**ПРАКТИЧЕСКАЯ РАБОТА №20**

Изучение базовых событий визуальных компонентов оконных приложений на языкеC#.

* **ЦЕЛЬ РАБОТЫ**

Изучить визуальные компоненты оконных приложений и их события*.*

* **ОБЕСПЕЧЕННОСТЬ ЗАНЯТИЯ**

Оборудование лаборатории и рабочих мест лаборатории:

компьютеры, принтер, сканер, проектор, программное обеспечение общего и профессионального назначения, комплект учебно-методической документации.

Реализация профессионального модуля предполагает обязательную учебную практику.

Оборудование и технологическое оснащение рабочих мест:

1.Комплект ТС компьютера IBM-PC

2.Методические указания для выполнения практических работ

4. Microsoft Visio.

5. Microsoft Visual Studio.

6.Microsoft Office.

* **ЛИТЕРАТУРА**

Хейлсберг А. Язык программирования C#. / Хейлсберг А., Торгерсен М., Вилтамут С., Голд П. Классика Computers Science. 4-е изд. — СПб.: Питер, 2012. — 784 с.

* **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.**

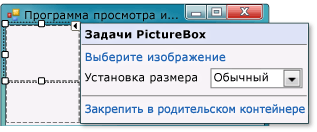
Откройте предыдущий проект.

Перейдите на вкладку "Панель элементов" (находящуюся в левой части интегрированной среды разработки Visual Studio) и разверните группу **Стандартные элементы управления**. В результате этого действия отображается большая часть стандартных элементов управления, которые можно увидеть в формах.

Выберите элемент управления TableLayoutPanel в форме. Чтобы подтвердить, что элемент TableLayoutPanel выбран, убедитесь, что его имя отображается в раскрывающемся списке в верхней части окна **Свойства**. Выбирать элементы управления в форме можно также с помощью раскрывающегося списка в верхней части окна **Свойства**. Выбирать элементы управления таким образом зачастую проще, чем выбрать крошечный элемент управления с помощью мыши.

Дважды щелкните элемент **PictureBox**, чтобы добавить в форму элемент управления PictureBox. Поскольку элемент управления TableLayoutPanel закреплен так, чтобы заполнять собой форму, интегрированная среда разработки добавляет элемент управления PictureBox в первую пустую ячейку (левый верхний угол).

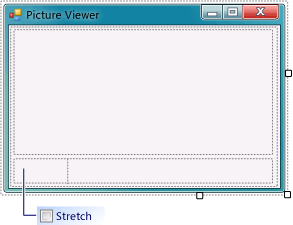
Щелкните новый элемент управления PictureBox, чтобы выбрать его, а затем щелкните черный треугольник на новом элементе управления PictureBox, чтобы отобразить его список задач, как показано на следующем рисунке.

  
Задачи элемента управления PictureBox

Выберите ссылку **Закрепить в родительском контейнере**. В результате этого действия у элемента управления PictureBox свойство **Dock** принимает значение **Fill**. Чтобы это увидеть, выберите элемент управления PictureBox, перейдите к окну **Свойства** и убедитесь, что свойство **Dock** имеет значение **Fill**.

Сделайте так, чтобы элемент управления PictureBox занимал два столбца с помощью его свойства **ColumnSpan**. Выберите элемент управления PictureBox и установите для его свойства **ColumnSpan** значение **2**. Также необходимо, чтобы когда элемент управления PictureBox был пустым, отображалась пустая рамка. Установите для его свойства **BorderStyle** значение **Fixed3D**.

Выберите элемент управления TableLayoutPanel в форме и добавьте в форму элемент управления **CheckBox**. Двойным щелчком выберите элемент **CheckBox** на панели элементов, чтобы добавить новый элемент управления CheckBox в следующую свободную ячейку таблицы. Поскольку элемент управления PictureBox занимает первые две ячейки в TableLayoutPanel, элемент управления CheckBox добавляется в нижнюю левую ячейку. Выберите свойство **Text** и введите слово Stretch, как показано на следующем рисунке.

  
Элемент управления TextBox со свойством Stretch

Выберите в форме элемент управления TableLayoutPanel, а затем на панели элементов перейдите к группе **Контейнеры** (из которой был взят элемент управления TableLayoutPanel) и дважды щелкните элемент управления **FlowLayoutPanel**, чтобы добавить новый элемент управления в последнюю ячейку в элементе управления PictureBox (справа внизу). Затем закрепите FlowLayoutPanel в TableLayoutPanel (выбрав **Закрепить в родительском контейнере** в списке задач FlowLayoutPanel, раскрываемом с помощью черного треугольника, или установив свойству **Dock** FlowLayoutPanel значение **Fill**).

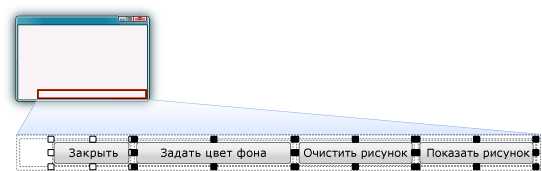
Выберите только что добавленный элемент управления FlowLayoutPanel. Перейдите к группе **Стандартные элементы управления** на панели элементов и двойным щелчком выберите элемент **Кнопка**, чтобы добавить в элемент управления FlowLayoutPanel кнопку с именем **button1**. Чтобы добавить другую кнопку, повторите это действие. Среда интегрированной разработки определяет, что уже существует кнопка с именем **button1** и называет следующую кнопку как **button2**.

Обычно другие кнопки добавляются при помощи панели элементов. На этот раз выберите **button2**, затем в строке меню выберите **Правка**, **Копировать** (или нажмите сочетание клавиш CTRL+C). В строке меню выберите **Правка**, **Вставить** (или нажмите сочетание клавиш CTRL+V), чтобы вставить копию кнопки. Повторите вставку еще раз. Интегрированная среда разработки добавила кнопки **button3** и **button4** в FlowLayoutPanel.

Выберите первую кнопку и установите для ее свойства **Text** значение "Показать рисунок". Затем установите для свойства **Text** следующих трех кнопок значения "Очистить рисунок", "Установить цвет фона" и "Закрыть".

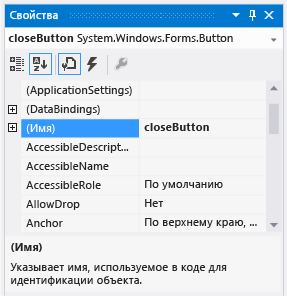
Следующий шаг — задать размер кнопок и разместить их таким образом, чтобы они были выровнены по правой стороне панели. Выберите элемент управления FlowLayoutPanel и обратите внимание на его свойство **FlowDirection**. Измените его значение на **RightToLeft**. После этого действия кнопки должны сами выровняться по правой стороне ячейки и изменить свой порядок таким образом, чтобы кнопка **Показать рисунок** располагалась с правой стороны.

Щелкните кнопку **Закрыть**, чтобы выбрать ее. Удерживая клавишу CTRL, щелкните три другие кнопки, чтобы все они были выбраны. При выделенных кнопках перейдите к окну **Свойства** и прокрутите его вверх до свойства **AutoSize**. Это свойство указывает кнопке автоматически изменять свой размер так, чтобы весь текст мог разместиться на ней. Задайте значение **true**. Кнопки теперь должны иметь соответствующий размер и быть расположены в правильном порядке. (пока выделены все четыре кнопки, можно одновременно изменить все четыре свойства **AutoSize**). На следующем рисунке показаны эти четыре кнопки.

  
Программа просмотра изображений с четырьмя кнопками

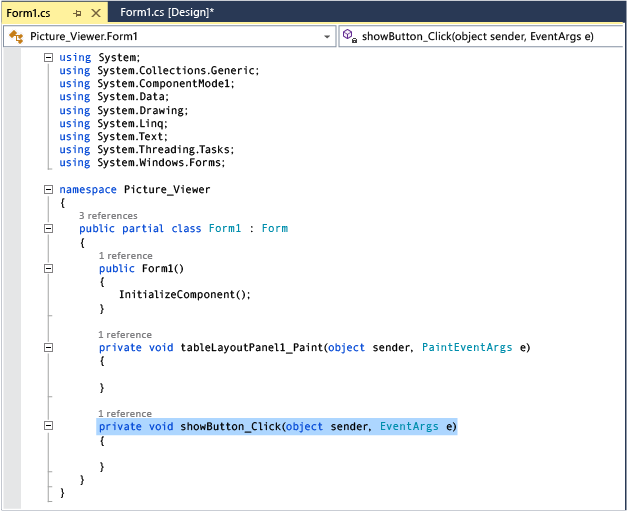
Теперь снова запустите программу, чтобы увидеть обновленную компоновку формы. При нажатии кнопок и установке флажка пока ничего не происходит, однако вскоре все заработает.

В форме нажмите кнопку **Закрыть**. (Если все еще выделены все кнопки, для отмены выделения нажмите клавишу ESC.) Прокрутите содержимое окна **Свойства**, пока не появиться свойство **(Name)**. Свойство **(Name)** расположено в верхней части, когда свойства расположены в алфавитном порядке. Измените имя на **closeButton**, как показано на рисунке ниже.

  
Окно "Свойства" с именем closeButton

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

Двойным щелчком нажмите кнопку **Показать рисунок** в форме. В качестве альтернативы можно нажать кнопку **Показать рисунок** в форме, а затем нажать клавишу ВВОД. При этом интегрированная среда разработки открывает в главном окне дополнительную вкладку, которая называется **Form1.cs** (**Form1.vb**, если используется Visual Basic). На этой вкладке отображается файл кода для формы, как показано на следующем рисунке.

  
Вкладка Form1.cs с кодом Visual C#

Обратите внимание на эту часть кода. (Откройте вкладку **VB** ниже, если используется Visual Basic для просмотра Visual Basic-версии кода.)

C#

[VB](https://msdn.microsoft.com/ru-ru/library/dd492140.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-1)

private void showButton\_Click(object sender, EventArgs e)

{

}

Вы видите код с именем showButton\_Click(). Интегрированная среда разработки добавила его в код формы при открытии файла кода для кнопки **showButton**. Во время разработки при открытии файла кода для элемента управления в форме для элемента управления создается код, если он еще не существует. Этот код, известный как *метод*, выполняется при запуске программы и использовании этого элемента управления (в данном случае — кнопка **Показать рисунок**).

Снова выберите вкладку конструктора Windows Forms (**Form1.cs [Design]** в Visual C#, **Form1.vb [Design]** в Visual Basic), а затем откройте файл кода для кнопки **Очистить рисунок**, чтобы создать метод для нее в коде формы. Повторите это действие для двух оставшихся кнопок. Каждый раз при этом действии среда интегрированной разработки добавляет в файл кода формы новый метод.

Чтобы добавить еще один метод, откройте файл кода для элемента управления CheckBox в конструкторе Windows Forms, чтобы интегрированная среда разработки создала метод checkBox1\_CheckedChanged(). Этот метод вызывается каждый раз, когда пользователь устанавливает или снимает флажок.

Ниже показан новый код, который представлен в редакторе кода.

C#

[VB](https://msdn.microsoft.com/ru-ru/library/dd492140.aspx?cs-save-lang=1&cs-lang=vb#code-snippet-2)

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)

{

}

Пять методов, которые были добавлены, называются *обработчики событий*, так как программа вызывает их каждый раз, когда происходит событие (например, пользователь нажимает кнопку или устанавливает флажок).

При просмотре кода для элемента управления в интегрированной среде разработки во время разработки Visual Studio добавляет метод обработчика событий для элемента управления, если он не существует. Например, при двойном щелчке по кнопке интегрированная среда разработки добавляет обработчик события Click, который вызывается каждый раз, когда пользователь нажимает кнопку. Если дважды щелкнуть флажок, интегрированная среда разработки добавляет обработчик события CheckedChanged, который вызывается каждый раз, когда пользователь устанавливает или снимает флажок.

После добавления обработчика событий для элемента управления к нему можно вернуться в любой момент из конструктора Windows Forms с помощью двойного щелчка по элементу управления или путем выбора пунктов **Вид**, **Код** в строке меню.

Имена являются важными при выполнении построения программы, и методы (включая обработчики событий) могут иметь любые имена, которые нужны. 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 метод обработчика событий showButton\_Click(). Также обычно после имени метода добавляются открывающая и закрывающая круглые скобки () для индикации того, какие методы рассматриваются. Если вы примете решение изменить имя переменной кода, щелкните правой кнопкой мыши по переменной в коде, а затем выберите команду **Рефакторинг**, **Переименовать**. Все экземпляры этой переменной в коде будут переименованы.

* **ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ РАБОТЫ**

1. Изучите методические указания и конспект лекций.
2. Проанализируйте задание по своему варианту.
3. Создайте приложение, отвечающее запросу задания.

* **МЕТОДИКА АНАЛИЗА РЕЗУЛЬТАТОВ, ОБРАЗЕЦ ОТЧЕТА.**

1. Отчет должен содержать цель работы.
2. Содержание индивидуального задания.
3. Перечень элементов, выбранных для создания приложения.