

Лабораторная работа №13

**ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ
АВТОМОБИЛЬНО-ДОРОЖНЫЙ УНИВЕРСИТЕТ**

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И МЕХАТРОНИКИ

Кафедра информационных технологий и мехатроники

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по проведению лабораторных работ по дисциплине
«Объектно-ориентированное программирование»
для студентов специальности 6.050101 «Компьютерные науки»

Разработчик - доцент кафедры информационных технологий и мехатроники
кандидат технических наук, старший научный сотрудник
Тимонин Владимир Алексеевич

Харків 2015

Лабораторная работа №13

Исследование возможностей ИСР Visual C# для создания приложений с использованием панелей инструментов.

Цель работы – исследовать возможностей интегрированной среды разработки Visual C# и получить практические навыки по созданию приложений с использованием панелей инструментов.

1. Теоретические сведения

На сегодняшний день трудно встретить приложение Windows, не имеющее панели инструментов. Как правило, кнопки на панели инструментов копируют команды некоторых разделов меню. На панель инструментов выносятся наиболее часто используемые кнопки, для того чтобы не тратить время на выбор этой команды из пункта меню. Панелей инструментов в одном окне может быть несколько, они могут располагаться в любой части окна, их можно передвигать и скрывать.

Панели инструментов реализуются при помощи элементов управления **ToolStrip**. Элемент управления **ToolStrip** — контейнерный элемент, используемый при создании панели инструментов, главного меню и строки состояния. Элемент управления **ToolStrip** используется непосредственно для панелей инструментов и служит базовым классом для элементов **MenuStrip** и **StatusStrip**, представляющих главное меню и строку состояния.

1.1. Компонент ToolStrip

Компонент **ToolStrip** предназначен для создания инструментальных панелей, содержащих кнопки, текстовые поля, списки и некоторые другие часто используемые интерфейсные компоненты.

Для создания панели инструментов необходимо разместить компонент **ToolStrip** на форме. После того как в форму будет добавлен компонент **ToolStrip**, в верхней части формы появляется строка инструментов (синяя полоса), в начале которой находится специализированная кнопка, которая существует виртуально, и содержит небольшую стрелку для вызова выпадающего меню. В правой части расположена кнопка, вызывающая выпадающее меню «ToolStrip Задачи», с помощью которого вставлять и править элементы панели инструментов. Значок компонента располагается в нижней части рабочего стола (на специальной полосе под формой), а не в самой форме (рис. 1).

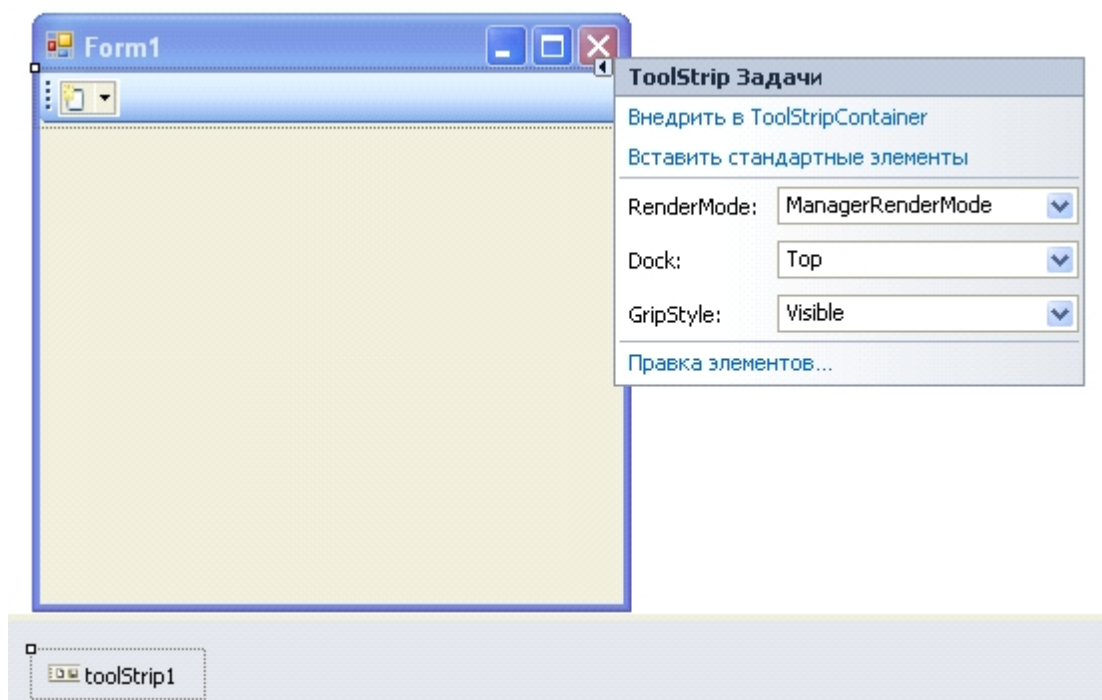


Рис. 1. Внешний вид компонента ToolStrip в конструкторе

Чтобы создать инструментальную кнопку класса ToolStripButton, можно сделать щелчок на специализированной кнопке. Если вызвать выпадающее меню, то можно выбрать тип создаваемого компонента на поверхности панели (рис 2).

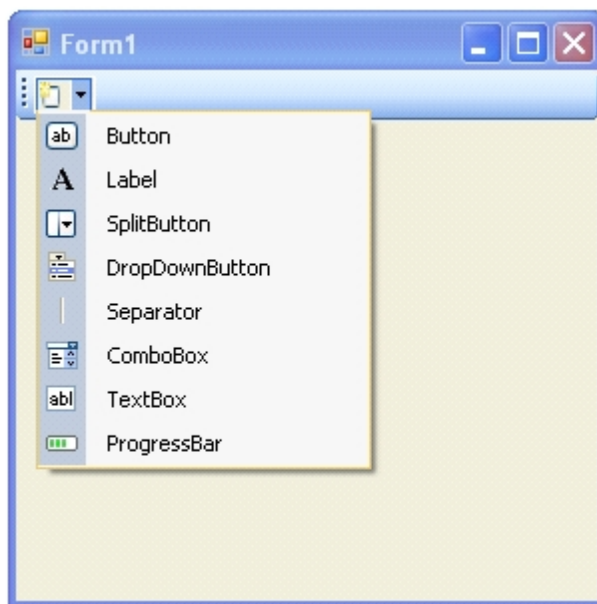


Рис. 2. Перечень элементов панели инструментов

Для добавления доступны следующие компоненты:

- **Button** – обычная инструментальная кнопка;
- **Label** – надпись (поле с текстом);
- **SplitButton** – кнопка с возможностью вызова меню;
- **DropDownButton** – кнопка с возможностью вызова списка выбора;
- **Separator** – разделитель компонентов на панели **ToolStrip**;
- **ComboBox** – список выбора;
- **TextBox** – текстовое поле;
- **ProgressBar** – индикатор процесса.

Формирование структуры панели инструментов может происходить с помощью свойства **Items** компонента **ToolStrip** путем нажатия кнопки с многоточием рядом со свойством **Items** в окне «Свойства». В результате откроется окно «Редактор коллекции элементов». Редактор обеспечивает прямой доступ ко всем свойствам вложенного элемента управления, а также позволяет редактировать, удалять и переупорядочивать элементы внутри полосы состояния.

В редакторе с помощью списка «Выбрать элемент и добавить в список ниже» выбирается тип добавляемого компонента (рис. 3), щелчком на кнопке «Добавить» выбранный компонент помещается на инструментальную панель. Щелчок на кнопке «Remove» (на кнопке изображены две перекрещивающиеся линии) удаляет текущий компонент.

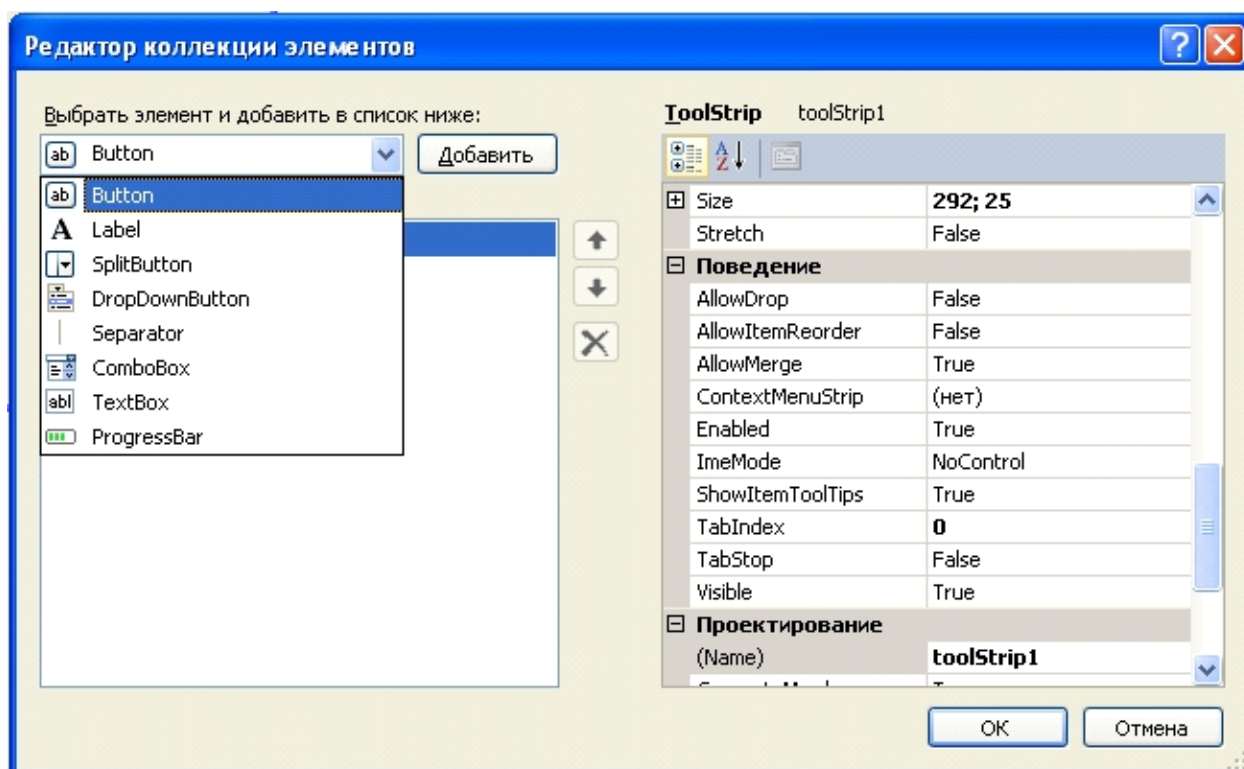


Рис. 3. Вид диалогового окна «Редактор коллекции элементов»

При добавлении нового пункта меню необходимо задать название пункта меню (свойство **Text**) и имя (**Name**) элемента **ToolStripButton**. При необходимости можно задавать и другие свойства элемента **ToolStripButton** с помощью окна «Свойства» (раздел **ToolStripButton**) (рис. 4).

Для формирования структуры меню можно воспользоваться диалоговым окном **MenuStrip Задачи**, которое раскрывается с помощью щелчка на кнопке в виде квадрата с черных треугольников в верхнем правом углу компонента **ToolStrip** (см. рис. 1). Это диалоговое окно обеспечивает доступ к типичным командам и свойствам:

- **Внедрить в ToolStripContainer** — позволяет (щелчком мыши) поместить панель инструментов в специальный контейнер (вместо расположения его в форме).
- **Вставить стандартные элементы** — добавляет общепринятые элементы панели инструментов.

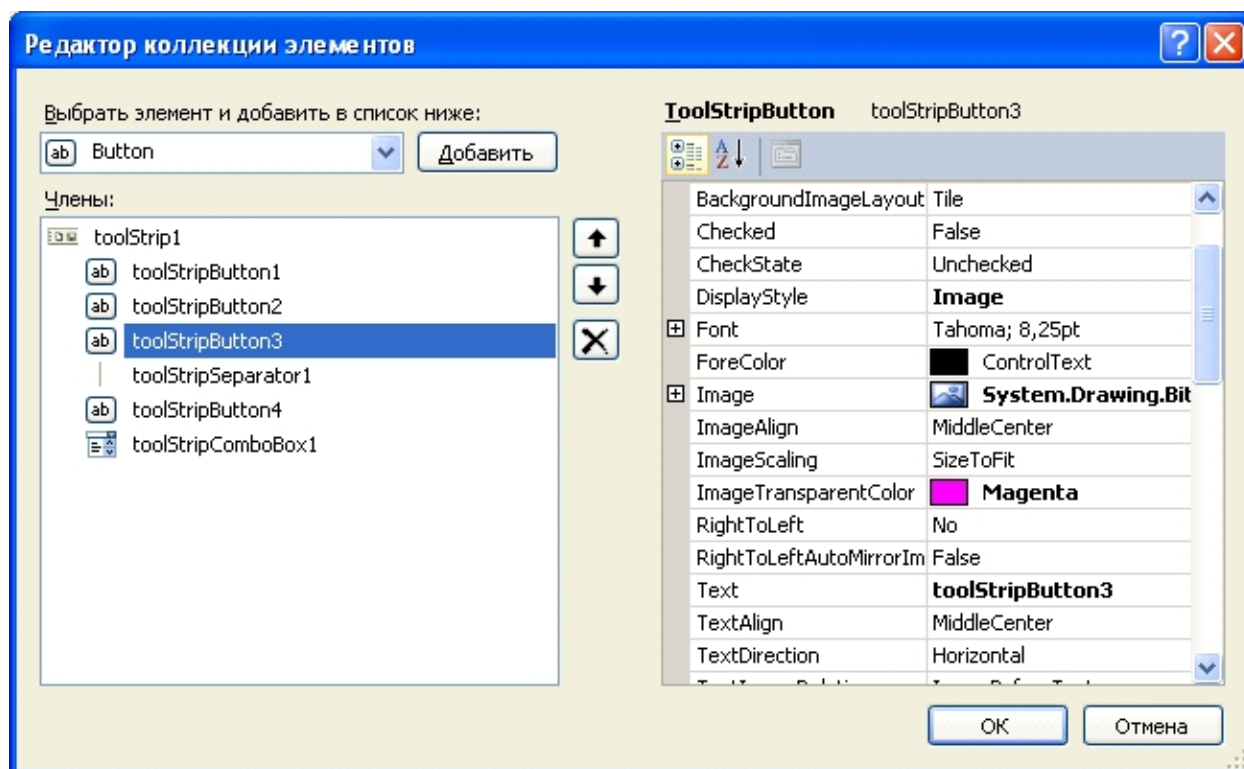


Рис. 4. Вид диалогового окна «Редактор коллекции элементов» при создании панели

- **RenderMode** — опция, которая дает возможность выбора из выпадающего списка способа изображения панели инструментов: системного (**System**), профессионального (**Professional**) или управляемого (**ManagerRenderMode**).

- **Dock** — выводит (по щелчку на кнопке) в поле этой опции схему причаливания панели инструментов к той или иной стороне формы.

- **GripStyle** — в этой опции существует выпадающий список, задающий элемент стиля полосы панели инструментов: невидима или видима будет специальная пунктирная канавка в верхней части полосы.

- **Правка элементов** — с помощью этой опции и задаются элементы панели инструментов. Если щелкнуть на этой опции, то откроется диалоговое окно для задания элементов панели инструментов.

После того как структура меню будет сформирована, можно выполнить настройку панели инструментов. Настройка выполняется путем изменения значений свойств компонента **ToolStrip** (свойства компонента **ToolStrip** приведены в табл. 1.) и свойств пунктов меню (объектов типа **ToolStripItem**). Свойства объекта **ToolStripItem** приведены в табл. 2.

Таблица 1. Свойства компонента ToolStrip

Свойство	Описание
Name	Имя элемента
BackgroundImage	Задаёт (с помощью выбора через диалоговое окно) фоновое изображение, которое помещается на элементе управления
Dock	Определяет схему причаливания панелей управления к той или иной стороне формы
Enabled	Определяет доступность панели инструментов
Items	Через диалоговое окно этого свойства формируются элементы панели инструментов.
LayoutStyle	Стиль размещения панели инструментов. Выбирается из выпадающего списка.
Size	Устанавливает размер
Visible	Позволяет скрыть элемент (при значении равном false)

Таблица 2. Свойства объекта ToolStripItem

Свойство	Описание
Name	Имя элемента, которое образуется из названия элемента и названия объекта ToolStripItem , например, ToolStripButton1
Text	Название элемента панели инструментов
Image	Картинка, которая отображается на элементе
Enabled	Признак доступности элемента панели. Если значение свойства равно false , то элемент панели недоступен (в результате щелчка на элементе меню событие Click не происходит, название элемента меню отображается инверсным, по отношению к доступному пункту меню, цветом)
Checked	Признак того, что элемент панели выбран. Если значение свойства равно true , то элемент выделяется оранжевым цветом.
CheckState	Устанавливает трех видовое состояние: элемент выбран – Checked , элемент не выбран – Unchecked , состояние неопределенности – Indeterminate .
DisplayStyle	Задаёт значение, определяющее содержание внешнего вида элемента(None – на элементе ничего не отображается, Text – отображается текст, Image – отображается рисунок, ImageAndText – отображается рисунок и текст). Когда AutoSize установлено в true , объект ToolStripItem будет изменять свой размер, потому что потребуется минимальное пространство.
TextDirection	Устанавливает ориентацию текста. Значения могут быть любыми из перечисления ToolStripTextDirection (Horizontal , Inherit , Vertical90 – поворачивает текст на 90 градусов, и Vertical270 – поворачивает текст на 270 градусов).

Элемент управления **ToolStrip** использует набор элементов управления, классы которых являются производными от класса **ToolStripItem**. Основные свойства элементов управления **ToolStripItem** — это **Image** и **Text**. Графические изображения могут быть установлены либо через свойство **Image**, либо с использованием элемента управления **ImageList** и установкой свойства **ImageList** элемента управления **ToolStrip**. Затем могут быть установлены свойства **ImageIndex** индивидуальных элементов управления.

Создаваемая кнопка **ToolStripButton** (кнопки такого класса создаются на панели **ToolStrip**) очень схожа по свойствам с кнопкой **Button**, только у нее есть свойства **Checked**, **CheckOnClick** и **CheckState**, как у пункта меню и у компонента **CheckBox**, чтобы кнопку можно было не просто нажимать, но и изменять состояние — утапливать и отпускать. Когда свойство **CheckOnClick** равно **true**, то при нажатии кнопка залипает, меняя состояние **Checked** на **true**, а при повторном нажатии кнопка отпускается, меняя **Checked** на **false**. В противном случае при нажатии кнопка нажимается и тут же отпускается.

Для работы с изображениями в среде разработки существует редактор изображений, который позволяет создать изображения с использованием простейших инструментов. Для создания файла с изображением вам необходимо воспользоваться пунктом меню **Файл → Создать → Файл...** В появившемся окне «Создать файл» (рис. 5) выберите тип файла «Файл точечного рисунка».

Появится пустое изображение с дополнительной панелью управления (рис. 6), а также в основном меню появится новый пункт меню «Изображение».

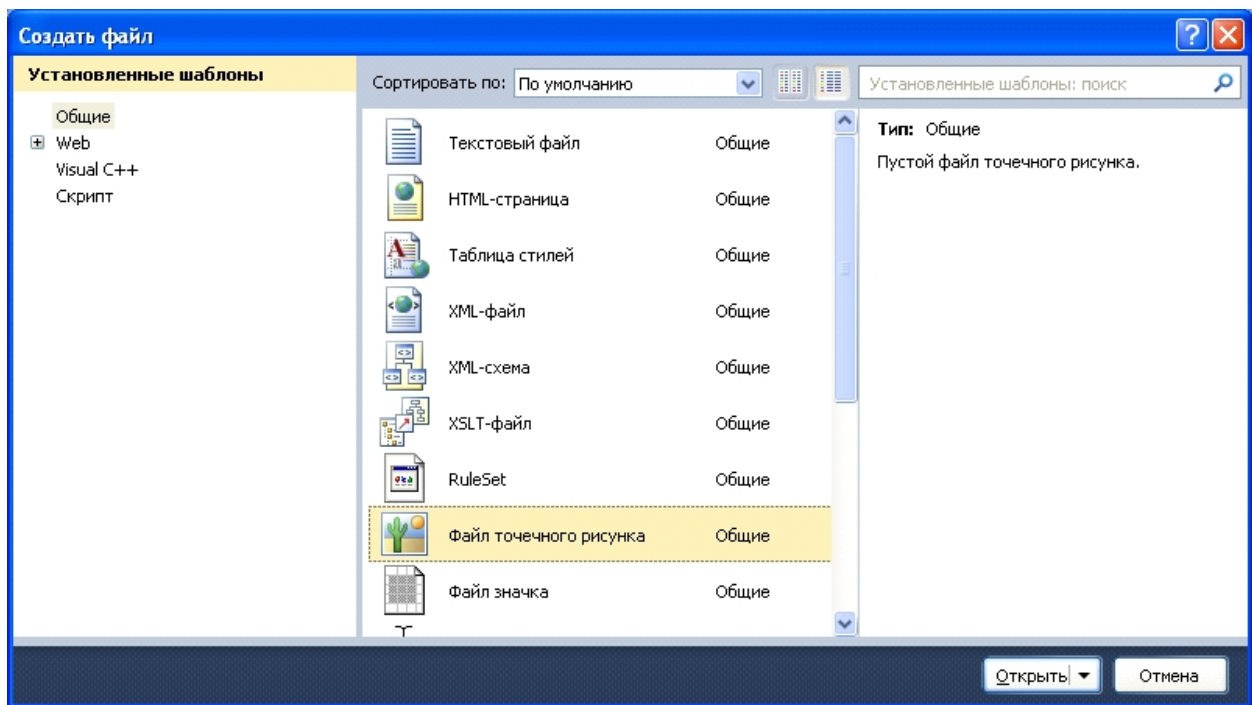


Рис. 5. Внешний вид окна «Создать файл»

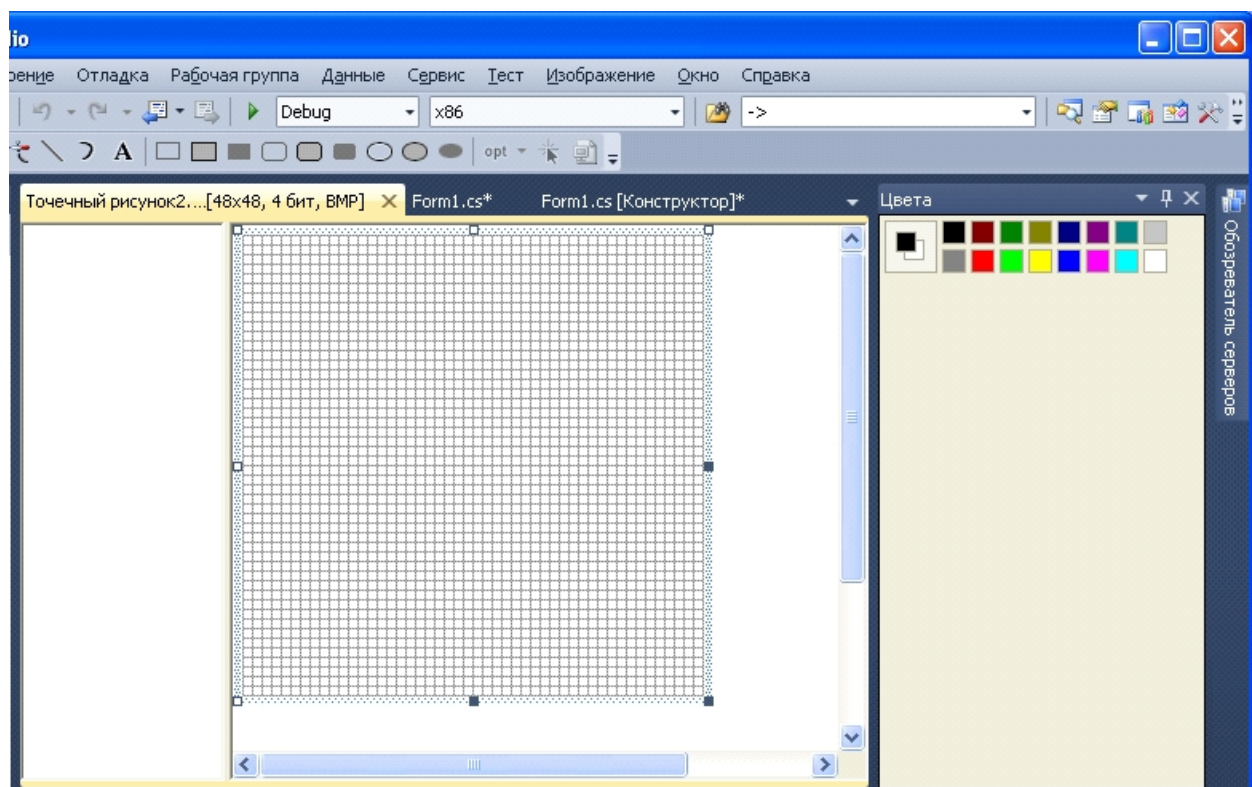


Рис. 6. Внешний вид редактора изображений

Пример 1. Создайте приложение "Панель инструментов", демонстрирующее возможности компонента **ToolStrip**. Для этого на форму разместите компоненты **MenuStrip**, **ToolStrip** и **Label**. Компонент **MenuStrip** должен содержать три пункта («Вопросы» (содержит подпункты «Вопрос1», «Вопрос2», «Вопрос3»), «Ответы» (содержит подпункты «Ответ1», «Ответ2», «Ответ3»), и «Справка» (не содержит не одного подпункта)) (рис. 7).

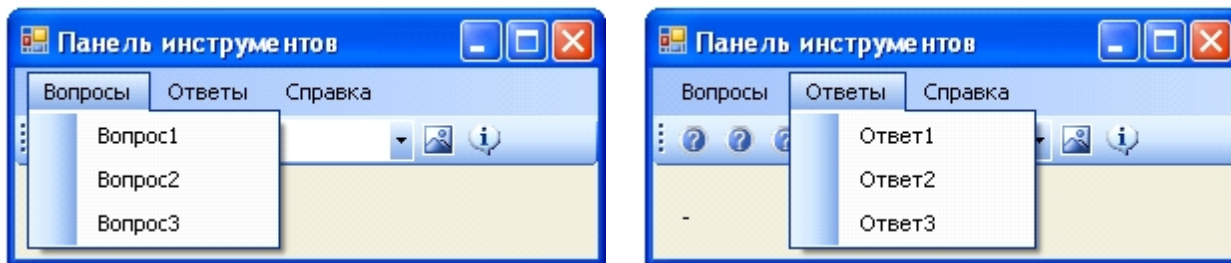


Рис. 7. Внешний вид приложения с перечнем пунктов меню

Компонент **ToolStrip** должен содержать следующие элементы (рис. 8):

- три стандартные кнопки **Button** (1-я кнопка – дублирует действие подпункта «Вопрос1», 2-я кнопка – дублирует действие подпункта «Вопрос2», 3-я кнопка – дублирует действие подпункта «Вопрос3»);
- компонент **ComboBox**, который содержит три ответа на поставленный вопрос и предназначен для выбора правильного ответа. Выбор варианта ответа в компоненте **ComboBox** дублирует действия подпунктов «Ответ1», «Ответ2», «Ответ1»;
- стандартную кнопку **Button**, которая предназначена для выдачи сообщения «Демонстрация компонента ToolStrip»;
- стандартную кнопку **Button**, которая предназначена для выдачи справочной информации (дублирует действие пункта «Справка»).

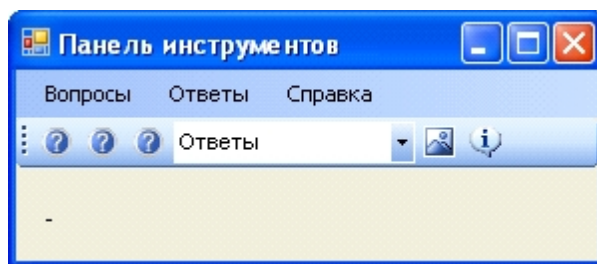


Рис. 8. Внешний вид приложения с перечнем кнопок панели инструментов

Самостоятельно запишите код реализации обработчиков событий **Click** для каждого подпункта меню. При выборе какого-либо подпункта меню должно выдаваться соответствующее сообщение, которое отображается в компоненте **Label**. Например, при выборе подпункта «Вопрос1» выдается сообщение «**Выбран вопрос №1**».

Сделайте привязку кнопок панели инструментов к соответствующим подпунктам меню. Для этого необходимо выбрать кнопку на панели, перейти на панель «Свойства», выбрать перечень событий (нажать кнопку «События»), выбрать соответствующее событие (для нашего примера – это событие **Click**) и в раскрывающемся списке обработчиков событий (рис. 9) выбрать соответствующий обработчик.

Для кнопки «Демонстрация» запишите собственный обработчик события **Click**, так как данная кнопка не связана с каким-либо подпунктом меню.

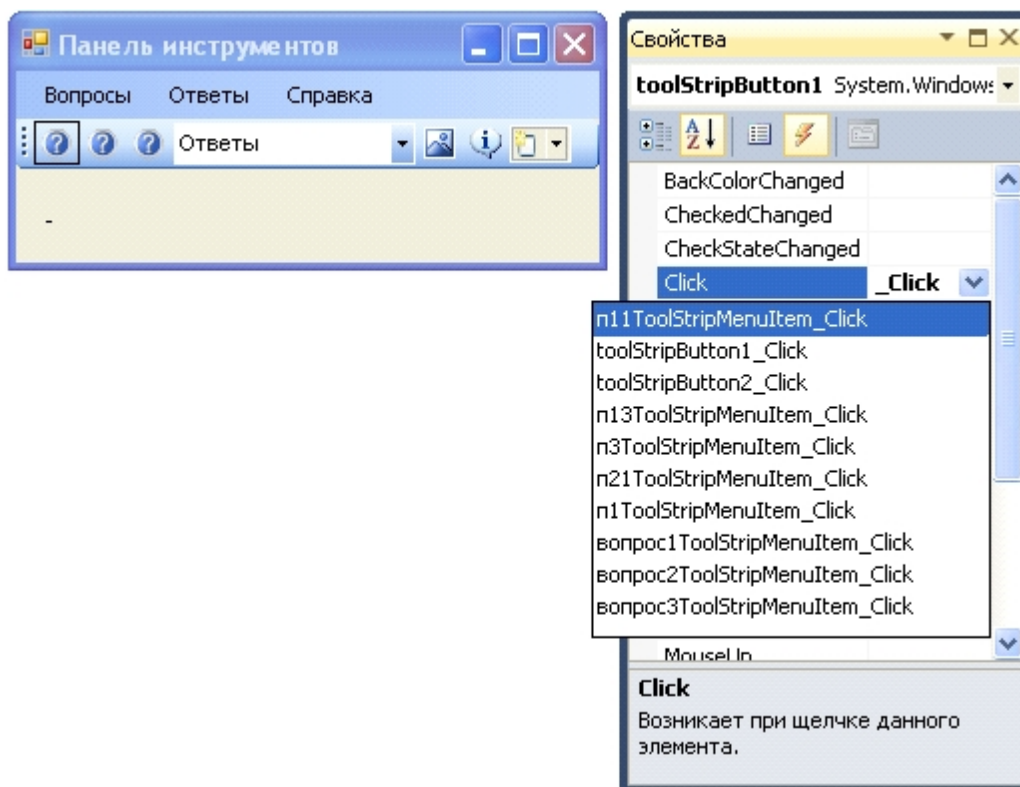


Рис. 9. Перечень обработчиков события Click

Алгоритм работы приложения следующий:

- при запуске приложения пункт меню «**Ответы**» и компонент **ComboBox** должны быть недоступны, остальные элементы должны функционировать;
- при выборе вопроса (с помощью пункта меню или кнопки панели инструментов) разрешается доступ к пункту меню «**Ответы**» и компоненту **ComboBox**, с помощью которых выбирается вариант ответа с выдачей соответствующего сообщения (рис. 10).

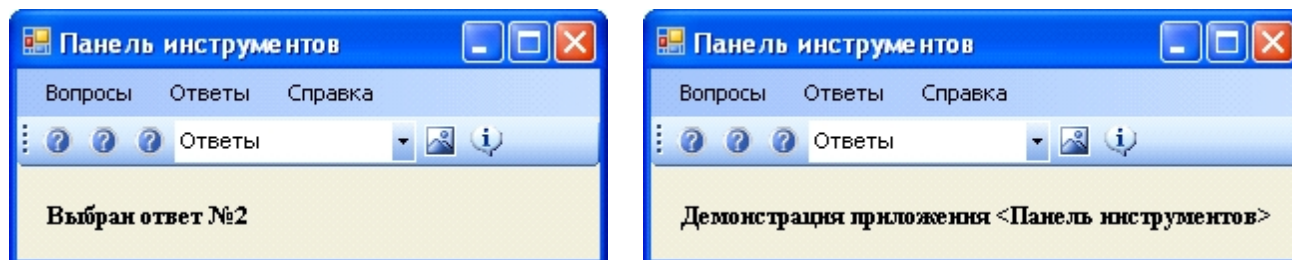


Рис. 10. Варианты выполнения приложения «Панель инструментов»

1.2. Компонент StatusStrip

Строка состояния (компонент **StatusStrip**) предназначена для того, чтобы отображать состояние работы программы. Она даёт пользователю информацию о текущем статусе приложения, о продвижении некоего действия, о выделенном на форме объекте и т. д.

Для создания строки состояния необходимо разместить компонент **StatusStrip** на форме. После того как в форму будет добавлен компонент **StatusStrip**, в нижней части формы появляется строка инструментов (серая полоса), в начале которой находится специализированная кнопка, которая существует виртуально, и содержит небольшую стрелку для вызова выпадающего меню. В правой части расположена кнопка, вызывающая выпадающее меню «**StatusStrip Задачи**», с помощью которого вставлять и править элементы панели инструментов. Значок компонента располагается в нижней части рабочего стола (на специальной полосе под формой), а не в самой форме.

В режиме конструирования при нажатии специализированной кнопки выпадает список, в

котором имеется выбор всех четырех поддерживаемых дочерних элементов управления элемента **StatusStrip** (рис. 11).

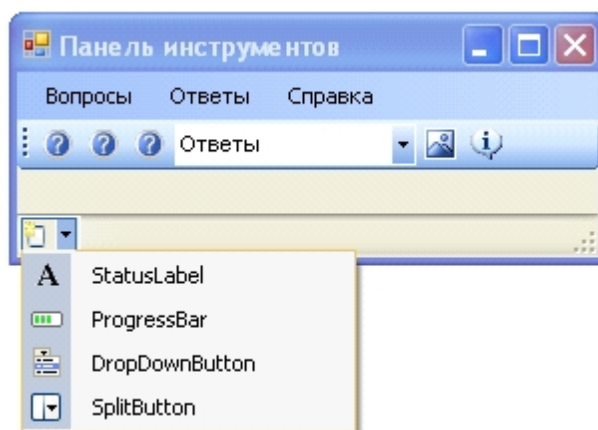


Рис. 11. Перечень элементов строки состояния

По умолчанию в элементе управления **StatusStrip** панели отсутствуют. Дочерние элементы управления можно добавлять внутри **StatusStrip** путем упорядочения потоком слева направо. Возможности редактирования "на месте" очень удобно для быстрого добавления дочерних элементов управления для **StatusStrip**.

Когда компонент строки состояния выделен, то слева появляется небольшая кнопка, по нажатию которой создается метка класса **ToolStripStatusLabel**, похожа на метку **Label** и также предназначена для отображения информации. Если нужно создать компонент другого класса, то можно маленькой кнопкой со стрелкой вызвать всплывающее меню, в котором можно найти дополнительные элементы:

- **StatusLabel** — метка;
- **ProgressBar** — индикатор процесса;
- **StripButton** — кнопка, которая имеет дополнительную кнопку для вызова всплывающего меню. Основная и дополнительная кнопки работают независимо;
- **DropDownButton** — кнопка, которая вызывает всплывающее меню. Всплывающее меню появится именно по нажатию кнопки, без нажатия на какие-либо дополнительные кнопки.

Все они в функциональном плане повторяют соответствующие компоненты панели **ToolStrip**.

Если необходима более детальная настройка дочерних элементов управления, можно использовать диалоговое окно «Редактор коллекции элементов», вызываемое с помощью свойства **Items** компонента **StatusStrip** или выпадающее меню «**StatusStrip Задачи**».

Основное свойство каждого элемента **StatusStrip** — **Text**, в которое заносится отображаемый текст. Его можно занести в процессе проектирования, а затем можно изменять программно во время выполнения. Другое существенное свойство — **Width**, определяющее ширину полосы состояния.

Программный доступ к текстам отдельных элементов можно осуществлять:

- с помощью индексированного свойства **Items** компонента **StatusStrip**. Например, оператор `statusStrip1.Items[0].Text = "текст";` напечатает текст в первом элементе;
- с помощью свойства **Text** элемента **toolStripStatusLabel**. Например, оператор `toolStripStatusLabel2.Text = "текст";` напечатает текст в элементе с именем **toolStripStatusLabel2**.

2. Рабочее задание



Задание 1. Модифицировать приложение «Панель инструментов», добавив строку состояния, в которой должна отображаться информация о названии выбранного вопроса и ответа (один из вариантов интерфейса представлен на рис. 12).

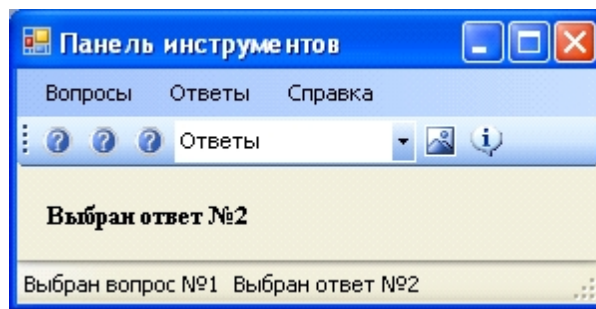


Рис. 12. Результат выполнения модифицированного приложения «Панель инструментов»



Задание 2. Разработать приложение «Действия над матрицей». Данное приложение должно выполнять следующие действия:

- преобразовывать исходную матрицу в диагональную, единичную и транспонированную;
- осуществлять поиск минимального, максимального и элемента с нулевым значением;
- выдавать сведения о разработчике приложения.

Возможные варианты интерфейса представлены на рис. 13, 14, 15.

Ввод значений элементов матрицы осуществляется с клавиатуры.

В результате выполнения преобразований на форме должны отобразиться исходная и преобразованная матрицы, минимальный, максимальный и элемент с нулевым значением, а также в строке состояния необходимо отобразить размерность матрицы и местоположение соответствующего элемента.

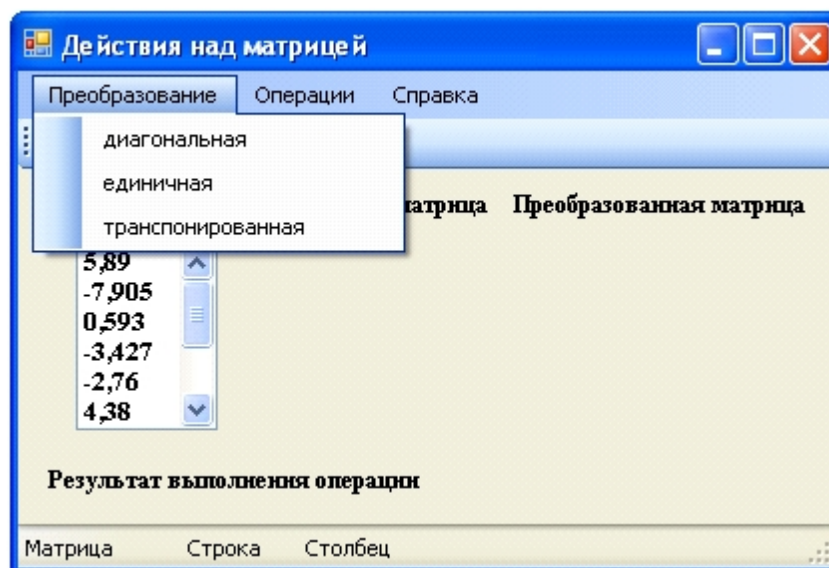


Рис. 13. Внешний вид приложения с раскрытым пунктом «Преобразование»

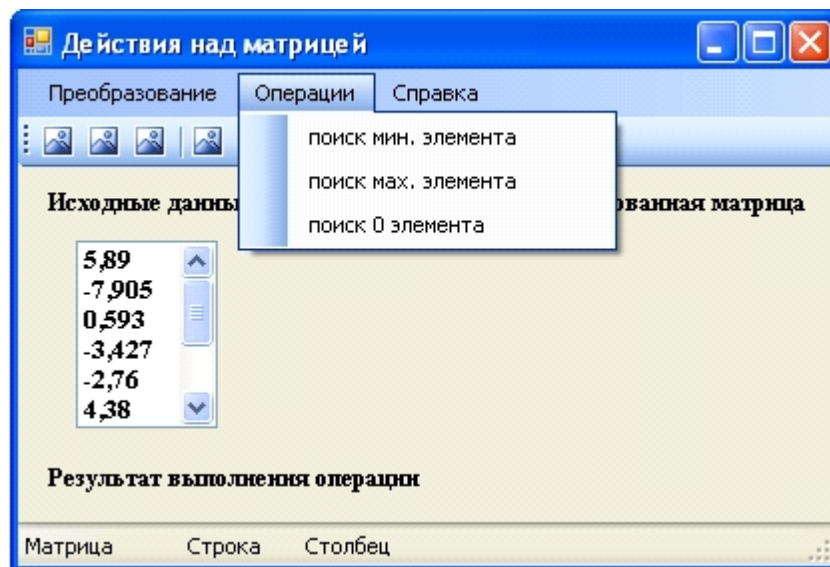


Рис. 14. Внешний вид приложения с раскрытым пунктом «Операции»

В результате выполнения операций на форме должны отобразиться исходная матрица и результат поиска, а также в строке состояния необходимо отобразить размерность матрицы, номер строки и столбца найденного элемента.

При выдаче справки необходимо отобразить информацию о разработчике.

Кнопки панели инструментов должны содержать подсказки об их функциональном назначении (рис. 15).

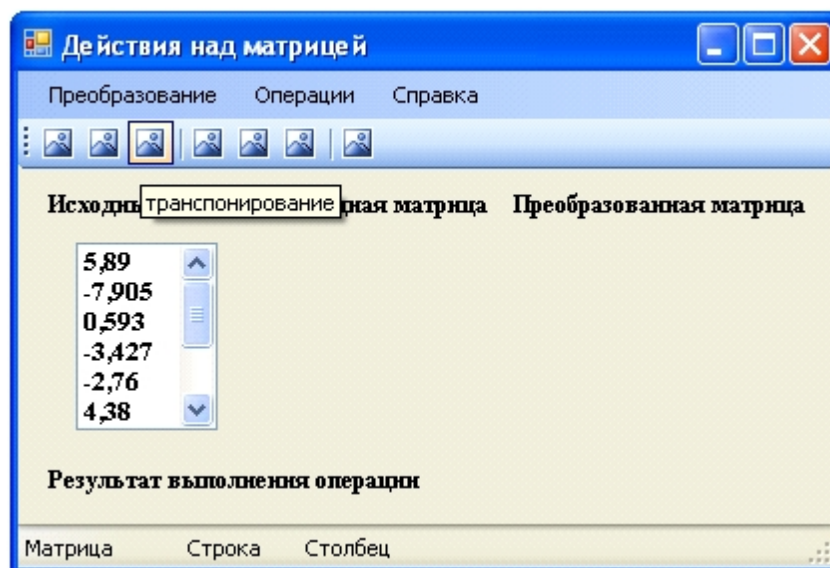


Рис. 15. Внешний вид приложения с текстом назначения кнопки

3. Контрольные вопросы

Литература

1. Голощапов А.Л. Microsoft Visual Studio 2010. – СПб.:БХВ-Петербург, 2011. – 544 с.: ил.
2. Культин Н.Б. Microsoft Visual C# в задачах и примерах. – СПб.: БХВ-Петербург, 2009. – 320 с.: ил.
3. Лабор В.В. Си Шарп: Создание приложений для Windows. – Мн.: Харвест, 2003. – 384 с.
4. Петцольд Ч. Программирование для Microsoft Windows на C#. В 2-х томах. Том 1. Пер. с англ. - М.: «Русская Редакция», 2002.- 576 с.: ил.

5. Петцольд Ч. Программирование для Microsoft Windows на C#. В 2-х томах. Том 2. Пер. с англ. - М.: «Русская Редакция», 2002.- 624 с.: ил.
6. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4.0. Пер. с англ. - М.: Издательский дом "Вильямс", 2011. — 1392 с.: ил.
7. Фаронов В.В. Программирование на языке C#. – СПб.: Питер, 2007. – 240 с.: ил.
8. Фленов М.Е. Библия C#. - СПб.: БХВ-Петербург, 2011. – 560с.: ил.