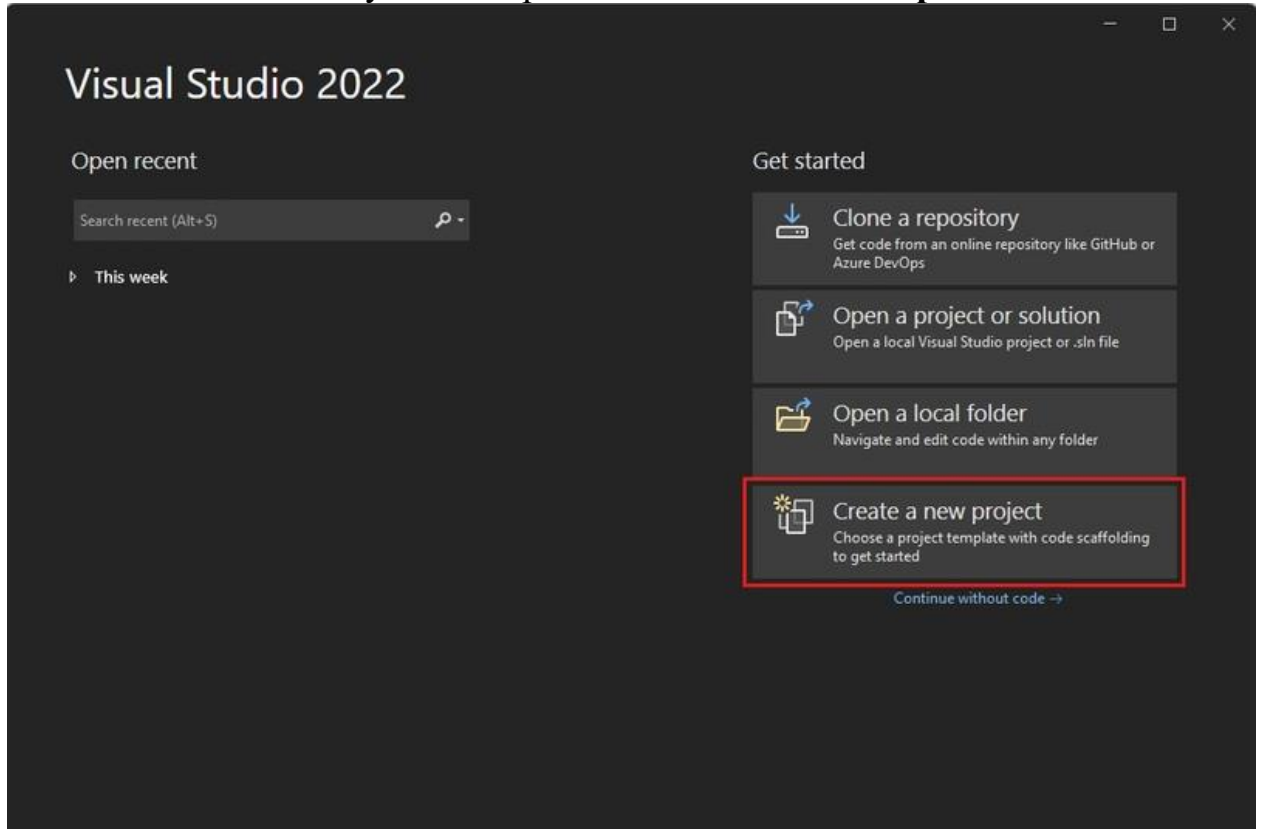


Практическая работа №3

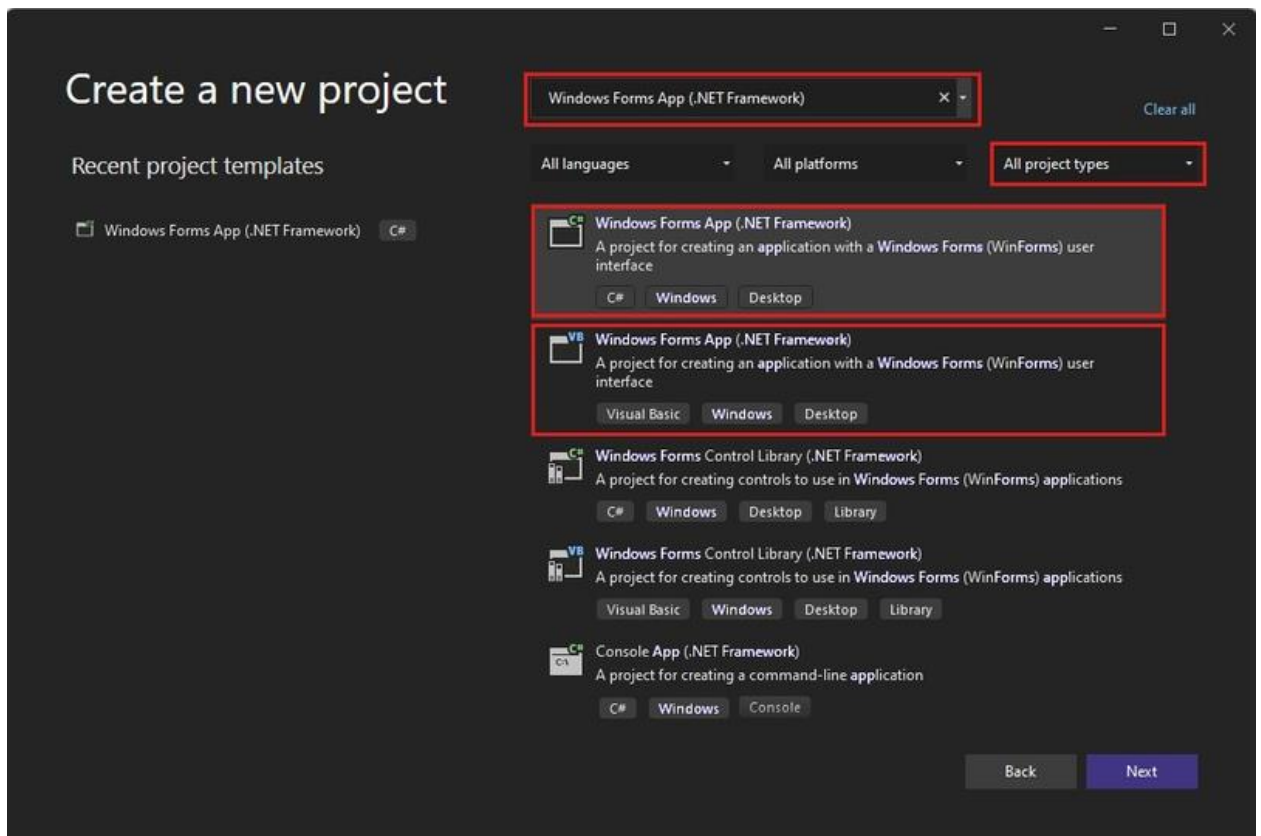
Первый шаг в создании программы для просмотра изображений — это создание проекта приложения Windows Forms.

1. Запустите Visual Studio.
2. В окне запуска выберите **Создание нового проекта**.



3. В окне **Создать проект** выполните поиск по фразе *Windows Forms*. Затем в списке **Тип проекта** выберите **Рабочий стол**.

4. Выберите шаблон **Приложение Windows Forms (.NET Framework)** для C# или Visual Basic, а затем нажмите **Далее**.



5. В окне **Настроить новый проект** назовите проект *PictureViewer*, а затем выберите **Создать**.

Visual Studio создает решение для приложения. Решение является контейнером для всех проектов и файлов, необходимых приложению.

На этом этапе Visual Studio отображает пустую форму в **конструкторе Windows Forms**.

Добавление элемента макета

Приложение для просмотра изображений содержит поле рисунка, флажок и четыре кнопки.

Элемент макета определяет свое расположение в форме. В этом разделе показано, как изменить название формы, ее размер и добавить элемент макета.

1. В проекте выберите конструктор Windows Forms. На этой вкладке для C# считывается файл **Form1.cs [Design]**.

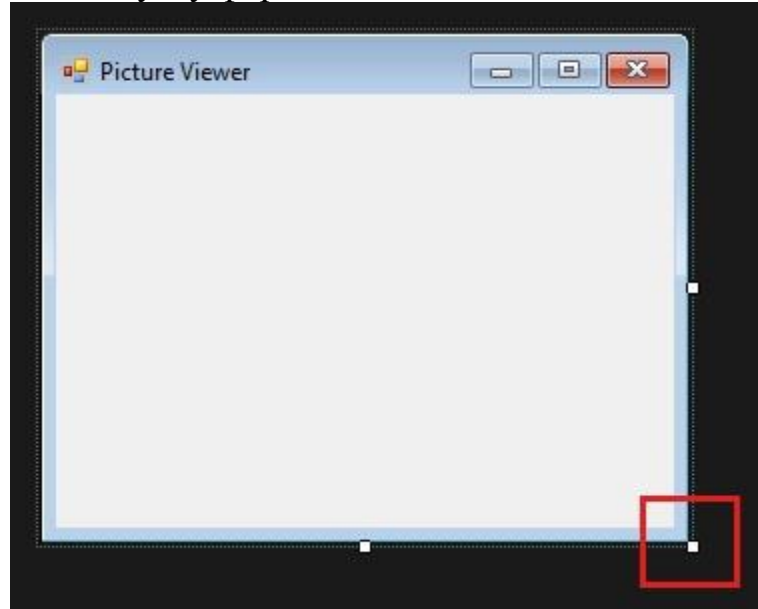
2. Нажмите в любом месте в **Form1**.

3. В окне **Свойства** теперь отображаются свойства формы. Окно **Свойства** обычно находится в правом нижнем углу окна Visual Studio. Этот раздел управляет различными свойствами, такими как цвет переднего плана и фона, текст заголовка, отображаемый в верхней части формы, и размер формы. Если окно **Свойства** не отображается, выберите **Вид>Окно свойств**.

4. В окне **Свойства** найдите свойство **Text**. В зависимости от того, как отсортирован список, может потребоваться прокрутить вниз. Введите *Средство просмотра изображений*, а затем нажмите клавишу **ВВОД**. Теперь в строке заголовка формы отображается текст **Средство просмотра изображений**.

Свойства можно отображать по категориям или по алфавиту. Для переключения между свойствами в окне **Свойства** используйте соответствующие кнопки.

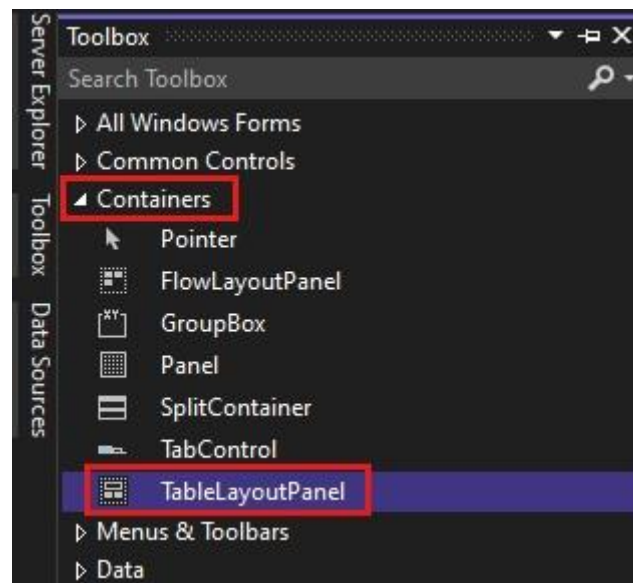
5. Снова выберите форму. Выберите нижний правый маркер перетаскивания формы. Этот маркер представляет собой небольшой белый квадрат в правом нижнем углу формы.



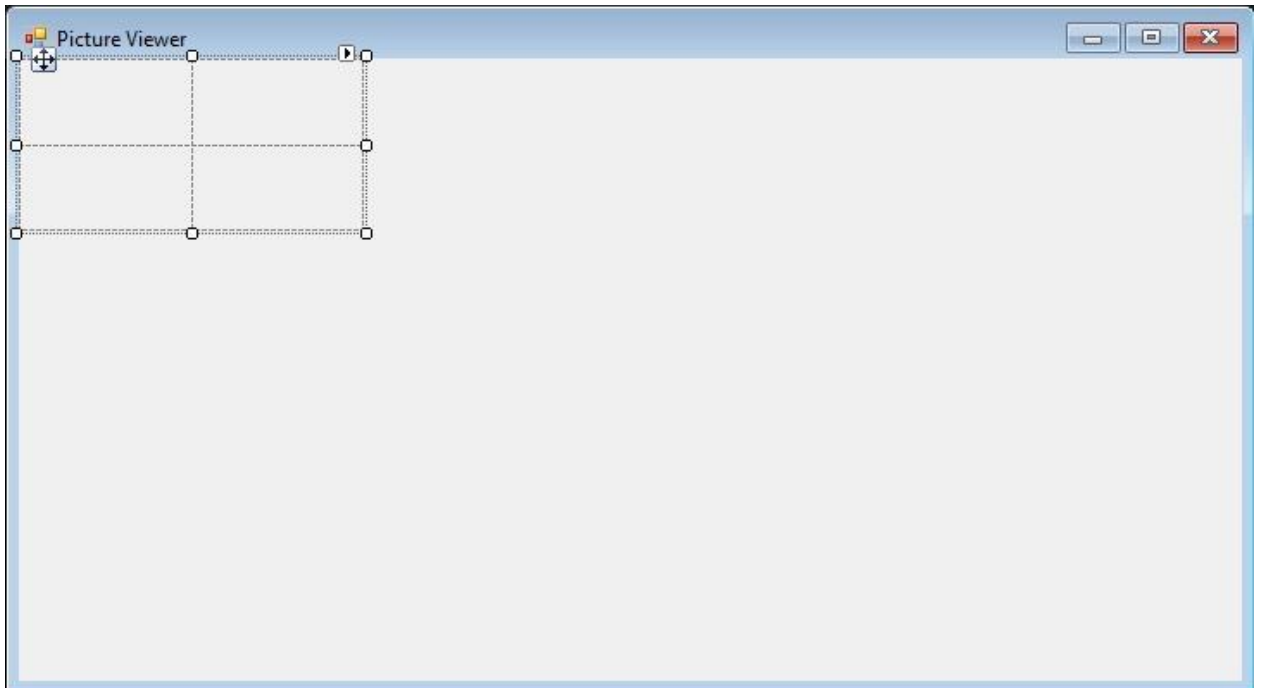
Перетащите маркер, чтобы изменить размер формы — она должна стать шире и немного выше. Если обратиться к окну **Свойства**, вы увидите, что свойства **Size** изменилось. Для изменения размера формы также можно использовать свойство **Size**.

6. В левой части интегрированной среды разработки Visual Studio выберите вкладку **Панель элементов**. Если вы ее не видите, выберите пункт **Представление > Панель элементов** в строке меню или воспользуйтесь комбинацией клавиш **CTRL+ALT+X**.

7. Выберите маленький треугольник рядом с группой **Контейнеры**, чтобы открыть ее.

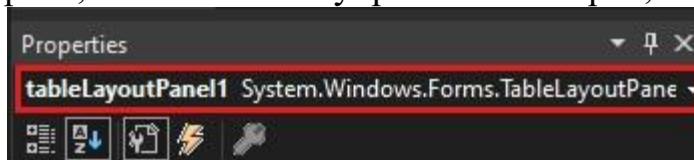


8. Дважды щелкните элемент управления **TableLayoutPanel** на панели элементов. Можно также перетащить элемент управления с панели элементов в форму. Элемент управления TableLayoutPanel появится в форме

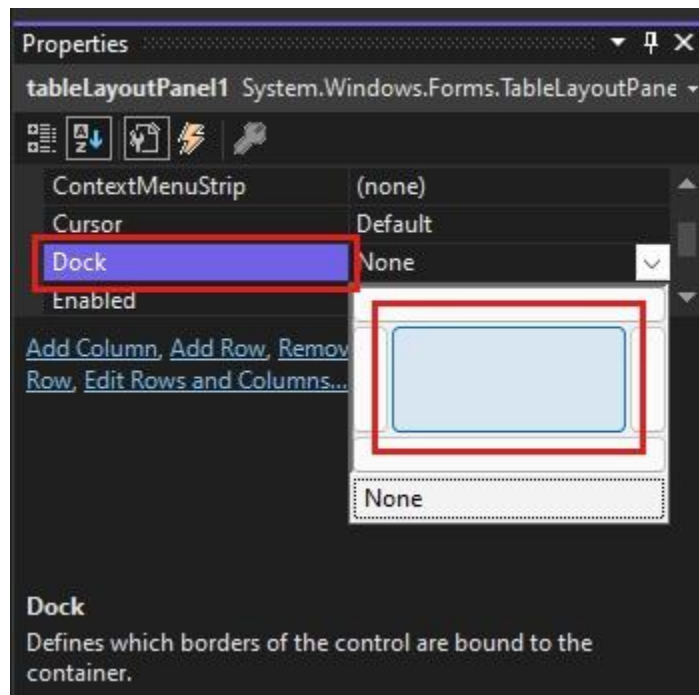


*После добавления элемента управления TableLayoutPanel, если внутри формы появляется окно с заголовком **Задачи TableLayoutPanel**, чтобы закрыть его, щелкните в любом месте внутри формы*

9. Выберите элемент управления **TableLayoutPanel**. Чтобы проверить, какой элемент управления выбран, обратитесь к окну **Свойства**.



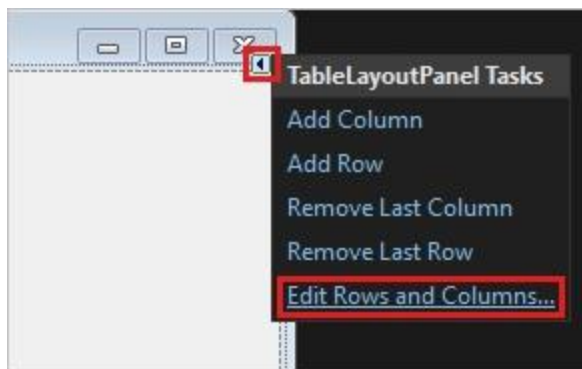
10. Выбрав элемент управления TableLayoutPanel, найдите свойство **Dock**, значение которого — **None**. Нажмите стрелку раскрывающегося меню и выберите **Fill** — это большая кнопка по середине раскрывающегося меню.



Закрепление означает то, как окно присоединено к другому окну или области. Теперь TableLayoutPanel заполняет всю форму. Если снова изменить размер формы, элемент управления TableLayoutPanel останется закрепленным и сам изменит свой размер для заполнения формы.

11. В форме выберите TableLayoutPanel. В правом верхнем углу расположена маленькая кнопка с черным треугольником. Выберите треугольник, чтобы отобразить список задач элемента управления.

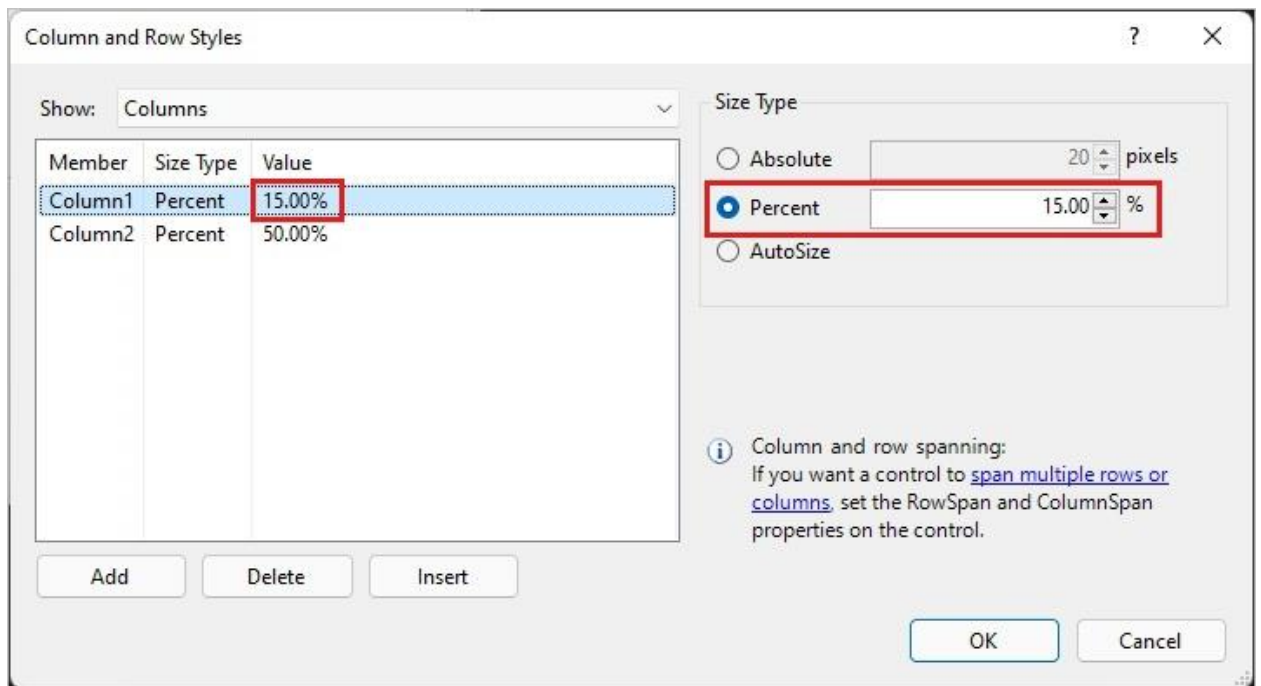
12.



12. Выберите задачу **Изменить строки и столбцы**, чтобы открыть диалоговое окно **Стили столбцов и строк**.

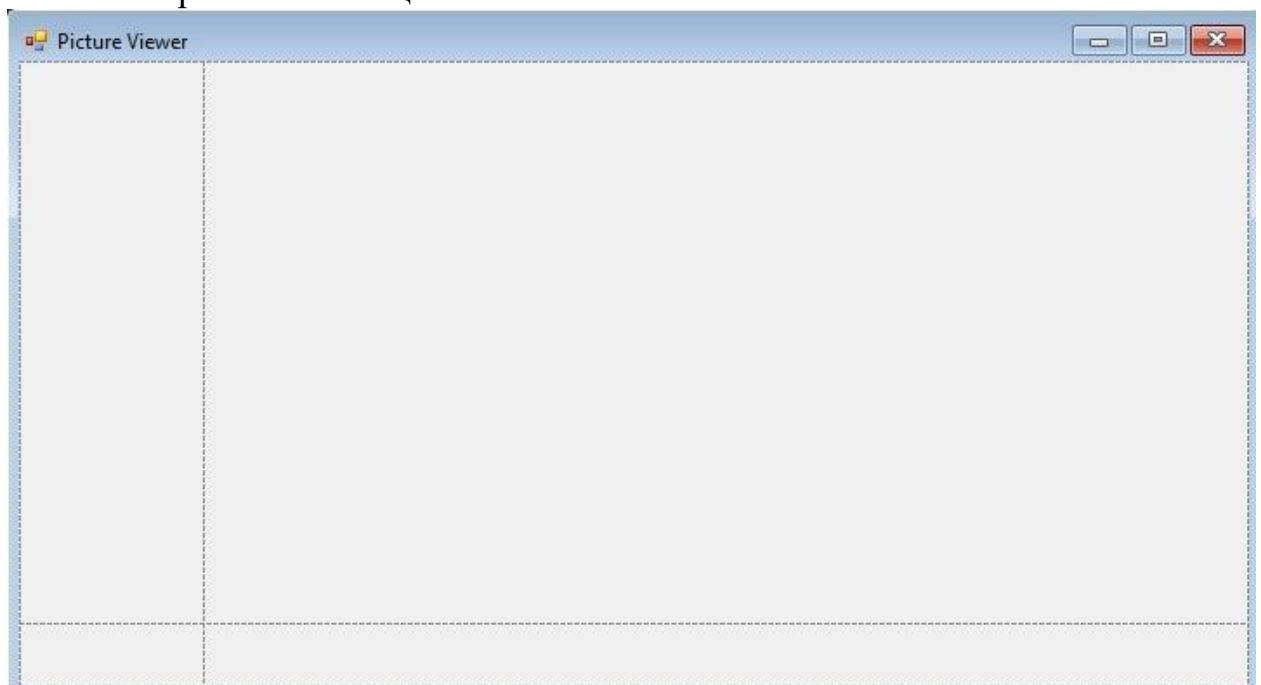
13. Выберите **Column1** и задайте размер 15 процентов. Убедитесь, что кнопка **Проценты** нажата.

14. Выберите **Column2** и задайте значение 85 процентов.



15. В меню **Показать** в верхней части диалогового окна **Стили столбцов и строк** выберите **Строки**. Задайте для **Row1** значение 90 процентов, а для **Row2** 10 процентов. Нажмите кнопку **ОК**, чтобы сохранить изменения.

Элемент управления `TableLayoutPanel` теперь содержит большую верхнюю строку, маленькую нижнюю строку, маленький левый столбец и большой правый столбец.



Макет готов.

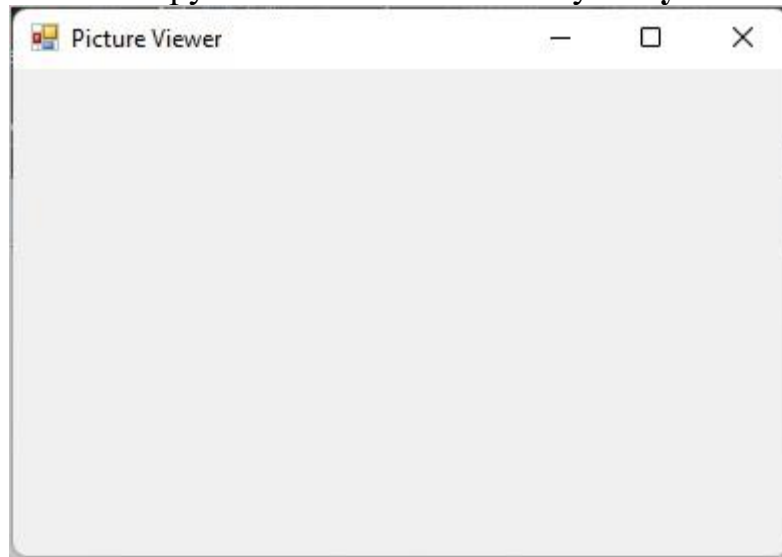
*Перед выполнением приложения сохраните его, нажав на панели инструментов кнопку **Сохранить все**. Кроме того, чтобы сохранить приложение, в строке меню можно выбрать пункт **Файл>Сохранить все** или нажать клавиши **CTRL+SHIFT+S**. Рекомендуется выполнять сохранение от начала разработки и как можно чаще.*

Запустите приложение.

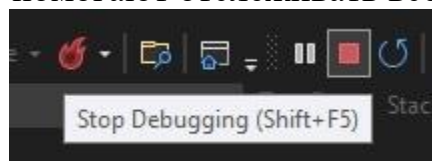
При создании проекта приложения Windows Forms выполняется сборка программы, которая в последующем запускается. На этом этапе функции приложения "Средство просмотра изображений" ограничены. Пока оно просто открывает пустое окно с надписью **Средство просмотра изображений** в строке заголовка.

Чтобы запустить приложение, выполните следующие действия.

1. Используйте один из следующих методов. Visual Studio запустит приложение. Появится окно с названием **Средство просмотра изображений**.
 - a. Нажмите клавишу **F5**.
 - b. В строке меню выберите **Отладка>Начать отладку**.
 - c. На панели инструментов нажмите кнопку **Запустить**.



Обратите внимание на панель инструментов Visual Studio IDE. При запуске приложения на панели инструментов появляются дополнительные кнопки. Эти кнопки позволяют выполнять такие действия, как остановка и запуск приложения, а также помогают отслеживать все ошибки.



1. Для остановки приложения используйте один из указанных ниже методов. Запуск приложения в интегрированной среде разработки Visual Studio называется отладкой. Приложение запускается для поиска и исправления ошибок. Для запуска и отладки других программ следует выполнить ту же процедуру.

- a. На панели инструментов нажмите кнопку **Остановить отладку**.
- b. В строке меню выберите **Отладка>Остановить отладку**.
- c. На клавиатуре нажмите клавиши **SHIFT+F5**.
- d. Нажмите кнопку **X** в верхнем углу окна **Средство просмотра изображений**.

Добавление элементов управления в приложение

Приложение "Средство просмотра изображений" использует элемент управления PictureBox для вывода изображения. Оно использует флажок и несколько кнопок для управления изображением и фоном, а также для закрытия приложения. Вы добавите элемент PictureBox и флажок из области панели элементов в Visual Studio IDE.

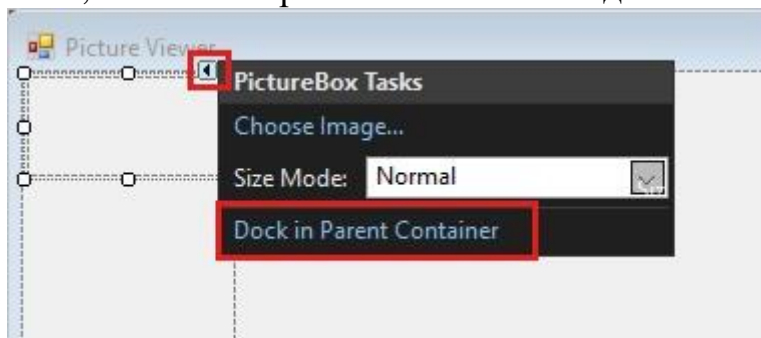
1. Запустите Visual Studio. Проект средства просмотра изображений находится в разделе **Открыть последние**.

2. В конструкторе **Windows Forms** выберите элемент управления **TableLayoutPanel**, добавленный в предыдущем учебнике. Убедитесь, что в окне **Свойства** отображается **tableLayoutPanel1**.

3. В левой части интегрированной среды разработки Visual Studio выберите вкладку **Панель элементов**. Если вы ее не видите, выберите пункт **Представление > Панель элементов** в строке меню или воспользуйтесь комбинацией клавиш **CTRL+ALT+X**. На панели элементов разверните узел **Общие элементы управления**.

4. Дважды щелкните элемент **PictureBox**, чтобы добавить в форму элемент управления PictureBox. Visual Studio IDE добавит элемент управления PictureBox в первую пустую ячейку **TableLayoutPanel**.

5. Щелкните новый элемент управления **PictureBox**, чтобы выбрать его, а затем щелкните черный треугольник на новом элементе управления PictureBox, чтобы отобразить его список задач.



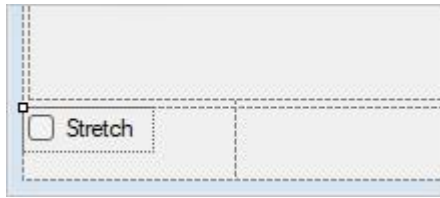
6. Выберите **Закрепить в родительском контейнере**, который задает для свойства **Dock** элемента управления PictureBox значение **Fill**. Это значение отображается в окне **Свойства**.

7. В окне **Свойства** для элемента управления PictureBox задайте для свойства **ColumnSpan** значение **2**. Теперь элемент управления PictureBox заполняет оба столбца.

8. Установите для его свойства **BorderStyle** значение **Fixed3D**.

9. В конструкторе **Windows Forms** выберите элемент управления **TableLayoutPanel**. Затем двойным щелчком выберите элемент **CheckBox** на панели элементов, чтобы добавить новый элемент управления CheckBox в следующую свободную ячейку таблицы. Элемент управления PictureBox занимает первые две ячейки в **TableLayoutPanel**, поэтому элемент управления CheckBox добавляется в нижнюю левую ячейку.

10. Выберите свойство **Text** и введите **Stretch**.



Добавление кнопок на панель макета

Мы добавили элементы управления в `TableLayoutPanel`. Далее показано, как добавить четыре кнопки в новую панель макета в `TableLayoutPanel`.

1. В форме выберите элемент управления `TableLayoutPanel`. Откройте **Панель элементов**, выберите **Контейнеры**. Дважды щелкните элемент управления `FlowLayoutPanel` для добавления нового элемента в последнюю ячейку `TableLayoutPanel`.

2. Присвойте свойству **Dock** элемента `FlowLayoutPanel` значение **Fill**. Это свойство можно задать, щелкнув черный треугольник и выбрав **Закрепить в родительском контейнере**. Элемент управления `FlowLayoutPanel` является контейнером, в котором другие элементы управления размещаются построчно в определенном порядке.

3. Выберите новый элемент `FlowLayoutPanel`, а затем откройте **Панель элементов** и выберите **Общие элементы управления**. Дважды щелкните элемент **Кнопки**, чтобы добавить кнопку с именем **button1**.

4. Снова дважды щелкните элемент **Кнопка**, чтобы добавить еще одну кнопку. Интегрированная среда разработки вызывает следующий элемент **button2**.

5. Добавьте еще две кнопки таким же образом. Другой вариант — выберите **button2**, а затем выберите **Правка > Копировать** или нажмите клавиши **CTRL+C**. Далее в строке меню выберите **Правка > Вставить** (или нажмите клавиши **CTRL+V**). Порядок вставки копии кнопки. Повторите вставку еще раз. Обратите внимание на то, что в интегрированной среде разработки были добавлены кнопки **button3** и **button4** в `FlowLayoutPanel`.

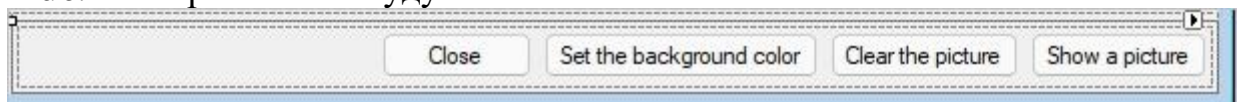
6. Выберите первую кнопку и установите для ее свойства **Text** значение **Показать рисунок**.

7. Установите для свойства **Text** следующих трех кнопок значения **Очистить рисунок**, **Установить цвет фона** и **Заккрыть**.

8. Чтобы изменить размер кнопок и расположить их, выберите элемент `FlowLayoutPanel`. Присвойте свойству **FlowDirection** значение **RightToLeft**. Кнопки должны сами выровняться по правой стороне ячейки и изменить свой порядок таким образом, чтобы кнопка **Показать рисунок** располагалась с правой стороны. Можно перетаскивать кнопки в элементе `FlowLayoutPanel`, чтобы разместить их в любом порядке.

9. Щелкните кнопку **Заккрыть**, чтобы выбрать ее. Затем выберите остальные кнопки, удерживая нажатой клавишу **CTRL**.

10. В окне **Свойства** задайте для свойства **Авторазмер** значение **True**. Размеры кнопок будут изменяться соответствии с текстом.

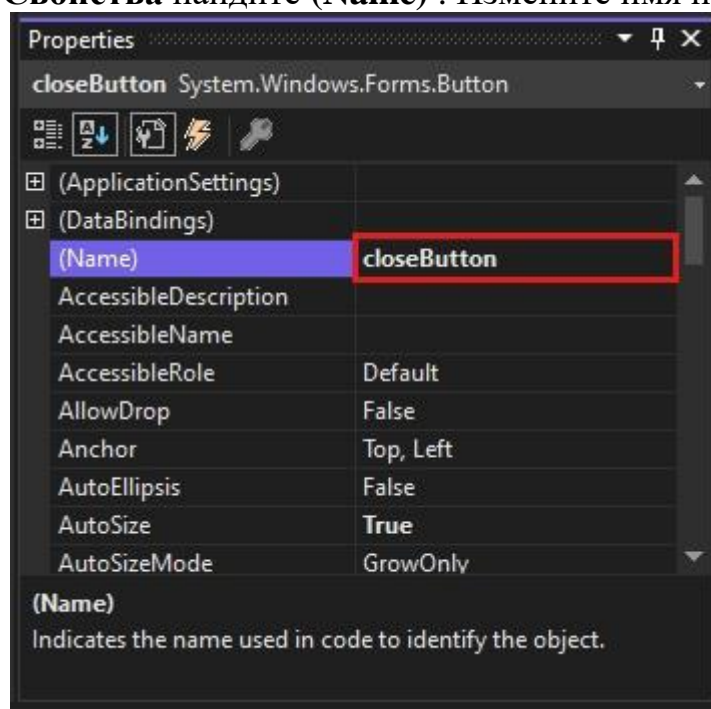


Вы можете запустить программу, чтобы увидеть, как выглядят элементы управления. Нажмите клавишу **F5**, выберите **Отладка > Начать отладку** или нажмите кнопку **Запустить**. Кнопки, которые вы добавили, пока еще не функционируют.

Переименование элементов управления

В форме есть четыре кнопки: **button1**, **button2**, **button3** и **button4** в C#. В Visual Basic в качестве первой буквы любого имени элемента управления по умолчанию используется прописная буква, поэтому в Visual Basic кнопки называются **Button1**, **Button2**, **Button3** и **Button4**. Чтобы присвоить им более информативные имена, выполните указанные ниже действия.

1. В форме нажмите кнопку **Заккрыть**. Если все еще выделены все кнопки, для отмены выделения нажмите клавишу **ESC**.
2. В окне **Свойства** найдите **(Name)**. Измените имя на **closeButton**.



Интегрированная среда разработки не принимает имена, содержащие пробелы.

3. Переименуйте другие три кнопки как **backgroundButton**, **clearButton**, **showButton**. Имена можно проверить в раскрывающемся списке селектора элементов управления в окне **Свойства**. Отобразятся новые имена кнопок.

Можно переименовать любой элемент управления, например **TableLayoutPanel** или **checkbox**.

Добавление компонентов диалоговых окон

С помощью компонентов приложение может открывать графические файлы и выбирать цвет фона. Компонент аналогичен элементу управления. Для добавления компонента в форму используется **панель элементов**. Для задания значений свойств используется окно **Свойства**.

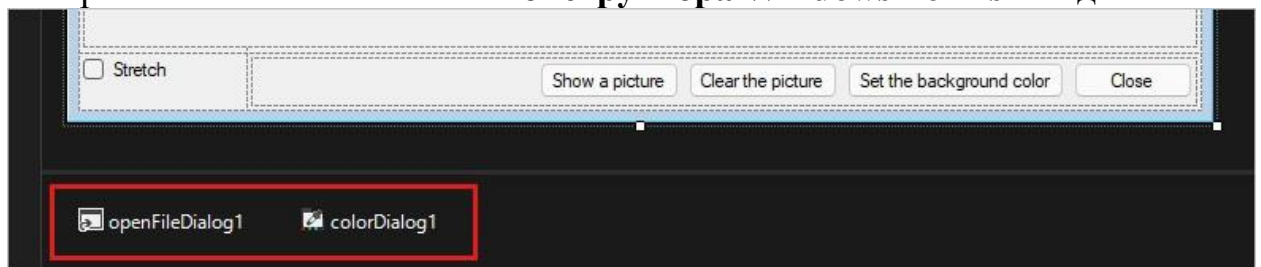
В отличие от элемента управления, добавление в форму компонента не добавляет в форму визуальный элемент. Вместо этого, компонент предоставляет определенное поведение, которое можно включать в коде. Например, это компонент, который открывает диалоговое окно **Открыть файл**.

В этом разделе добавьте в форму компоненты **OpenFileDialog** и **ColorDialog**.

1. Выберите **конструктор Windows Forms (Form1.cs[Design])**. Затем откройте **Панель элементов** и выберите группу **Диалоговые окна**.

2. Дважды щелкните элемент **OpenFileDialog**, чтобы добавить в форму компонент с именем **openFileDialog1**.

3. Дважды щелкните на панели элементов элемент **ColorDialog**, чтобы добавить в форму компонент с именем **colorDialog1**. Компоненты отображаются в нижней части **конструктора Windows Forms** в виде значков.



4. Щелкните значок **openFileDialog1** и задайте два свойства:

- Установите для свойства **Filter** следующее значение:

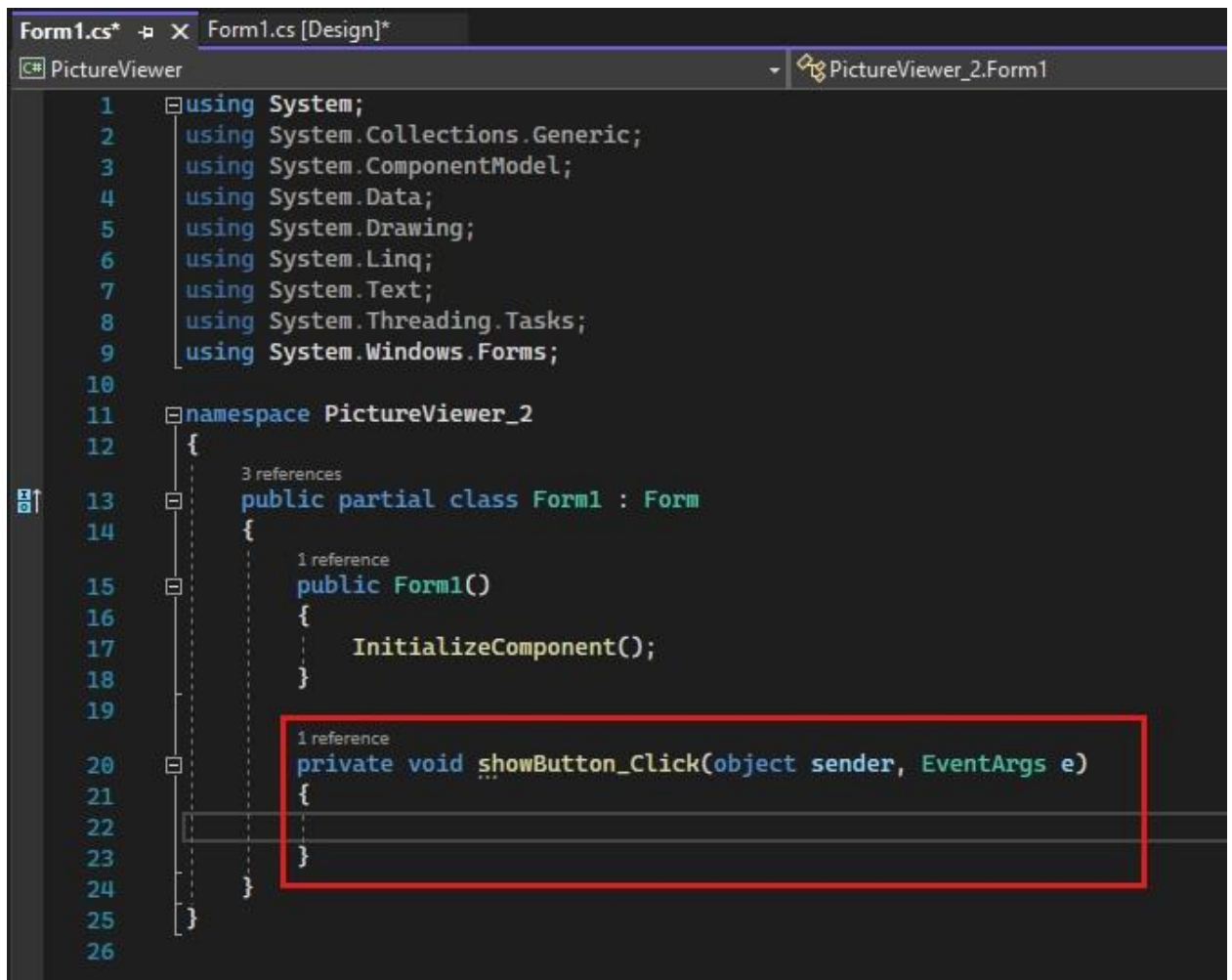
```
JPEG Files (*.jpg)|*.jpg|PNG Files (*.png)|*.png|BMP Files (*.bmp)|*.bmp|All files (*.*)|*.*
```

- Установите для свойства **Title** значение **Выбор файла изображения**.

Параметры свойства **Filter** определяют типы файлов, которые отображаются в диалоговом окне **Выбор файла изображения**.

Добавление обработчиков событий для элементов управления

1. В **конструкторе Windows Forms** дважды щелкните. **Показать рисунок**. Вместо этого можно нажать кнопку **Показать рисунок** в форме, а затем нажать клавишу **ВВОД**. В Visual Studio IDE откроется вкладка в главном окне. Для C# вкладка называется **Form1.cs**. На этой вкладке отображается файл кода, лежащий в основе формы.



На вкладке *Form1.cs* кнопка *showButton* может отображаться как *ShowButton*.

2. Обратите внимание на эту часть кода.

```

private void ShowButton_Click(object sender, EventArgs e)
{
}

```

3. Снова выберите вкладку **Конструктор Windows Forms**, а затем дважды нажмите кнопку **Очистить изображение**, чтобы открыть код. Повторите это действие для двух оставшихся кнопок. Каждый раз при этом действии Visual Studio IDE добавляет в файл кода формы новый метод.

4. Дважды щелкните элемент управления **CheckBox** в **конструкторе Windows Forms**, чтобы добавить метод `checkBox1_CheckedChanged()`. Если установить или снять флажок, будет вызываться этот метод. Ниже показан фрагмент нового кода, который представлен в редакторе кода.

```

private void clearButton_Click(object sender, EventArgs e)
{
}
private void backgroundButton_Click(object sender, EventArgs e)
{
}
private void closeButton_Click(object sender, EventArgs e)
{
}
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
}

```

Методам, включая обработчики событий, можно присвоить любое требуемое имя. При добавлении обработчика событий с помощью интегрированной среды разработки она создает имя на основе имени элемента управления и обрабатываемого события.

Например, событие *Click* для кнопки с именем **showButton** вызывает `showButton_Click()` или `ShowButton_Click()`. Если вы решите изменить имя переменной кода, щелкните правой кнопкой мыши переменную в коде, а затем выберите команду **Рефакторинг > Переименовать**. Все экземпляры этой переменной в коде будут переименованы.

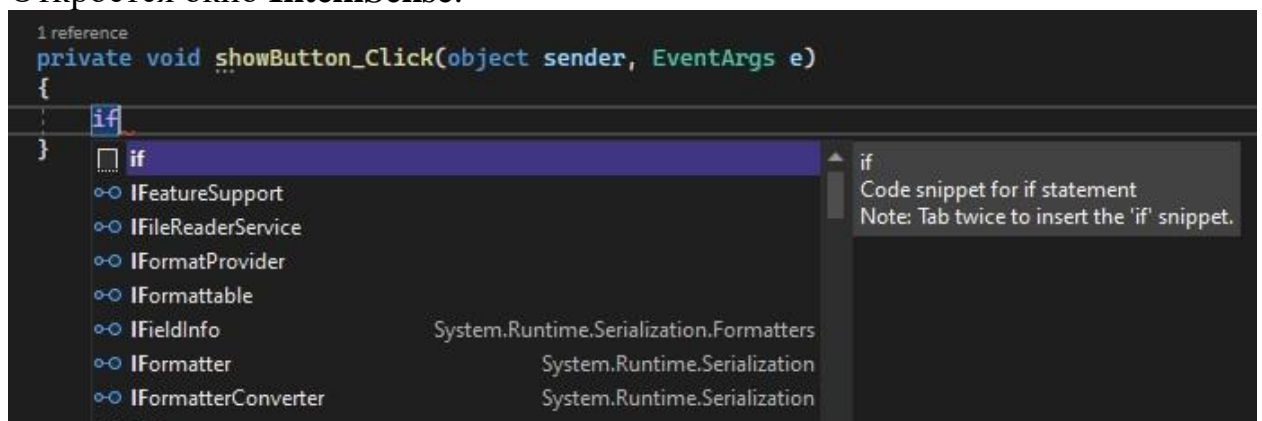
Написание кода для открытия диалогового окна

Кнопка **Показать рисунок** использует компонент `OpenFileDialog` для отображения файла изображения. Эта процедура добавляет код, используемый для вызова данного компонента.

В Visual Studio IDE имеется эффективный инструмент, который называется *IntelliSense*. При вводе текста *IntelliSense* предлагает возможный код.

1. В конструкторе **Windows Forms** дважды нажмите кнопку **Показать рисунок**. Интегрированная среда разработки перемещает курсор внутрь метода `showButton_Click()` или `ShowButton_Click()`.

2. Введите *i* в пустой строке между двумя фигурными скобками { }. Откроется окно **IntelliSense**.



3. Окно **IntelliSense** должно выделить слово `if`. Нажмите клавишу **TAB**, чтобы вставить `if`.

4. Выберите значение **true**, а затем введите **or**, чтобы перезаписать его для **C#**.

```
private void showButton_Click(object sender, EventArgs e)
{
    if(true)
    {
    }
}
```

В IntelliSense отобразится **openFileDialog1**.

5. Нажмите клавишу **TAB**, чтобы добавить **openFileDialog1**.

6. Введите точку (.) или *точку*, сразу после **openFileDialog1**. IntelliSense отобразит все свойства и методы компонента **OpenFileDialog**. Начните вводить **ShowDialog** и нажмите клавишу **TAB**. Метод **ShowDialog()** отобразит диалоговое окно **Открыть файл**.

7. Добавьте круглые скобки сразу после "g" в **ShowDialog: ()**. Код должен быть следующим: **openFileDialog1.ShowDialog()**.

8. В **C#** добавьте пробел, затем два знака равенства (**==**).

9. Добавьте еще один пробел. Используйте IntelliSense для ввода **DialogResult**.

10. Чтобы открыть значение **DialogResult** в окне **IntelliSense**, введите точку. Введите букву **O** и нажмите клавишу **TAB**, чтобы вставить **OK**.

Первая строка кода должна быть завершена. В C# это выглядит следующим образом.

if (openFileDialog1.ShowDialog() == DialogResult.OK

11. Добавьте следующую строку кода:

```
pictureBox1.Load(openFileDialog1.FileName);
```

Можно скопировать и вставить ее или использовать IntelliSense. Окончательный вариант метода **showButton_Click()** будет выглядеть аналогично приведенному ниже коду.

```
private void showButton_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        pictureBox1.Load(openFileDialog1.FileName);
    }
}
```

Добавьте следующий комментарий в код.


```
private void showButton_Click(object sender, EventArgs e)
{
    // Show the Open File dialog. If the user clicks OK, load the
    // picture that the user chose.
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        pictureBox1.Load(openFileDialog1.FileName);
    }
}
```

Рекомендуется всегда комментировать код. Комментарии к коду облегчают понимание и обслуживание кода в будущем.

Написание кода для других элементов управления

Если вы запустите приложение сейчас, можно выбрать параметр **Показать рисунок**. Средство просмотра изображений откроет диалоговое окно **Открыть файла**, в котором можно выбрать изображение для показа.

В этом разделе добавьте код для других обработчиков событий.

1. В конструкторе **Windows Forms** дважды нажмите кнопку **Очистить рисунок**. Добавьте код в фигурные скобки.

```
private void clearButton_Click(object sender, EventArgs e)
{
    // Clear the picture.
    pictureBox1.Image = null;
}
```

2. Дважды нажмите кнопку **Задать цвет фона** и добавьте код в фигурные скобки.

```
private void closeButton_Click(object sender, EventArgs e)
{
    // Close the form.
    this.Close();
}
```

3. Дважды нажмите кнопку **Закреть** и добавьте код в фигурные скобки.

```
private void closeButton_Click(object sender, EventArgs e)
{
    // Close the form.
    this.Close();
}
```

4. Дважды нажмите кнопку **Растянуть** и добавьте код в фигурные скобки.

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    // If the user selects the Stretch check box,
    // change the PictureBox's
    // SizeMode property to "Stretch". If the user clears
    // the check box, change it to "Normal".
    if (checkBox1.Checked)
        pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
    else
        pictureBox1.SizeMode = PictureBoxSizeMode.Normal;
}
```

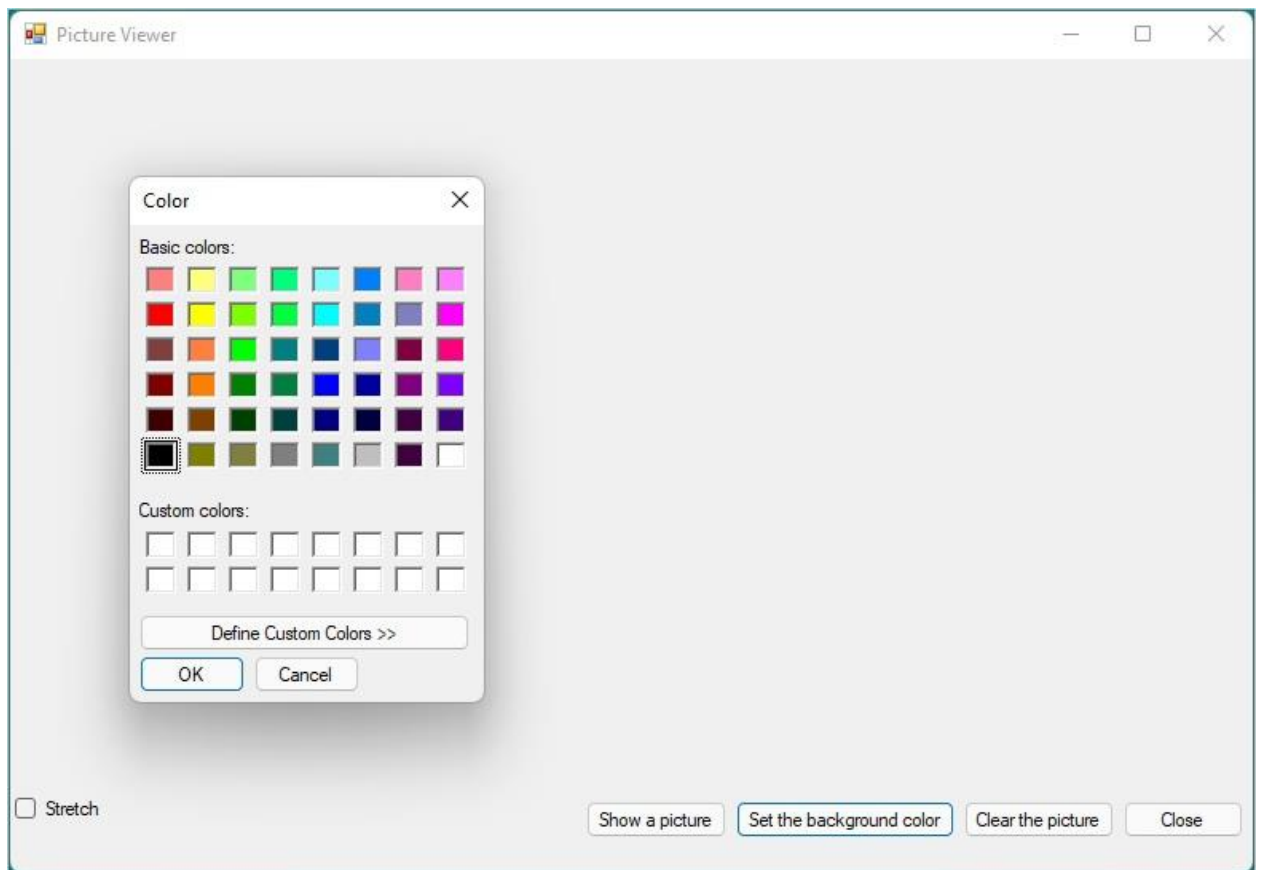
Запуск приложения

Приложение можно запустить в любое время во время его написания. После добавления кода в предыдущем разделе средство просмотра изображений готово. Как и в предыдущих руководствах, используйте один из следующих методов для запуска приложения:

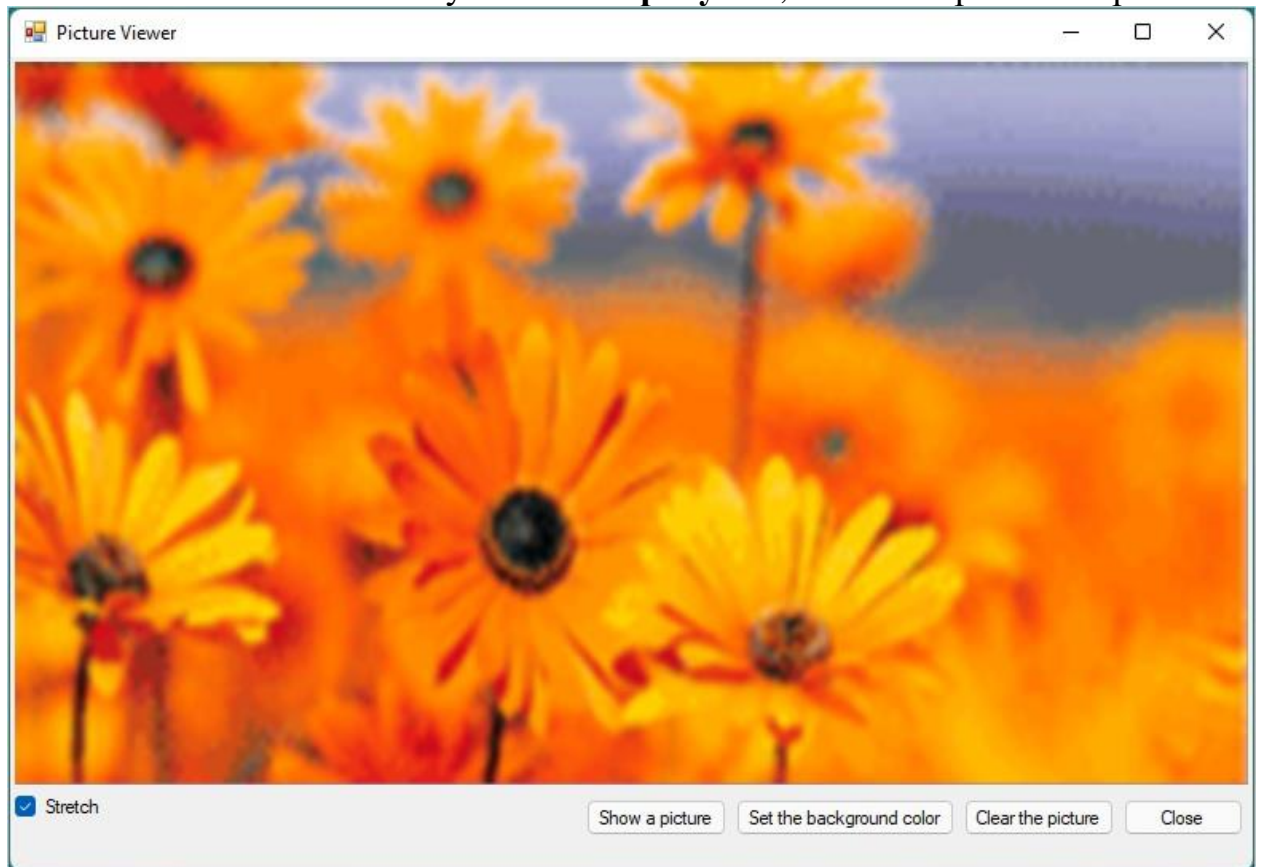
- Нажмите клавишу **F5**.
- В строке меню выберите **Отладка>Начать отладку**.
- На панели инструментов нажмите кнопку **Запустить**.

Появится окно с названием **Средство просмотра изображений**. Протестируйте все элементы управления.

1. Нажмите кнопку **Задать цвет фона**. Откроется диалоговое окно **Цвет**.



2. Выберите цвет для задания цвета фона.
3. Нажмите кнопку **Показать рисунок**, чтобы открыть изображение.



4. Установите и снимите флажок **Растянуть**.

5. Нажмите кнопку **Очистить рисунок**, чтобы очистить окно просмотра.
6. Нажмите кнопку **Заккрыть**, чтобы выйти из приложения