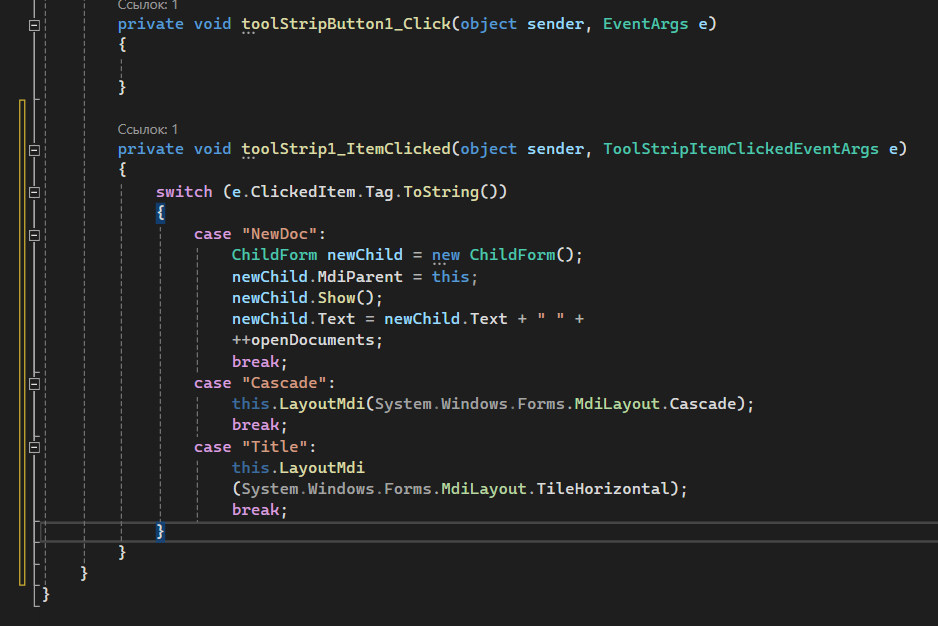
NewDoc. Аналогично укажите для toolStripButton2 и toolStripButton3 для

свойства Tag значения Cascade и Title соответственно.

19. Укажите для кнопок всплывающие подсказки в свойстве

ToolTipText: Create new document, Windows cascade и Windows title.



20. В обработчике события Click объекта toolStrip1\_ItemClicked

добавьте код, который будет реализовывать различные действия в

зависимости от нажимаемой кнопки:

switch(e.ClickedItem.Tag.ToString())

{

case "NewDoc":

ChildForm newChild = new ChildForm();

newChild.MdiParent = this;

newChild.Show();

newChild.Text = newChild.Text+" "+

++openDocuments;

break;

case "Cascade":

this.LayoutMdi (System.Windows.Forms.MdiLayout.Cascade);

break;

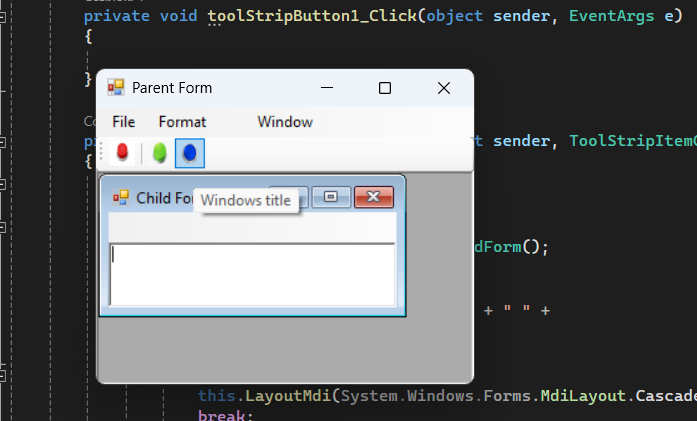
case "Title":

this.LayoutMdi

(System.Windows.Forms.MdiLayout.TileHorizontal);

break;

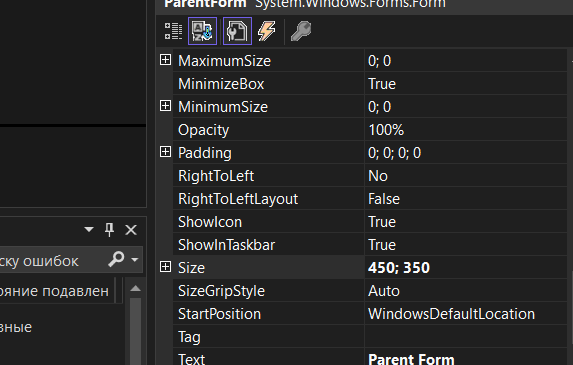
}



21. Откомпилируйте и запустите приложение. Проверьте

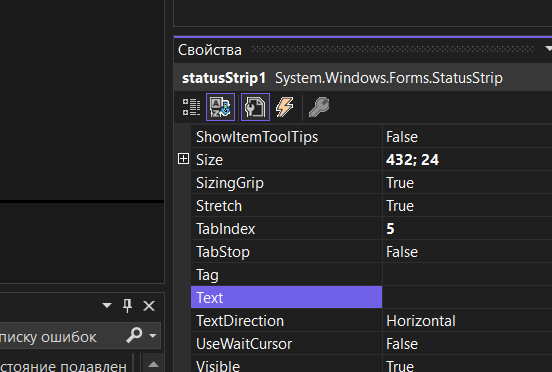
работоспособность кнопок.

**Упражнение 4. Использование элемента управления StatusStrip**



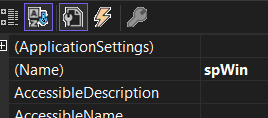
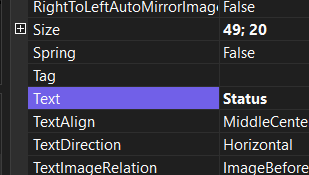
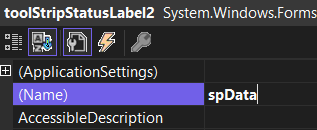
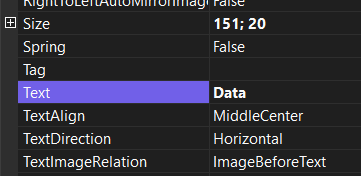
1. Откройте проект MdiApplication.

2. Увеличьте размер формы ParentForm до значения (450;350).



3. Добавьте на форму ParentForm элемент управления StatusStrip.

4. Удалите содержимое поля свойства Text.

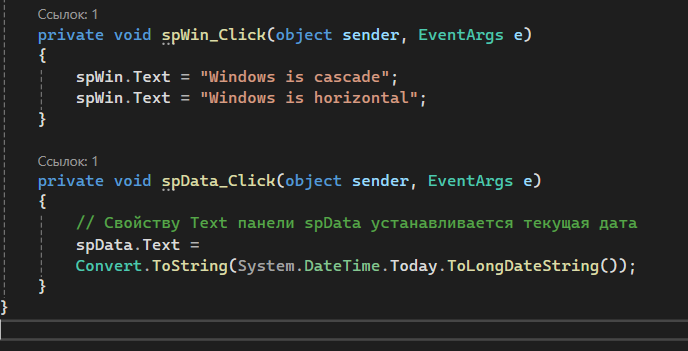
5. Щелкните на кнопку выпадающего списка панели и выберите

StatusLabel. Добавится элемент toolStripStatusLabel1 – первая панель для

отображения.

6. Создайте еще одну панель аналогичным способом –

toolStripStatusLabel2 и установите им следующие свойства:



7. Для отображения информации на первой панели вставьте в

соответствующие обработчики команд меню и кнопок на панели

инструментов следующую строку кода:

a. Для каскадной ориентации:

spWin.Text="Windows is cascade";

b. Для горизонтальной ориентации:

spWin.Text="Windows is horizontal";

8. Для отображения даты на второй панели в конструкторе формы

ParentForm добавьте код:

public ParentForm()

{

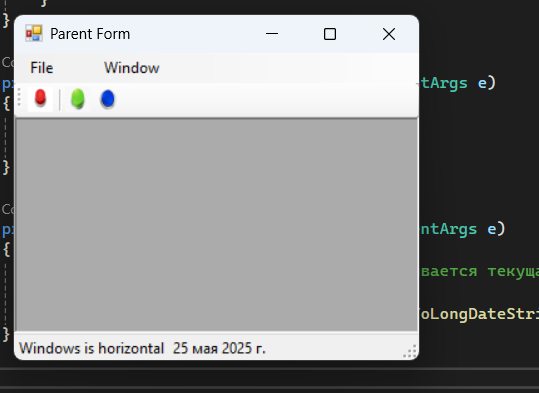
InitializeComponent();

// Свойству Text панели spData устанавливается текущая дата

spData.Text =

Convert.ToString(System.DateTime.Today.ToLongDateString());

}

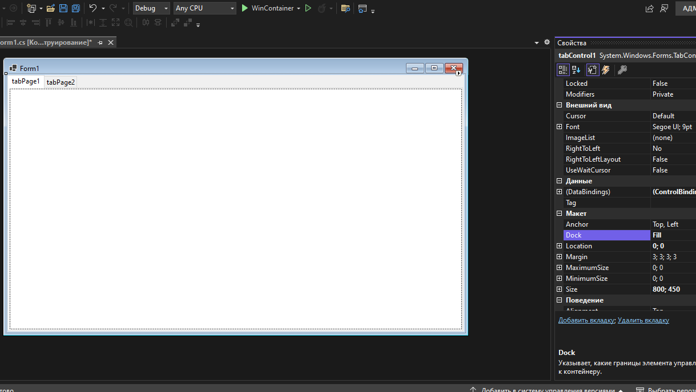


9. Откомпилируйте и запустите приложение. Проверьте

работоспособность панели состояния.

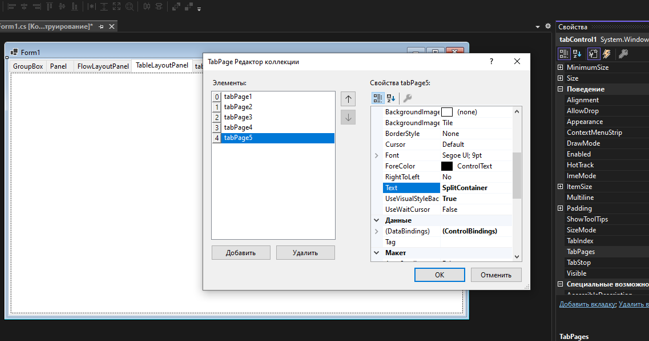
**Упражнение 5. Работа с контейнерными элементами управления**

**Создание проекта с возможностью группировки элементов на вкладках**



1. Откройте Visual Studio и создайте новый проект Windows Forms. Назовите его WinContainer.

2. Перетащите из Toolbox в форму элемент управления TabControl. В окне Properties задайте свойству Dock значение Fill.



3. В окне Properties выберите свойство TabPages, чтобы открыть

TabPage Collection Editor. Добавьте вкладки так, чтобы их стало всего пять. Задайте свойствам Text этих пяти элементов управления TabPage значения GroupBox, Panel, FlowLayoutPanel, TableLayoutPanel и SplitContainer. Щелкните ОК.

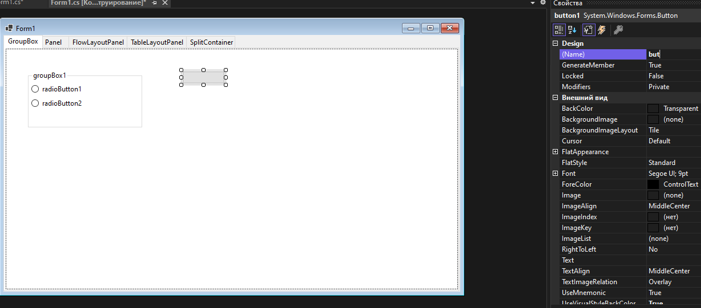
**Настройка контейнерного элемента GroupBox**



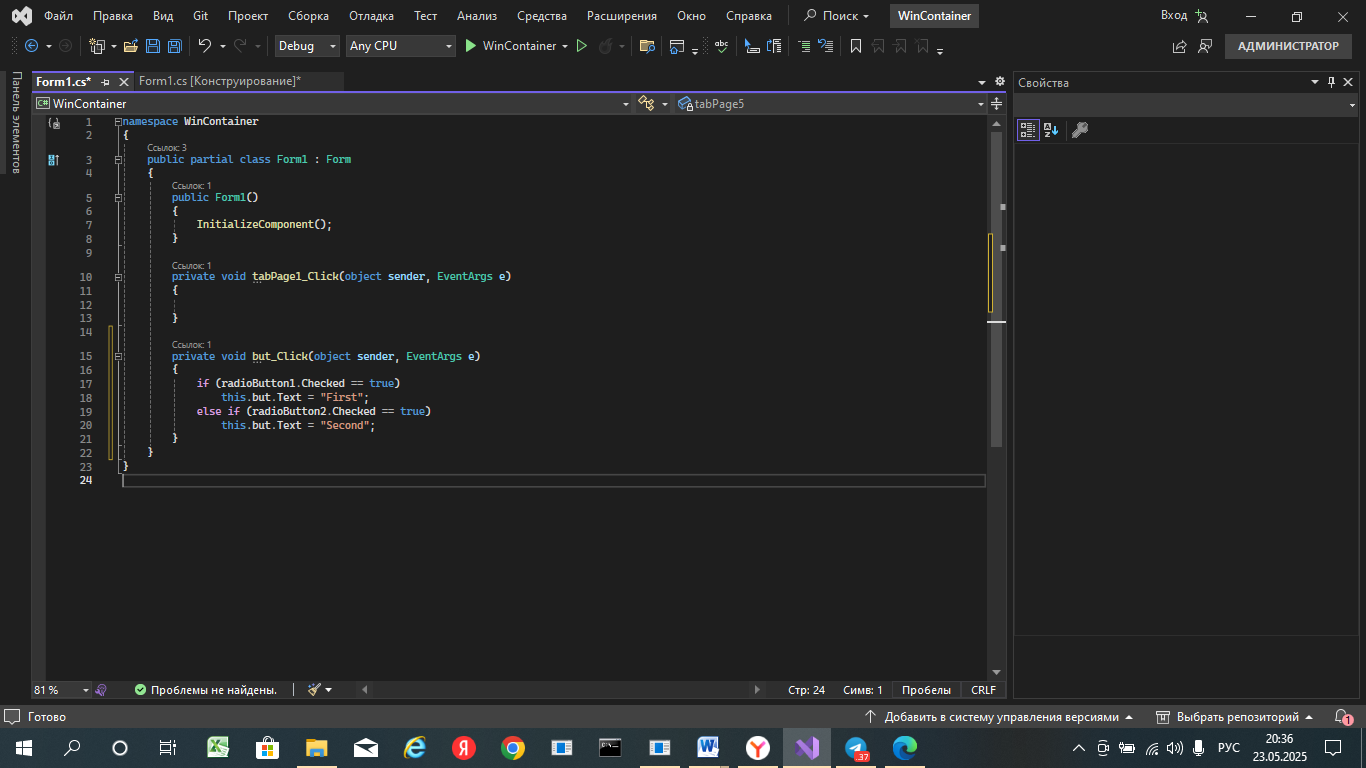
4. В форме выберите вкладку GroupBox. Перетащите элемент

управления GroupBox из Toolbox в элемент управления TabPage.

5. Перетащите в GroupBox два элемента управления RadioButton.



6. Добавьте на вкладку GroupBox вне элемента управления GroupBox кнопку (элемент управления Button). Для кнопки свойство Text сделайте пустым, а свойству Name укажите значение but.



7. Дважды кликните по кнопке и добавьте код обработчика события

установки надписи на кнопке в зависимости от выбранного переключателя (RadioButton):

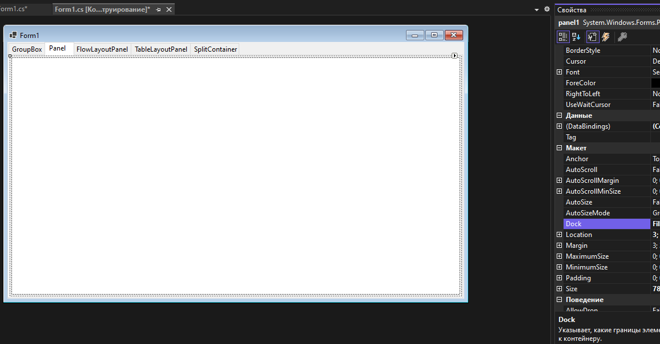
if (radioButton1.Checked == true)

this.but.Text = "First";

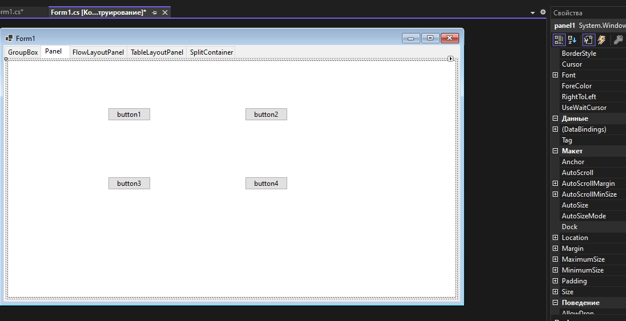
else if (radioButton2.Checked == true)

this.but.Text = "Second";

**Настройка элемента Panel**



8. Выберите в форме вкладку Panel. Перетащите элемент управления Panel из Toolbox в элемент управления TabPage. Для элемента Panel задайте свойству Dock значение Fill.

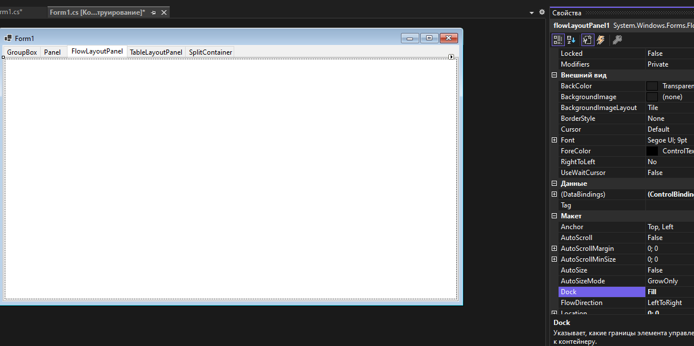


9. Перетащите четыре элемента управления Button из Toolbox в элемент управления Panel.

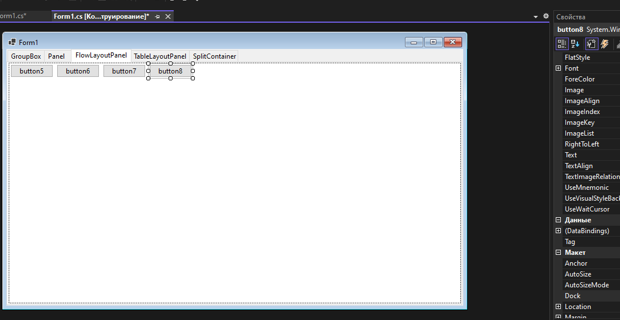
10. Свойству AutoScroll установите значение True, в этом случае

элемент управления Panel будет отображать полосы прокрутки, если элементы находятся за пределами видимых границ.

**Настройка элемента FlowLayoutPanel**



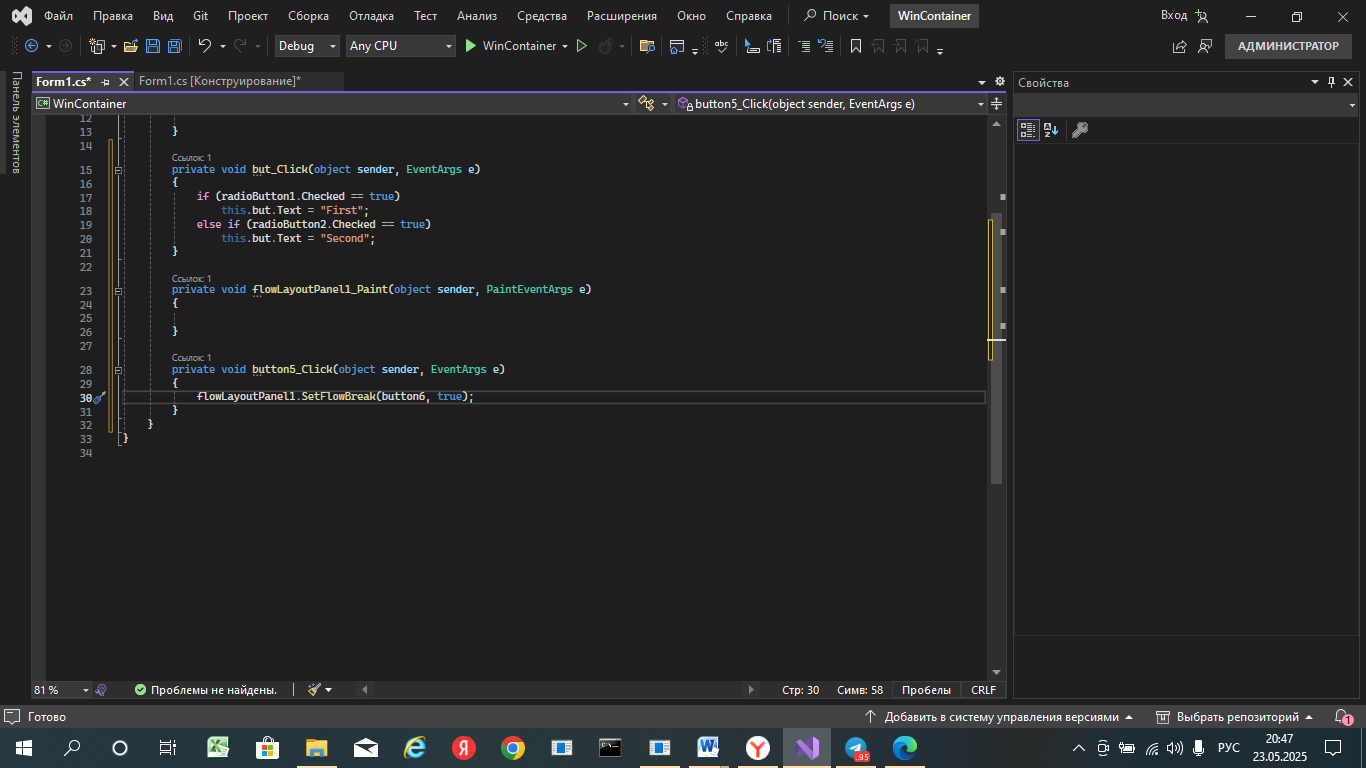
11. Выберите в форме вкладку FlowLayoutPanel. Перетащите элемент управления FlowLayoutPanel из Toolbox в элемент управления TabPage. Задайте значение Fill свойству Dock элемента управления FlowLayoutPanel.



12. Перетащите четыре элемента управления Button из Toolbox в

элемент управления Panel. Обратите внимание на размещение

добавляемых элементов: по умолчанию порядок следования элементов управления в FlowLayoutPanel — слева направо. Это значит, что элементы управления, расположенные в FlowLayoutPanel, будут находиться в левом верхнем углу и размещаться вправо до тех пор, пока не достигнут края панели. Такое поведение контролируется свойством FlowDirection, которому может быть задано четыре значения заливки в FlowLayoutPanel: LeftToRight — по умолчанию, RightToLeft —справа налево, TopDown — сверху вниз и BottomUp — снизу вверх.

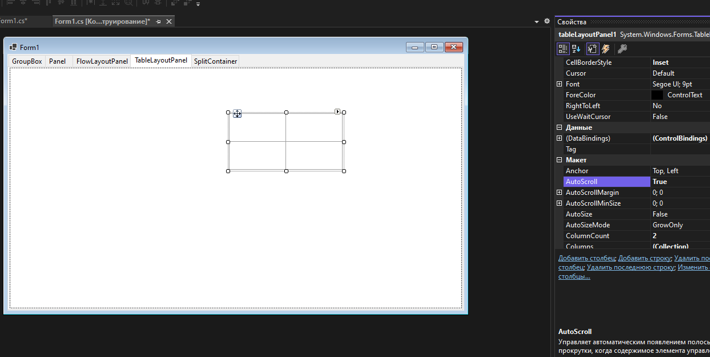


13. Дважды щелкните кнопку button5 и добавьте в обработчик

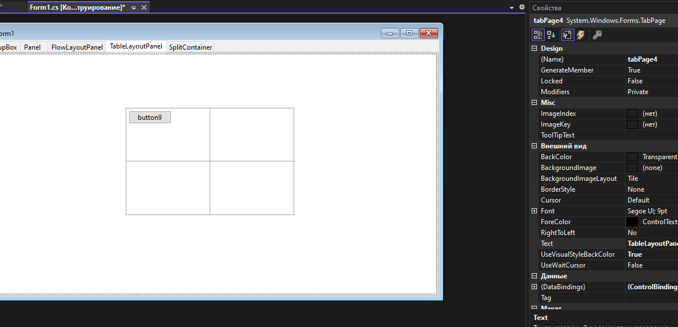
события button5\_Click следующий код:

flowLayoutPanel1.SetFlowBreak(button6, true);

**Настройка элемента TableLayoutPanel**

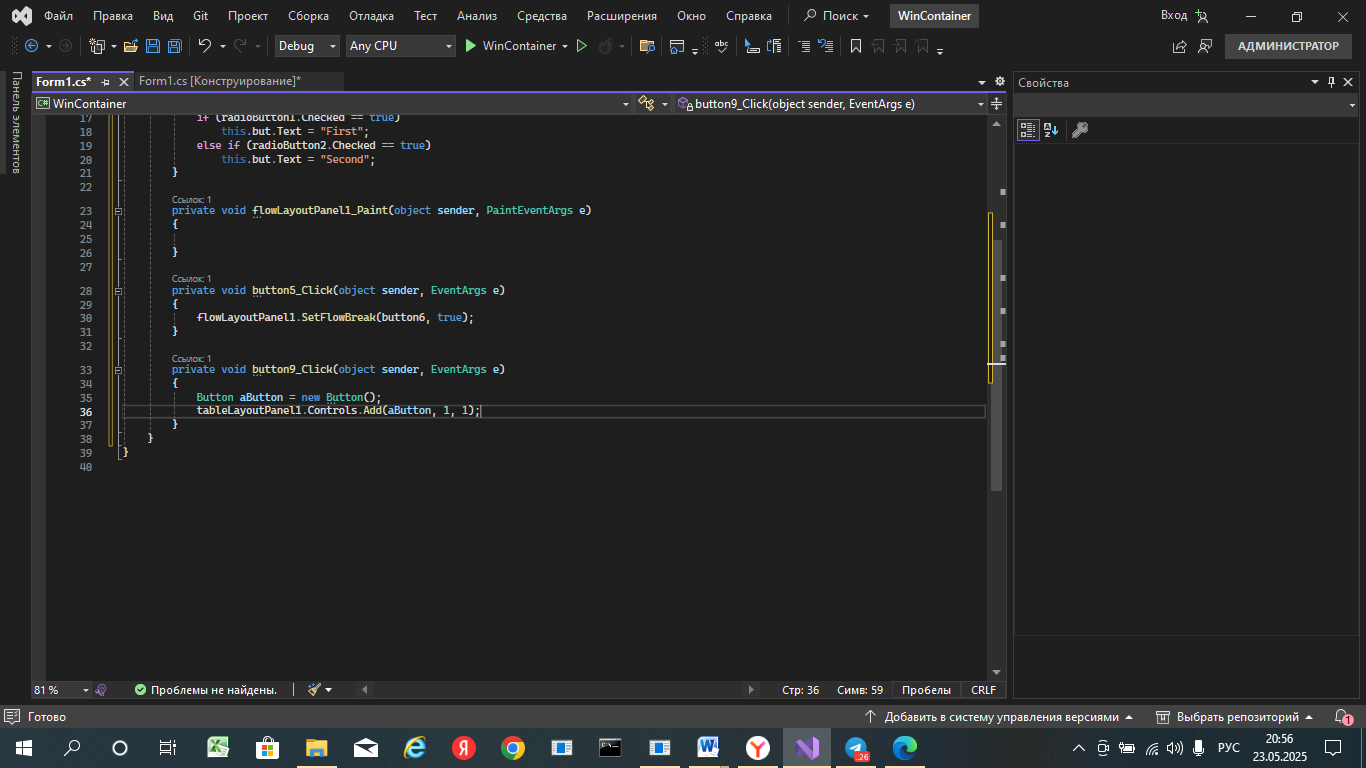


14. Выберите конструктор формы (если это необходимо). В форме выберите вкладку TableLayoutPanel. Перетащите элемент управления TableLayoutPanel из Toolbox в TabPage. Задайте свойству CellBorderStyle (определяет вид ячеек таблицы и их поведение) значение Inset, а свойству AutoScroll — True.



15. Перетащите элемент управления Button из Toolbox в левую

верхнюю ячейку элемента управления TableLayoutPanel.



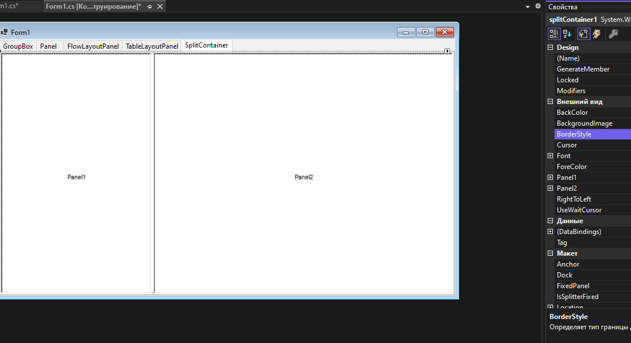
16. Дважды щелкните Button9 и добавьте в обработчик события

Button9\_Click следующий код:

Button aButton = new Button();

tableLayoutPanel1.Controls.Add(aButton, 1, 1);

**Настройка элемента SplitContainer**



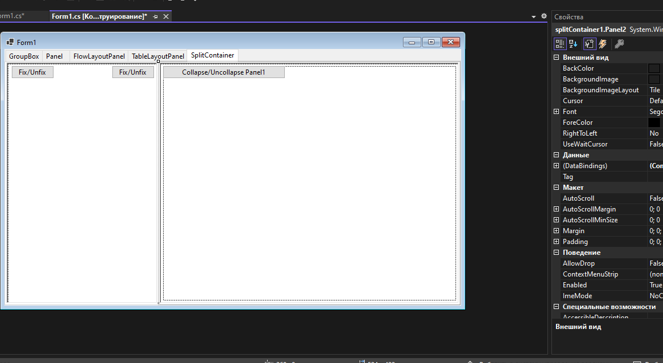
17. В конструкторе выберите вкладку SplitContainer. Перетащите

элемент управления SplitContainer из Toolbox в TabPage. Задайте свойству BorderStyle значение Fixed3D.



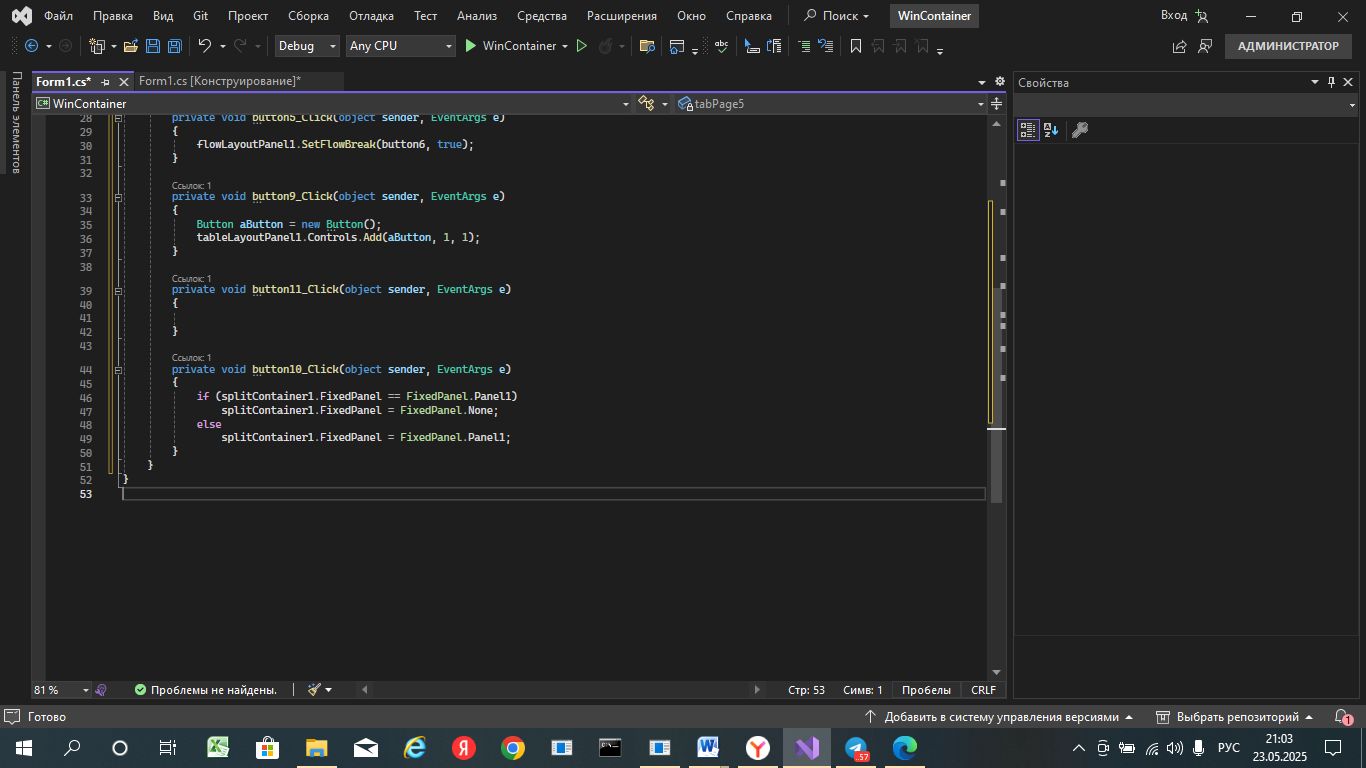
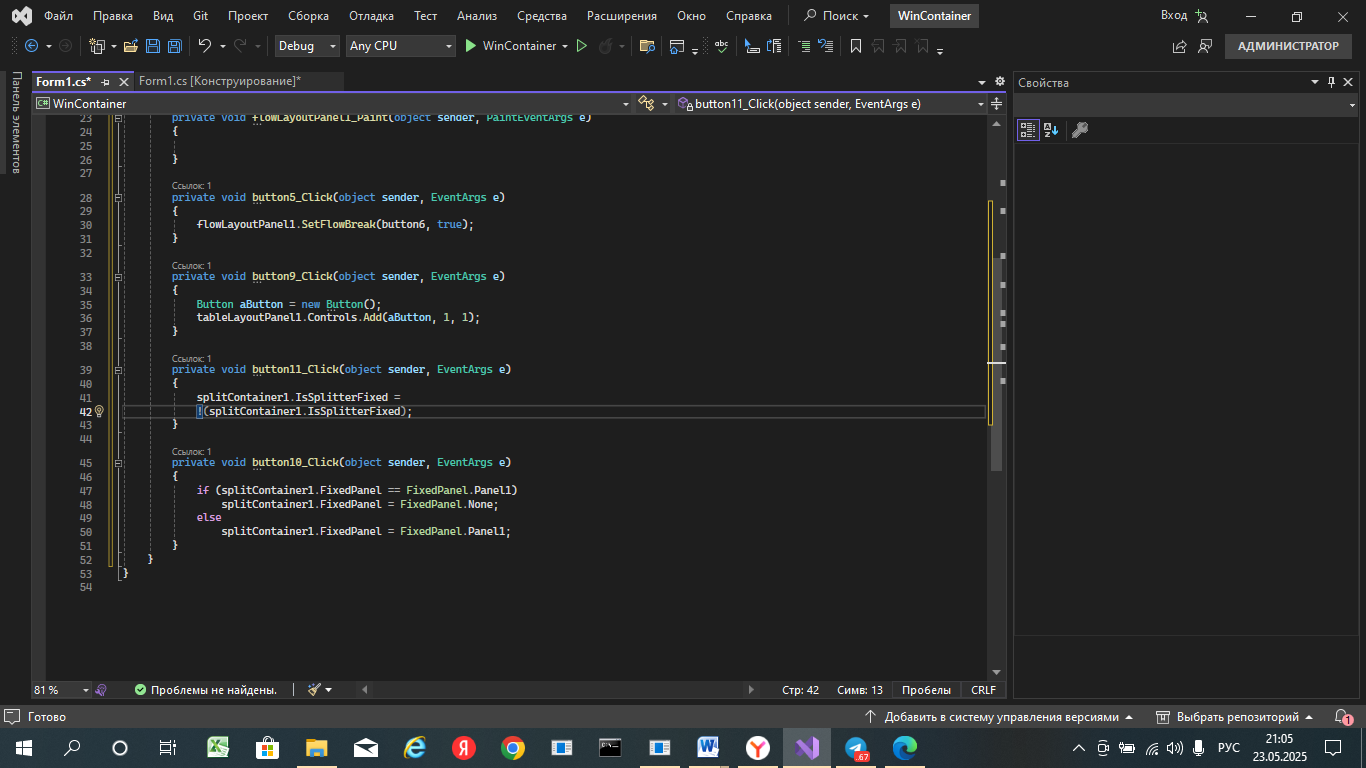
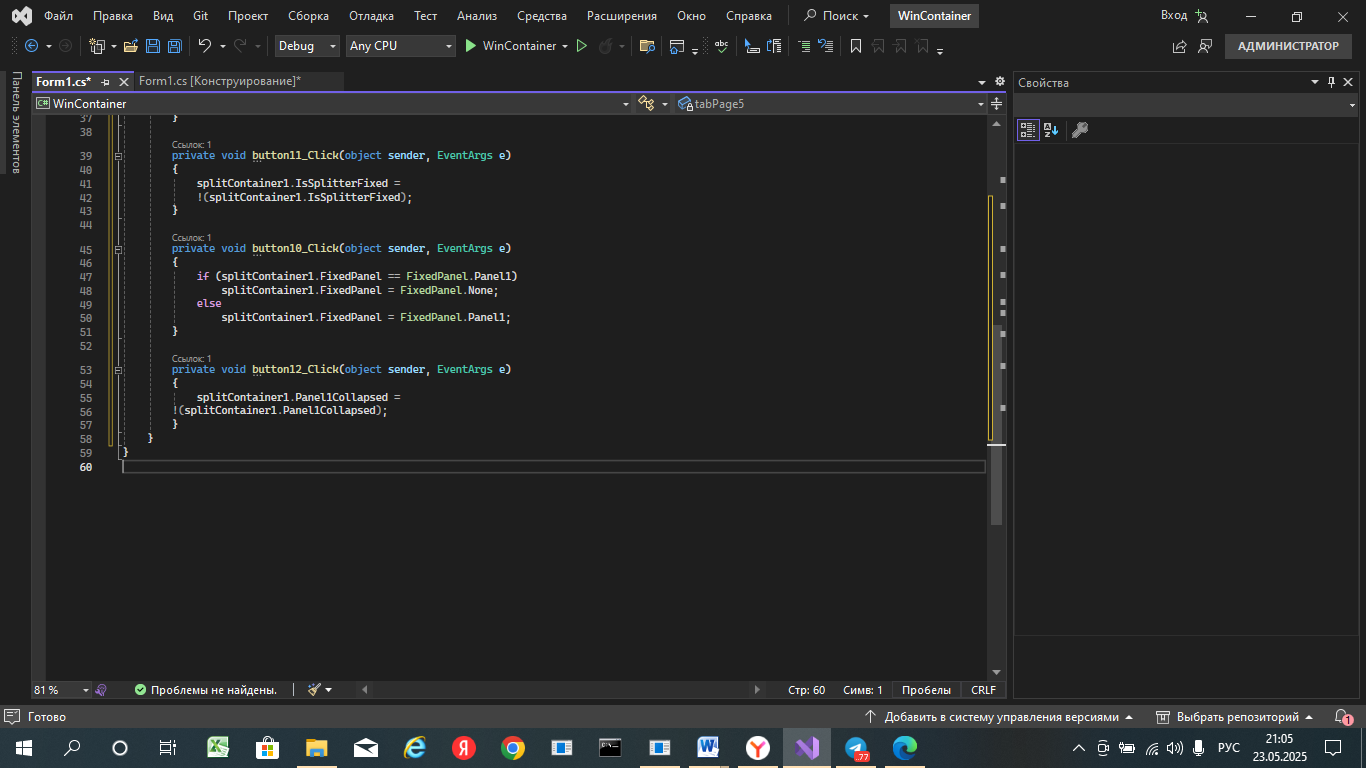
18. Перетащите два элемента управления Button из Toolbox в Рапеl1

элемента управления SplitContainer. Задайте свойствам Text этих кнопок значения Fix/Unfix Рапеl1 и Fix/Unfix Splitter. Измените размеры кнопок так, чтобы отображался текст.



19. Добавьте кнопку в Рапе12 и задайте свойству Text значение

Collapse/Uncollapse Рапеl1. Измените размеры кнопок так, чтобы отображался текст.

20. Дважды щелкните кнопку Fix/Unfix Panel1 и добавьте в

обработчик события Click следующий код:

if (splitContainer1.FixedPanel == FixedPanel.Panel1)

splitContainer1.FixedPanel = FixedPanel.None;

else

splitContainer1.FixedPanel = FixedPanel.Panel1;

21. Дважды щелкните кнопку Fix/Unfix Splitter и добавьте в обработчик события Click следующий код:

splitContainer1.IsSplitterFixed =

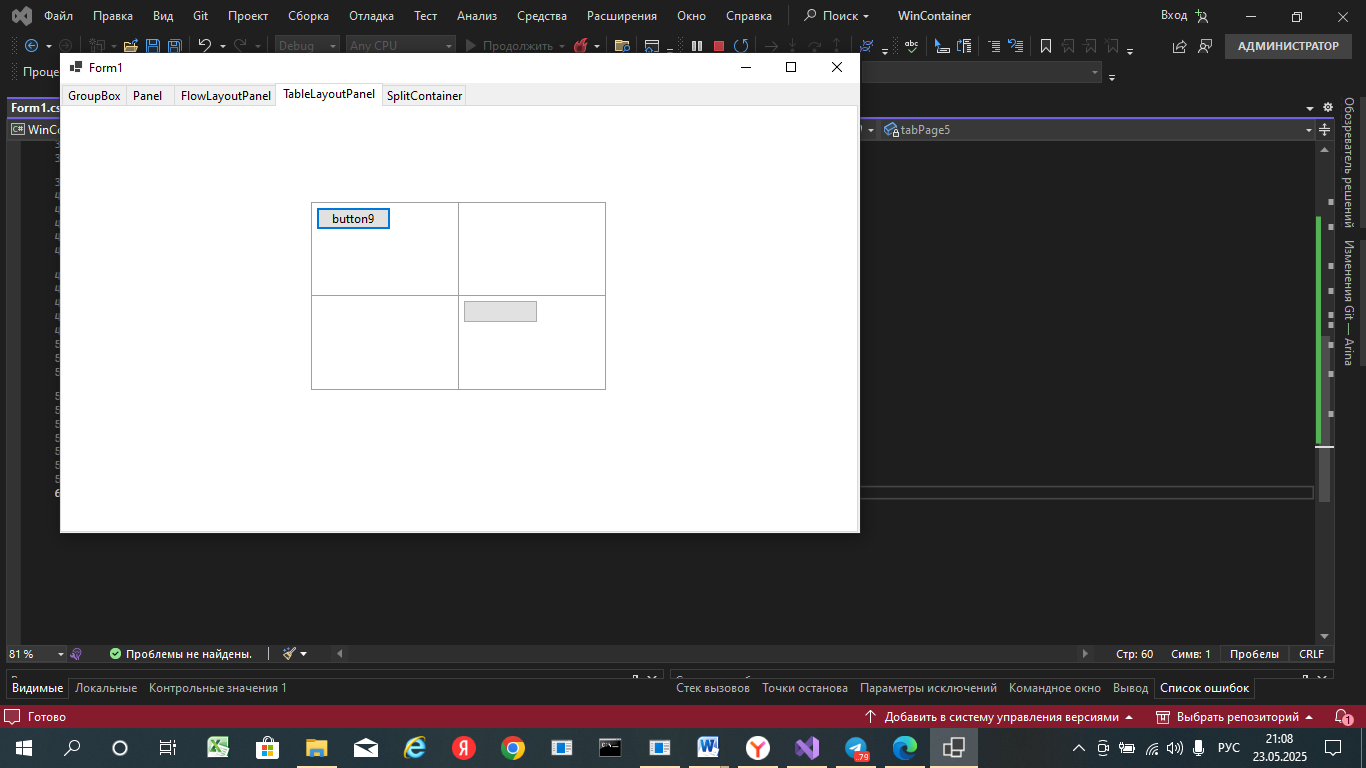
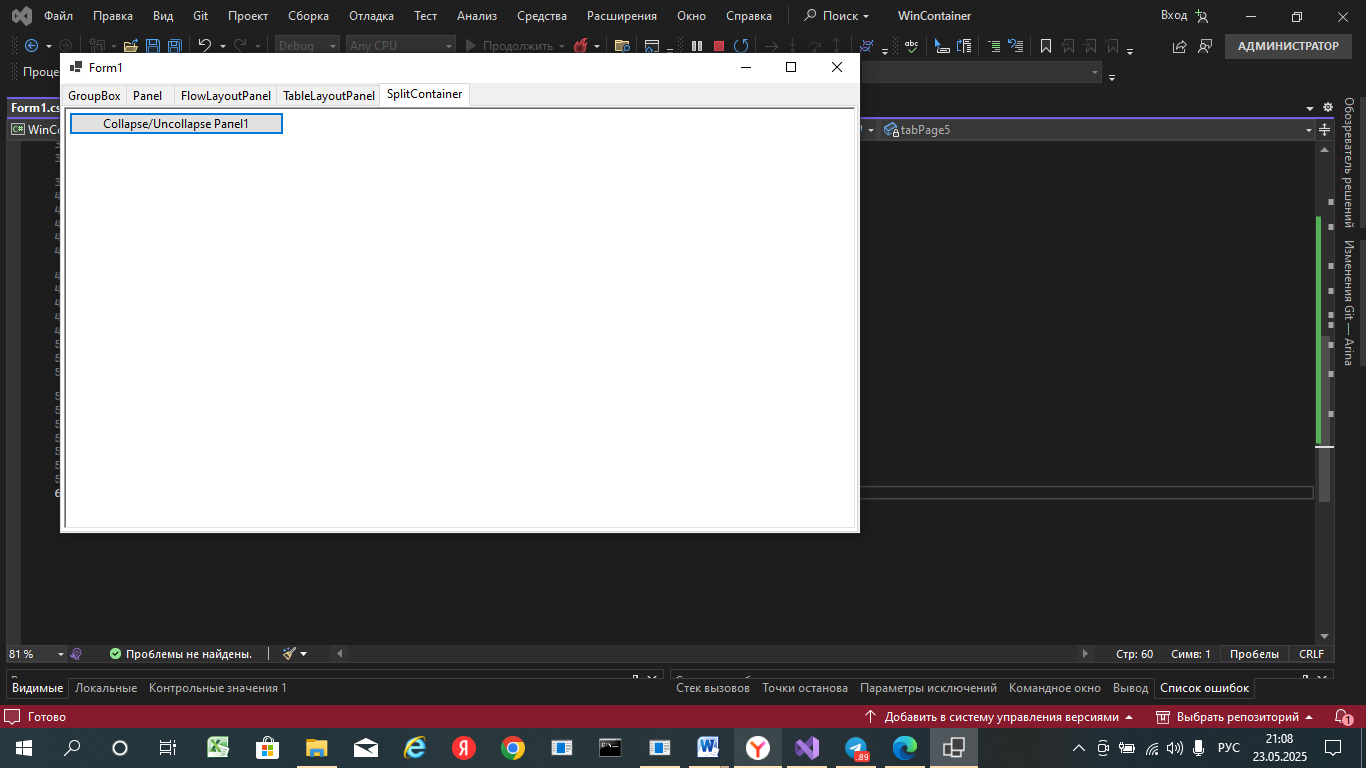
!(splitContainer1.IsSplitterFixed);

22. Дважды щелкните кнопку Collapse/Uncollapse Panell и добавьте в обработчик события Click следующий код:

splitContainer1.Panel1Collapsed =

!(splitContainer1.Panel1Collapsed);

23. Постройте и запустите приложение.

24. На вкладке GroupBox поочередно выбирайте переключатели

следите за изменением надписи на кнопке.

25. На вкладке Panel измените размер формы с помощью мыши. Проверьте, появились ли полосы прокрутки.

26. На вкладке FlowLayoutPanel измените размер формы с помощью мыши. Просмотрите, что автоматически изменилась компоновка. Щелкните кнопку button5 и проверьте, прервалась ли последовательность на элементе управления button6 (это было реализовано вызовом метода SetFlowBreak).

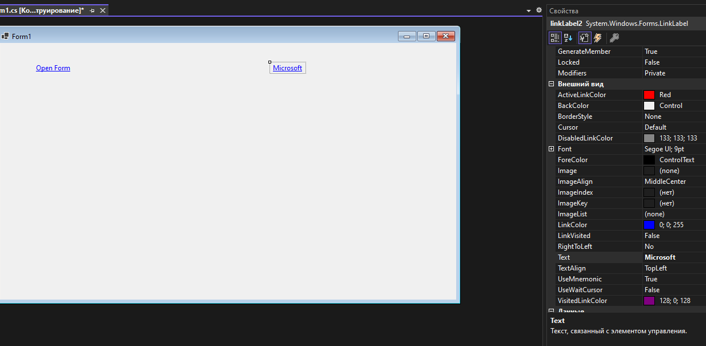
27. На вкладке TableLayoutPanel щелкните кнопку button9, добавится новая кнопка.

28. На вкладке SplitContainer измените размеры формы, а также

размеры каждой панели, передвинув Splitter. По очереди щелкайте каждую кнопку и смотрите, как это отражается на возможности элемента управления изменять свои размеры.

**Упражнение 6. Элементы с поддержкой отображения текста**

**Настройка элемента управления LinkLabel**



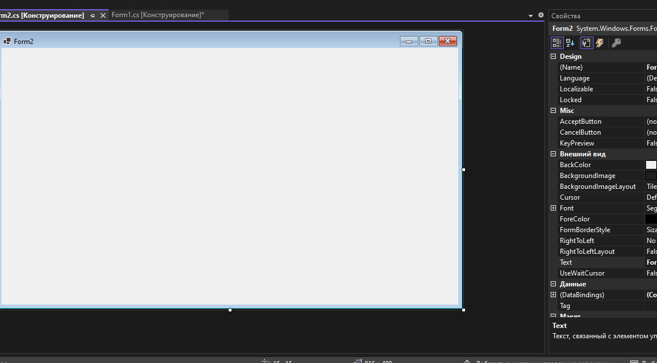
1. Откройте Visual Studio и создайте новый проект Windows Forms. Назовите его WinLinkLabel.

2. Перетащите два элемента управления LinkLabel из Toolbox в форму.

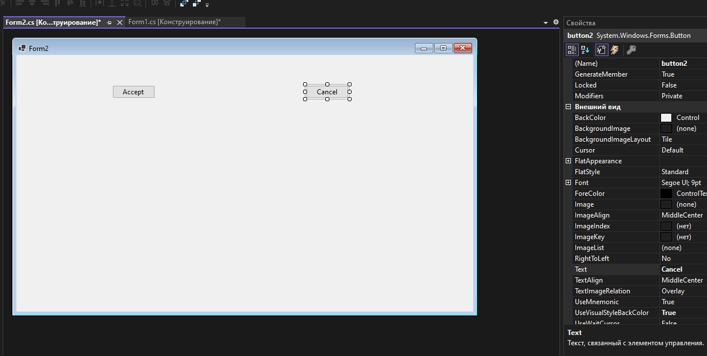
3. В окне Properties для первого элемента задайте свойству Text

значение Open Form, для второго – значение Microsoft.

**Создание диалоговой формы**

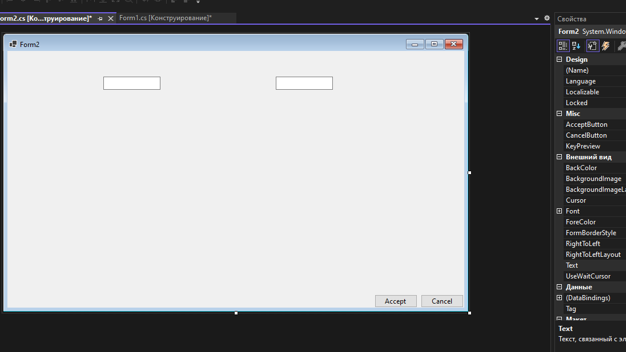


4. В меню Project выберите Add Windows Form и добавьте в свой проект новую форму Windows с именем Form2.



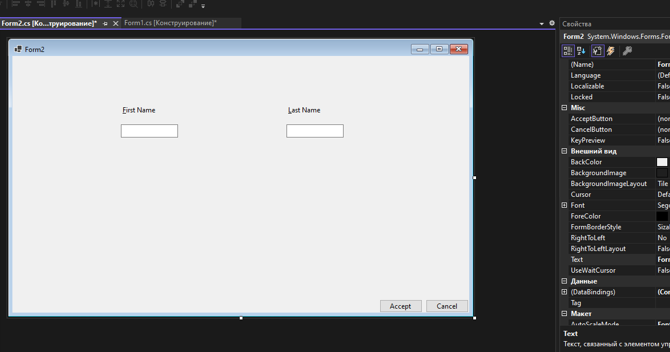
5. В конструкторе перетащите два элемента управления Button в форму Form2. Задайте свойству Text этих кнопок значения Accept и Cancel. Расположите их в правом нижнем углу формы.

6. Задайте свойству DialogResult кнопки Accept значение ОК, а свойству DialogResult кнопки Cancel — значение Cancel.



7. Перетащите два элемента управления TextBox из Toolbox в форму.

8. Задайте свойству Modifiers каждого элемента управления TextBox значение Internal. Свойство Text для них оставьте пустым.



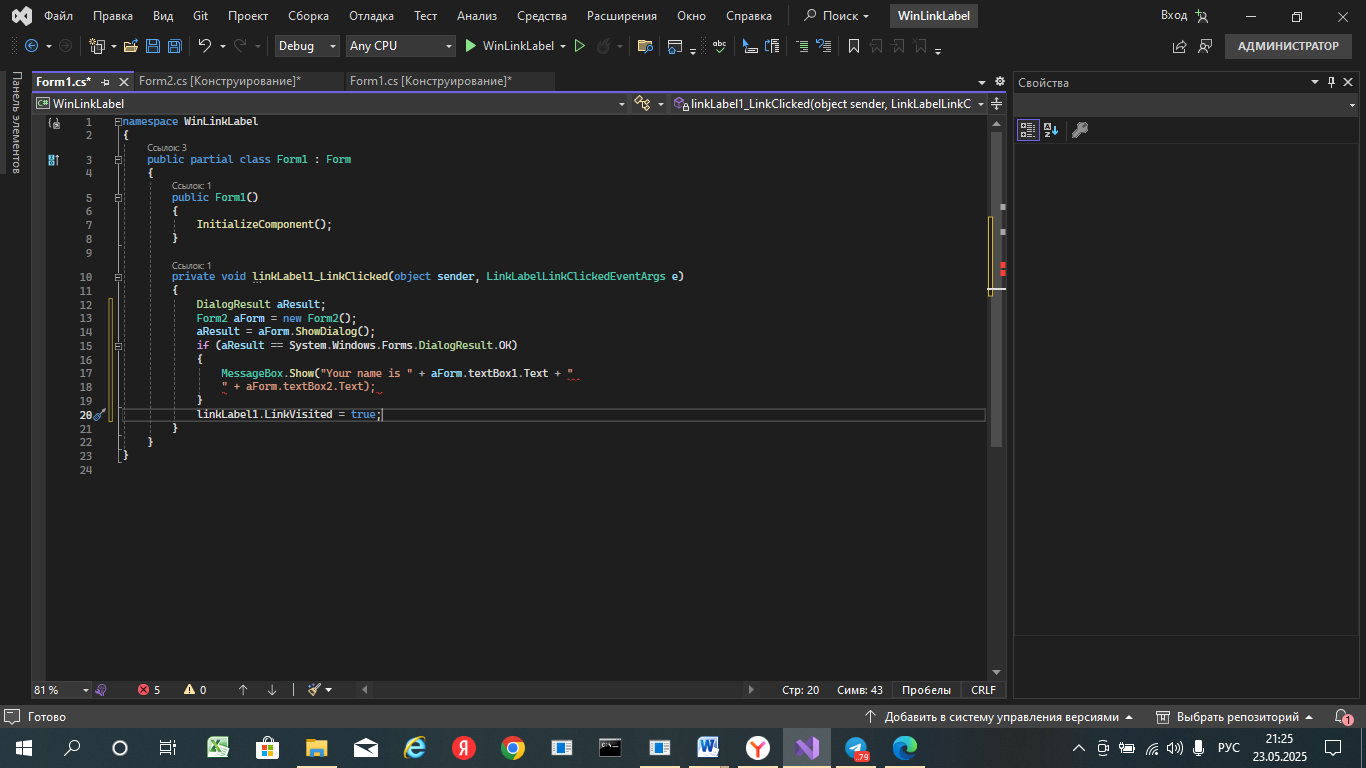
9. Перетащите два элемента управления Label из Toolbox в форму и разместите их рядом с элементом управления TextBox.

10. Задайте свойствам Text элементов управления Label значения &First Name и &Last Name.

11. Проверьте, что свойству UseMnemonic всех надписей установлено значение True.

12. Установите свойство Tablndex в окне Properties, как показано ниже.

**Реализация обработчика события вызова диалогового окна**



13. Выберите в конструкторе закладку для формы Form1. Дважды щелкните первый элемент управления linkLabel1 для создания обработчика события linkLabel1\_LinkClicked. Добавьте следующий код:

DialogResult aResult;

Form2 aForm = new Form2();

aResult = aForm.ShowDialog();

if (aResult == System.Windows.Forms.DialogResult.OK)

{

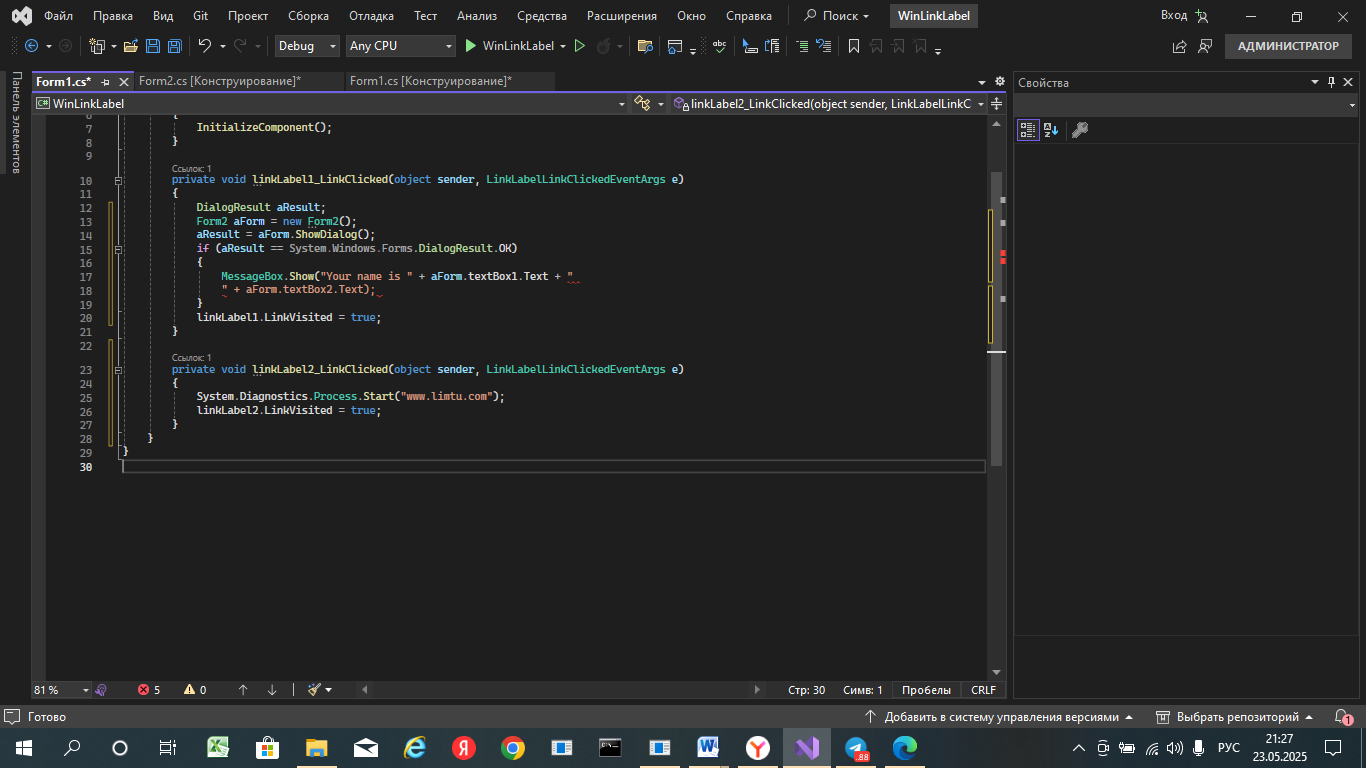
MessageBox.Show("Your name is " + aForm.textBox1.Text + "

" + aForm.textBox2.Text);

}

linkLabel1.LinkVisited = true;

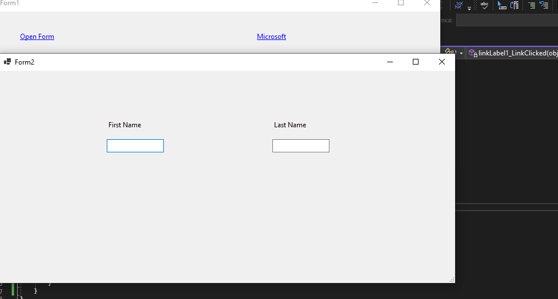
**Реализация обработчика события вызова веб-страницы**



14. Выберите в конструкторе закладку для формы Form1. Дважды щелкните второй элемент управления linkLabel2 для создания обработчика события linkLabel2\_LinkClicked. Добавьте следующий код:

System.Diagnostics.Process.Start("www.limtu.com");

linkLabel2.LinkVisited = true;

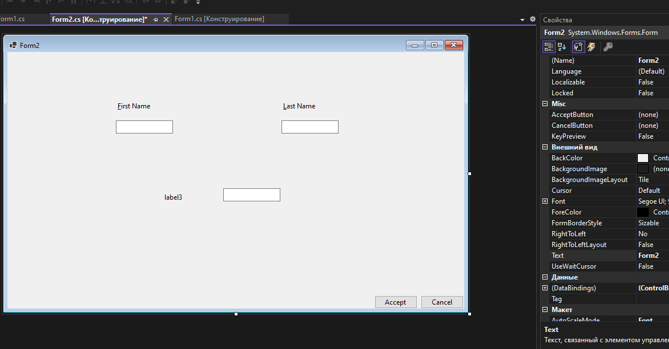


15. Постройте и запустите приложение.

16. Щелкните элемент управления linkLabel – Open Form, чтобы открыть форму. Введите соответствующую информацию в поля ввода и проверьте кнопки Accept и Cancel.

17. Щелкните элемент управления linkLabe2 – Microsoft, чтобы открыть сайт известного учебного центра.

**Упражнение 7. Элементы с поддержкой редактирования текста**



1. Откройте выполненное вами в предыдущем упражнении решение WinLinkLabel.

2. Отобразите конструктор для формы Form2.

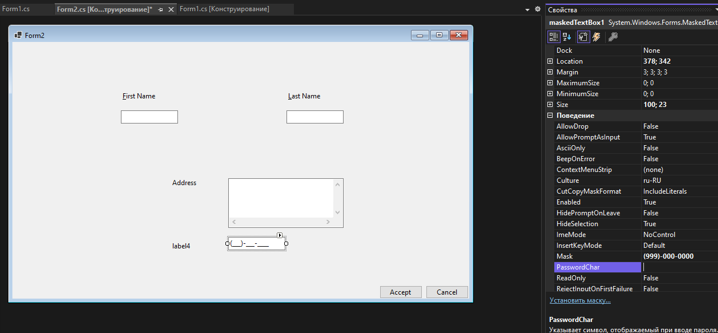
3. Добавьте элемент управления TextBox на форму под расположенными ранее элементами. Перетащите элемент управления Label в форму и разместите слева от этого элемента.



4. Задайте свойству Text элемента управления Label значение Address.

5. Для элемента управления TextBox задайте следующие свойства:

6. Измените размеры элемента управления TextBox так, чтобы он вмещал адрес. При необходимости увеличьте размеры формы и переместите кнопки Accept и Cancel.

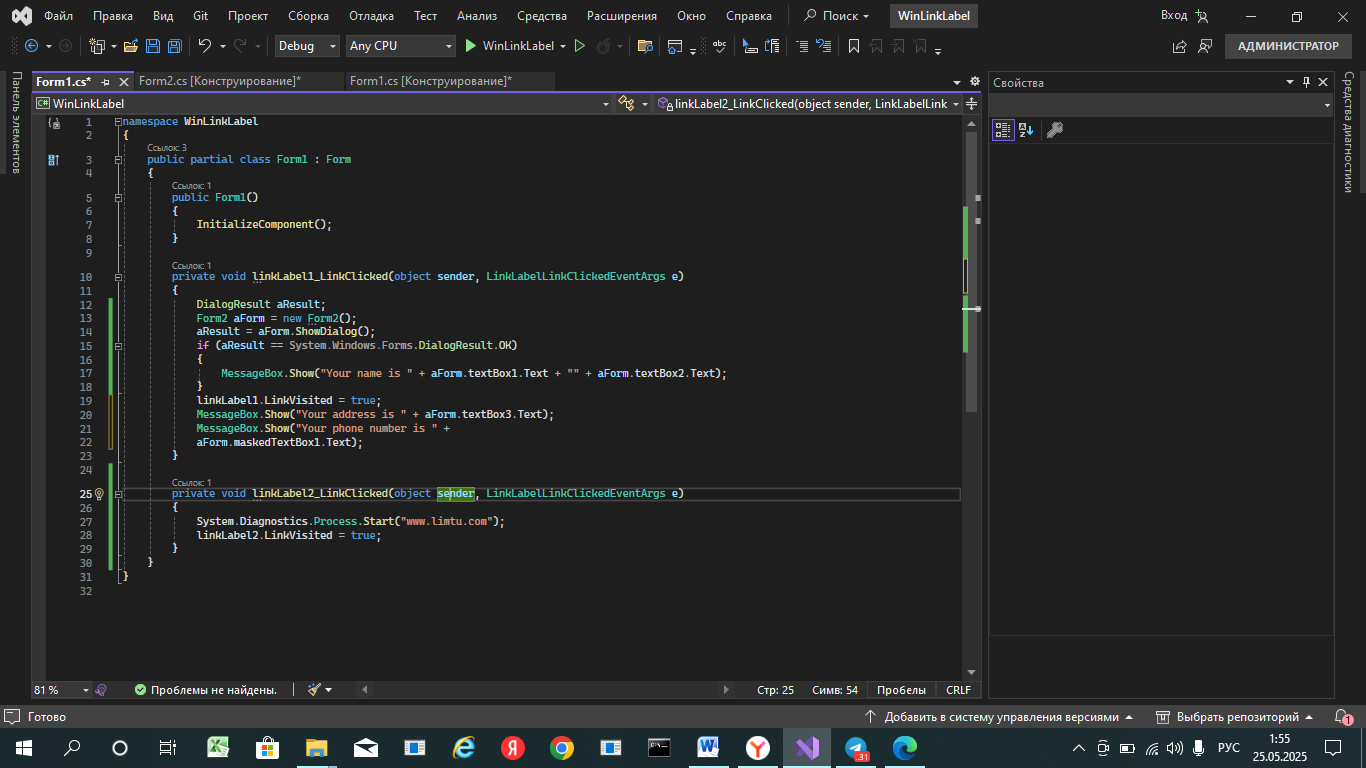


7. Перетащите элементы управления MaskedTextBox и Label из Toolbox на форму и разместите их под ранее введенные элементы.

8. Свойству Text элемента управления Label задайте значение Phone Number.

9. Задайте значение (999)-000-0000 свойству Mask элемента управления MaskedTextBox.

10. Задайте значение Internal свойству Modifiers для последних элементов управления TextBox и MaskedTextBox.



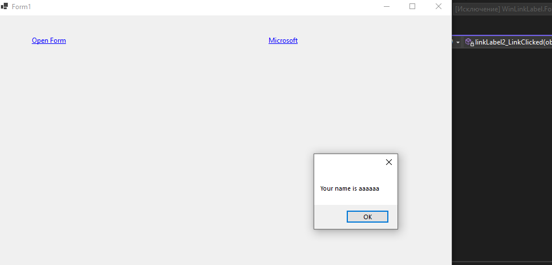
11. Откройте окно кода формы Form1.

12. В обработчике события linkLabel1\_LinkClicked добавьте в блок if, расположенный под кодом, который вы добавили в предыдущем упражнении, следующий код:

MessageBox.Show("Your address is " + aForm.textBox3.Text);

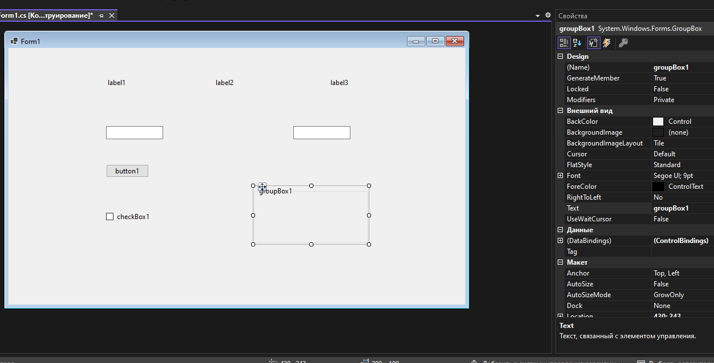
MessageBox.Show("Your phone number is " +

aForm.maskedTextBox1.Text);



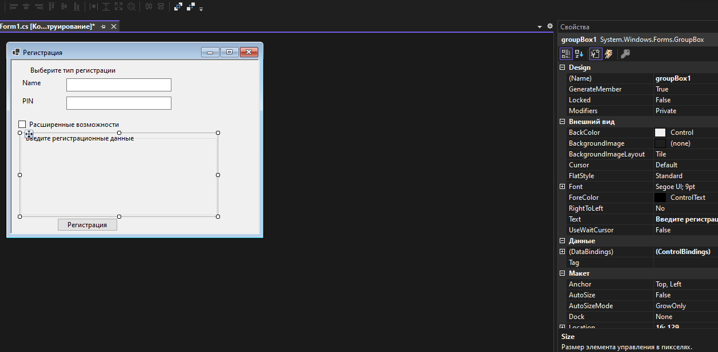
13. Постройте и запустите приложение. Введите в текстовое поле свой телефон. Проверьте, что номер отображается согласно требуемому формату.

**Упражнение 8. Добавление и удаление элементов управления в режиме работы приложения**

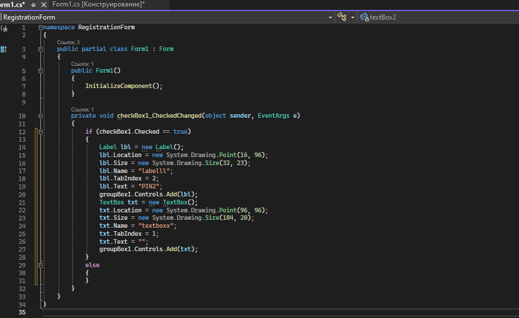


1. Создайте новое приложение и назовите его RegistrationForm.

2. Добавьте на форму три надписи, два текстовых поля, кнопку, элементы CheckBox и GroupBox



3. Установите следующие значения свойств формы и элементов управления:



4. Для реализации возможности добавления и удаления элементов в процессе выполнения программы реализуйте обработчик события CheckedChanged: щелкните дважды на элементе checkBox1 и добавьте следующий код:

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 = "labelll";

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;

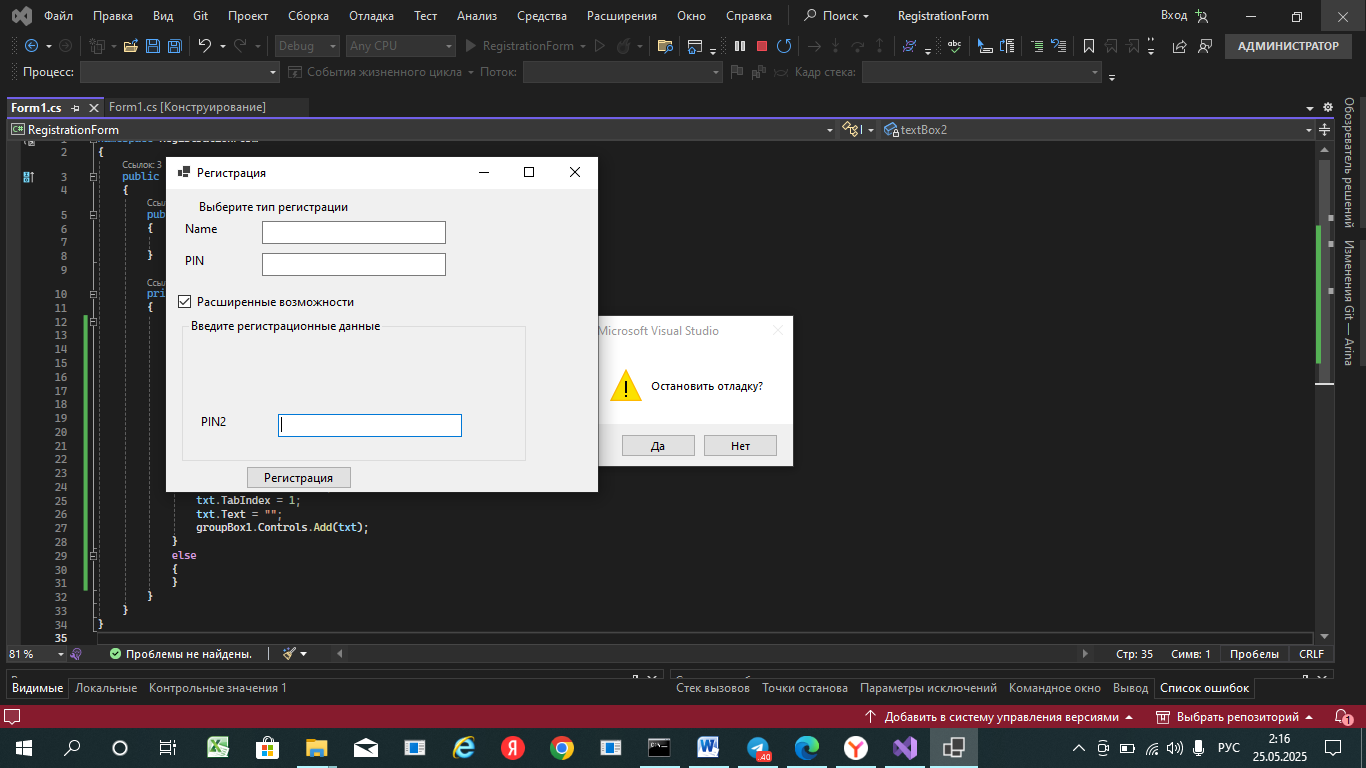
txt.Text = "";

groupBox1.Controls.Add(txt);

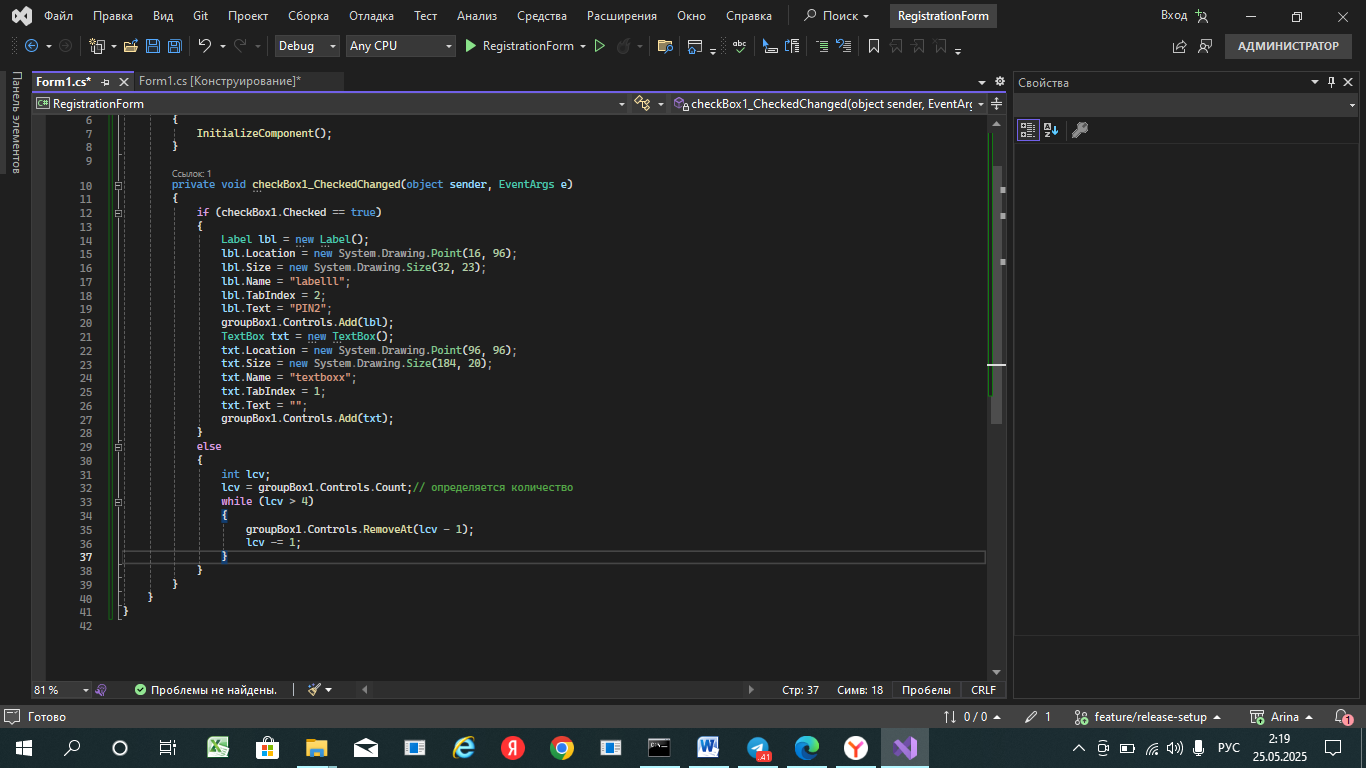
}

else{

}



5. Откомпилируйте и запустите приложение. Проверьте, что при установке флажка в ЭУ checkBox “Расширенные возможности” на форме появляется надпись и поле ввода для дополнительных данных.



6. Для удаления ЭУ с формы могут применяться методы: Clear (удаление всех элементов из коллекции), Remove (удаление элемента из коллекции) и RemoveAt (удаление элемента по заданному индексу). В тело оператора else добавьте код для удаления ЭУ по индексу:

int lcv;

lcv = groupBox1.Controls.Count;// определяется количество

while (lcv > 4)

{

groupBox1.Controls.RemoveAt(lcv - 1);

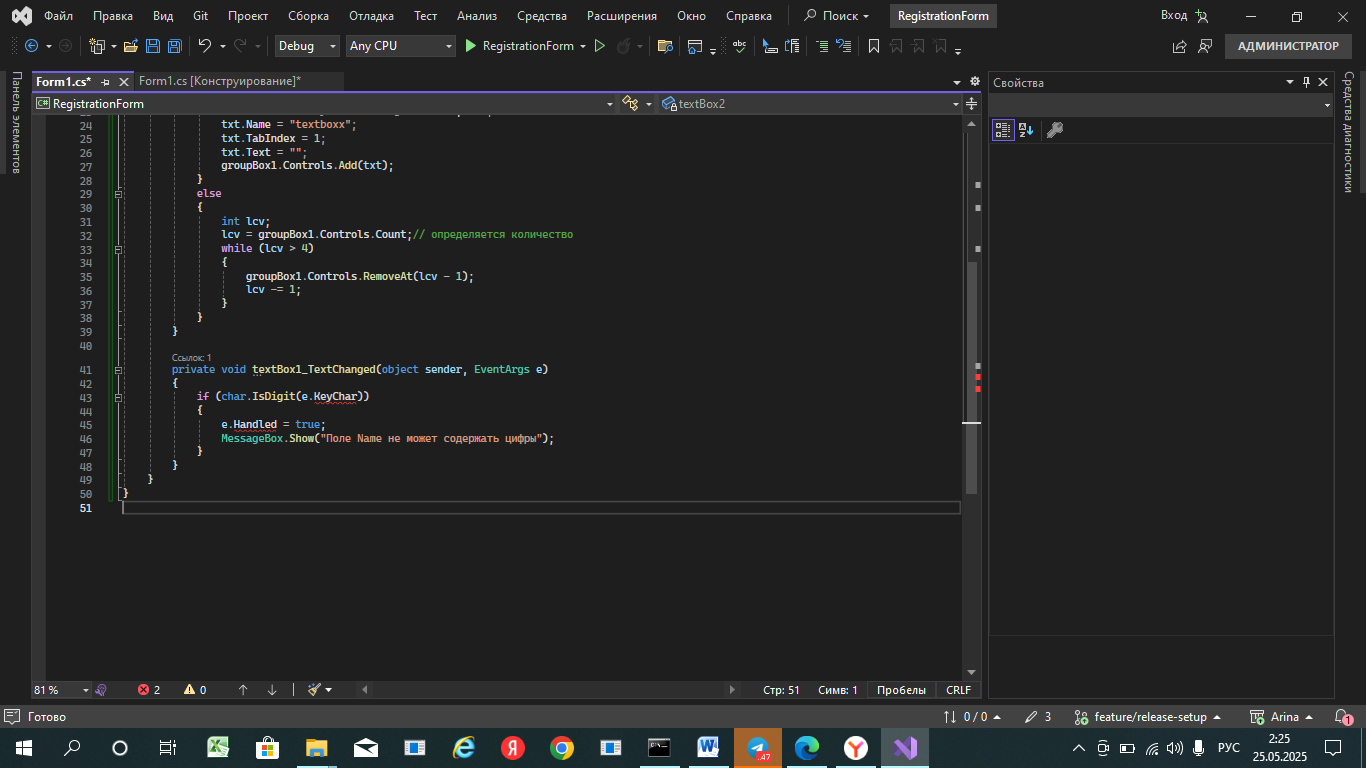
lcv -= 1;

}

7. Запустите приложение. Убедитесь, что при включении “Расширенные возможности” дополнительные элементы появляются на форме, а при выключении – исчезают.

**Упражнение 9. Проверка вводимых значений. События KeyPress и Validating. Элемент управления ErrorProvider**

**Использование события KeyPress**



1. Откройте приложение RegistrationForm.

2. Выделите поочередно текстовые поля TextBox1 и TextBox2, в окне Properties создайте обработчики события KeyPress, возникающего при нажатии любой клавиши в поле.

3. В тело обработчика события KeyPress для текстового поля TextBox1 укажите следующий код (для элемента TextBox1 недопустимыми значениями будут цифры):

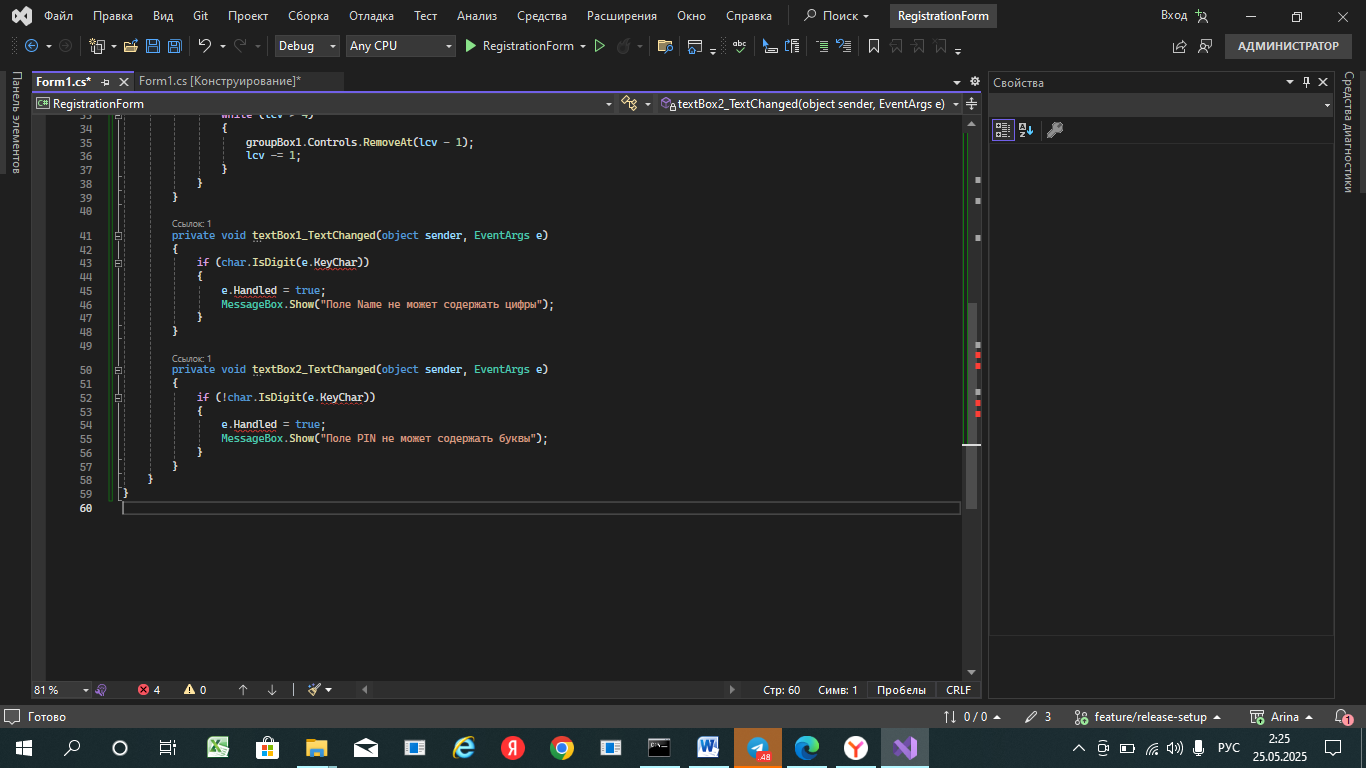
if (char.IsDigit(e.KeyChar) )

{

e.Handled = true;

MessageBox.Show("Поле Name не может содержать цифры");

}



4. Для элемента TextBox2, наоборот, недопустимыми значениями будут буквы, в обработчике события KeyPress для текстового поля TextBox2 укажите код:

if (!char.IsDigit(e.KeyChar) )

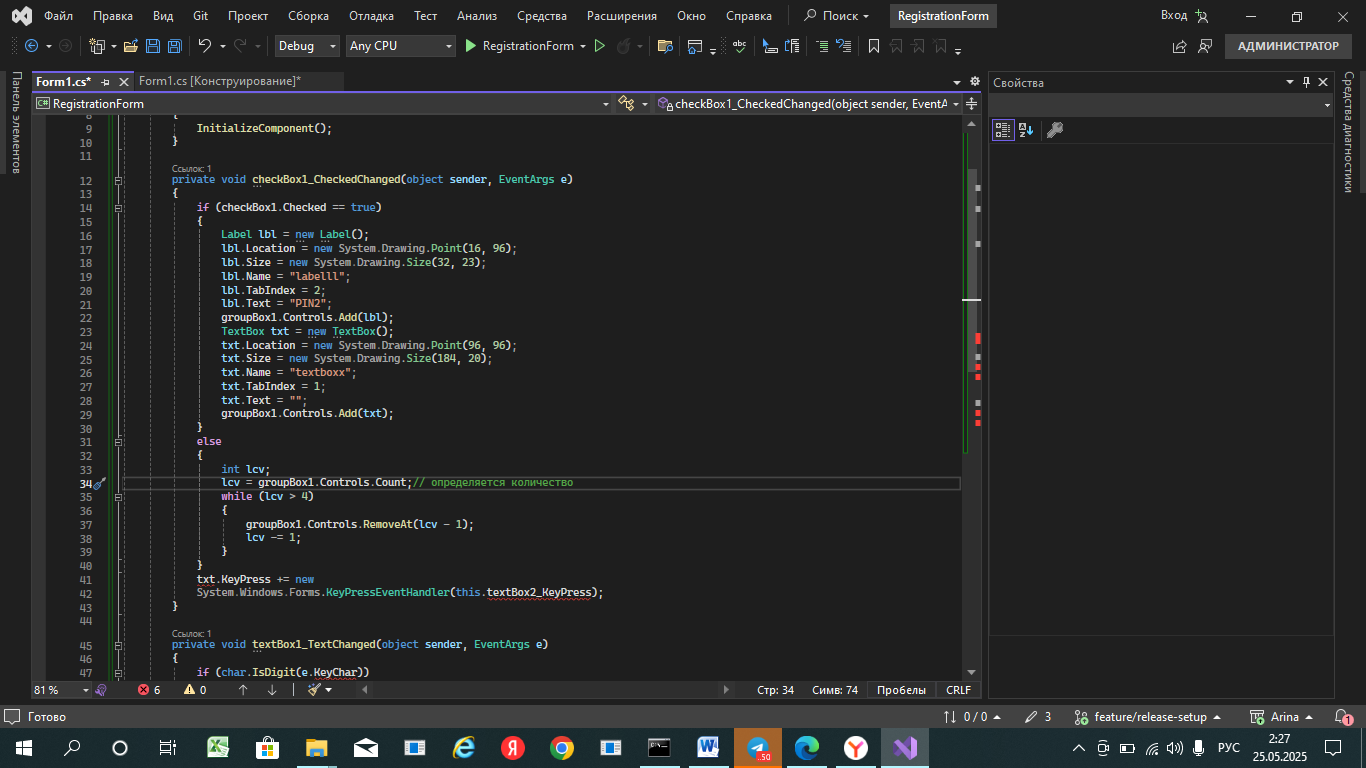
{

e.Handled = true;

MessageBox.Show("Поле PIN не может содержать буквы");

}

5. Откомпилируйте и запустите приложение. Попробуйте ввести в поле Name цифры, в поле PIN – буквы.



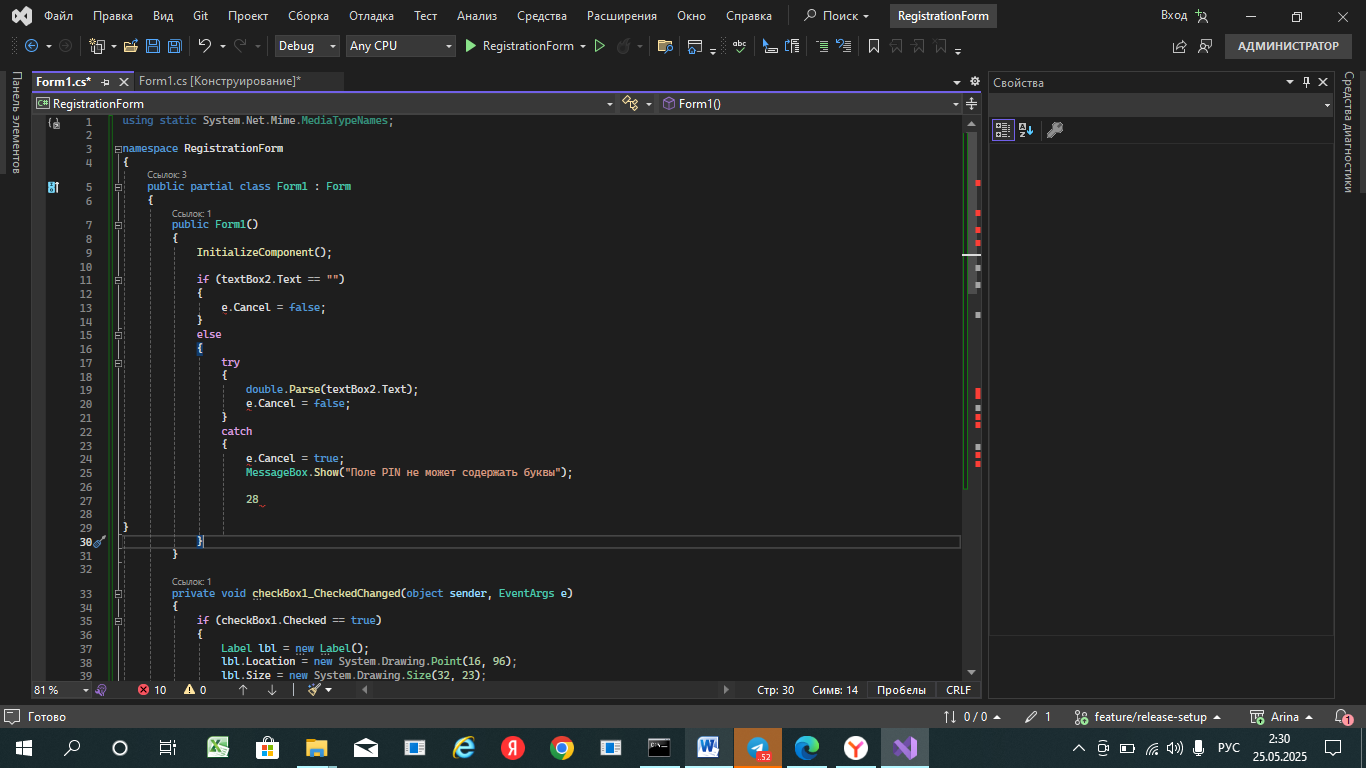
6. Для защиты текстового поля, появляющегося при установке галочки в чекбоксе "Расширенные возможности", необходимо вручную определить событие KeyPress. В обработчике события CheckedChanged для элемента CheckBox1 укажите код:

txt.KeyPress+= new

System.Windows.Forms.KeyPressEventHandler(this.textBox2\_KeyPress);

7. Запустите и протестируйте приложение.

**Применение события Validating**



8. Закомментируйте обработчик элемента TextBox2.

9. В режиме дизайна формы в окне Properties элемента TextBox2 создайте обработчик события Validating и запишите следующий код:

if(textBox2.Text =="")

{

e.Cancel=false;

}

else

{

try

{

double.Parse(textBox2.Text);

e.Cancel = false;

}

catch

{

e.Cancel = true;

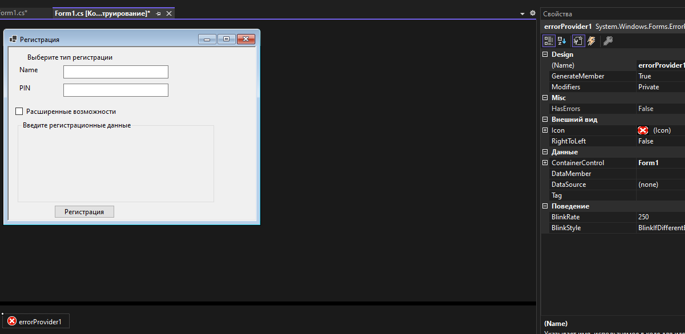
MessageBox.Show("Поле PIN не может содержать буквы");

}

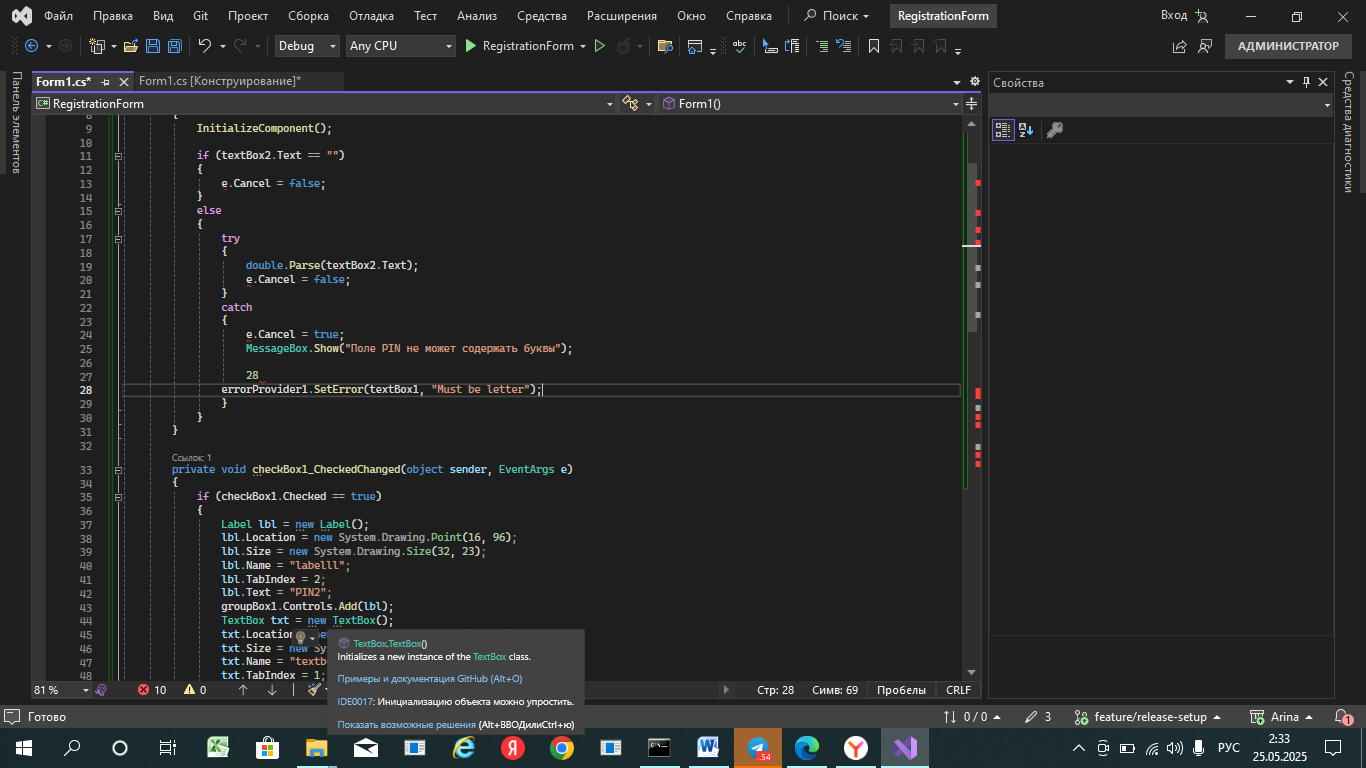
}

10. Запустите приложение. При переключении фокуса ввода или нажатии на кнопку "регистрация" происходит событие Validating.

**Применение элемента управления ErrorProvider**



11. В режиме дизайна из окна ToolBox перенесите на форму элемент управления ErrorProvider.



12. В коде формы в обработчике textBox1\_KeyPress добавьте следующую строку:

errorProvider1.SetError(textBox1, "Must be letter");

13. Запустите приложение. При ошибке ввода появляется мигающая иконка уведомления, при наведении на нее всплывает поясняющее сообщение об ошибке.