

Лекция 4 ИСПОЛЬЗОВАНИЕ МЕНЮ В ПРИЛОЖЕНИИ

4.1 Меню программы

Меню программы должно соответствовать основным режимам работы программы, поэтому к выбору пунктов меню и команд отдельных пунктов необходимо относиться с особой тщательностью.

Для лучшего понимания технологии использования меню в программах рассмотрим последовательность действий при решении следующей учебной программы.

Задача 4.1 Разработать приложение, позволяющее создавать матрицу 6*6 случайных целых чисел в диапазоне от 10 до 100. Напечатать эту матрицу с помощью **richTextBox**. Записывать клиентскую область приложения (включая распечатку матрицы) в текстовый файл. Считывать текстовый файл в клиентскую область – печатать содержимое текстового файла. Из содержимого клиентской области выделять матрицу. Все действия оформить с использованием меню.

Таким образом, условно можно выделить три режима работы программы: Матрица (создание и печать), Файл (запись и чтение) и Текст (очистка и преобразование).

Создадим обычное приложение для работы в системе Windows.

Для добавления меню в главное окно нашего приложения необходимо переместить из окна элементов управления Toolbox значок меню с названием MenuStrip. Как только Вы это сделаете, окно форм примет вид, показанный на рисунке 4.1

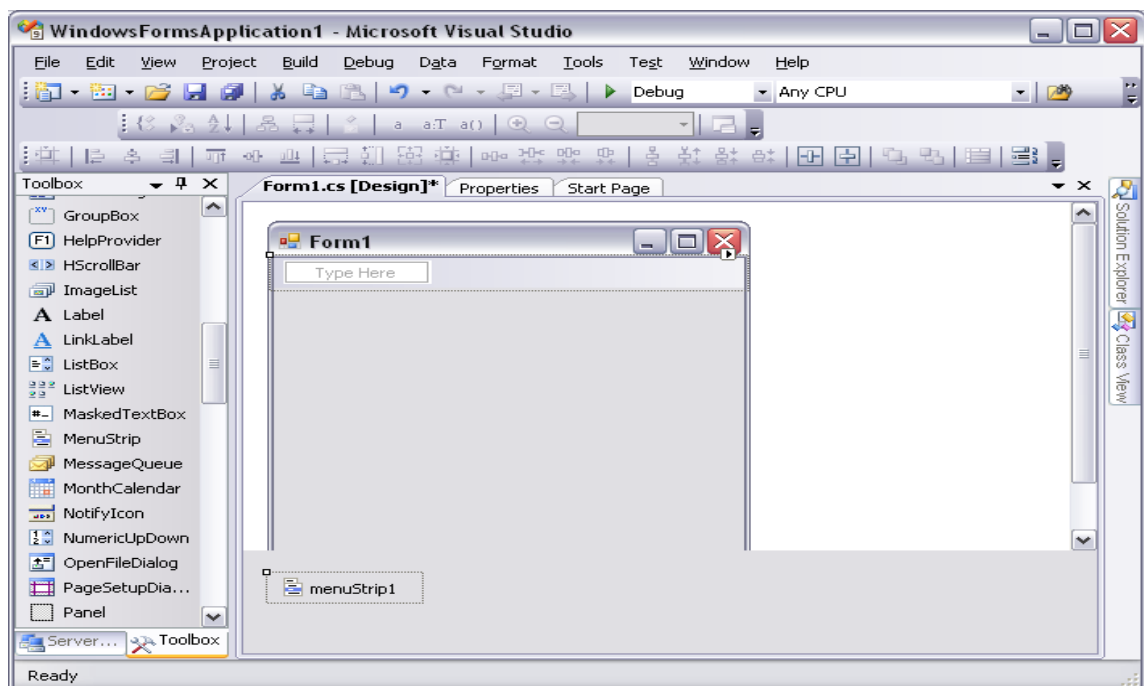


Рисунок 4.1 – Добавление меню

В нижней части этого окна появится значок управляющего элемента — меню `menuStrip1`. Кроме того, непосредственно под заголовком окна появится пустое пока меню, представленное полем с надписью `Type Here` (что можно перевести как «печатать здесь»).

Напечатайте в этом поле строку «&Matric». В результате этого в окне нашего приложения появится меню Matric (рисунок 4.2).

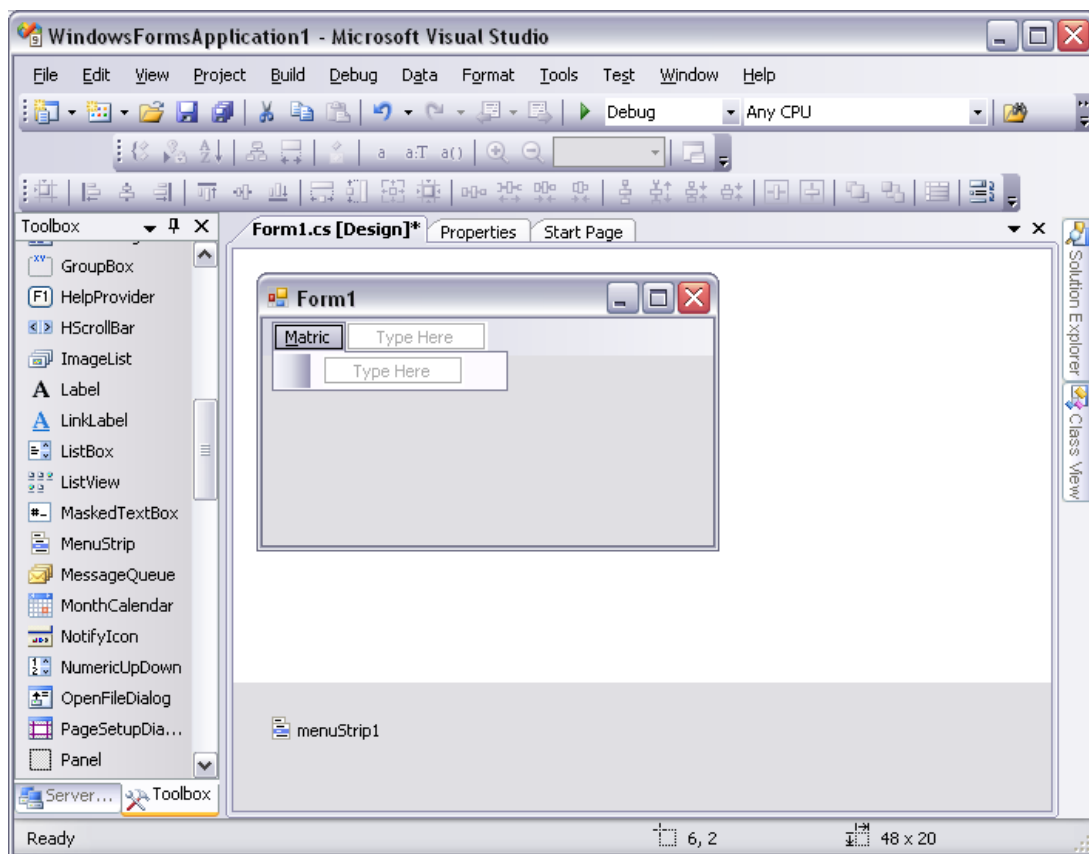


Рисунок 4.2 – Создание меню для работы с матрицей

Ввод символа & позволяет создать клавиатурный акселератор для ввода команды с помощью клавиатуры (Alt+P). Применение акселераторов является альтернативным мышки способом выбора команд. В комбинацию с клавишей Alt включается буква, стоящая за символом амперсанд. В тексте меню программы буквы, включенные в акселератор, отображаются подчеркнутыми.

После ввода режима работы программы поле ввода `Type Here` опустилось вниз, предлагая вводить команды заданного режима. Ведем две команды &SozM и &PrintM. После этого переходим в правое поле ввода `Type Here` для ввода режима работы с файлом. Последним введем режим работы и команды для работы с текстом клиентской области.

Если при вводе Вы допустили ошибку, то ее можно исправить. Для этого нужно кликнуть правой кнопкой мышки в нужной строке меню и в появившемся на экране контекстном меню выбрать необходимый режим редактирования.

С помощью строки Insert New Вы можете вставить новую строку меню между уже существующих строк. Строка Insert Separator предназначена для вставки разделительной линии между строками меню. И, наконец, при помощи строки Edit Names можно отредактировать идентификаторы строк и меню.

Если же Вам нужно изменить введенные названия строк и меню, это можно сделать по месту, выбрав нужную строку мышью.

После этого Вы можете оттранслировать приложение и запустить его, нажав кнопку F5. Убедитесь, что меню отображается, и Вы можете выбирать его строки. Если все нормально, можно переходить к следующему этапу создания нашего приложения.

Работу с текстом в нашем приложении мы будем выполнять с помощью элемента RichTextBox, позволяющего работать с файлами разного типа. Для этого перетащим из окна Toolbox на нашу форму элемент RichTextBox.

Настройте свойства компонента, чтобы он занимал все окно приложения. Далее отредактируйте свойство Dock. Это свойство задает расположение выбранного компонента внутри содержащей его формы – необходимо щелкнуть в центре (см. рисунок 4.3).

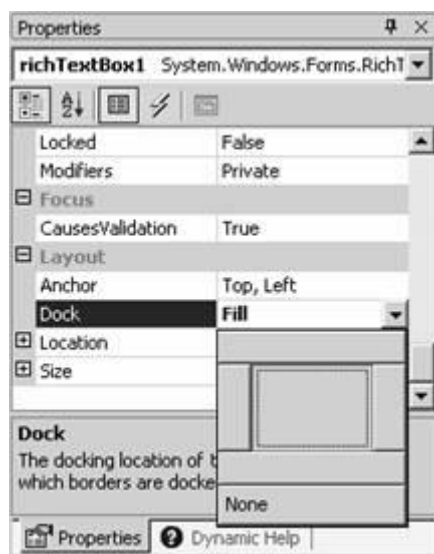


Рисунок 4.3 – Редактирование свойства Dock

Если запустить полученное приложение на выполнение, то в результате на экране должно появиться окно, показанное на рисунке 4.4.

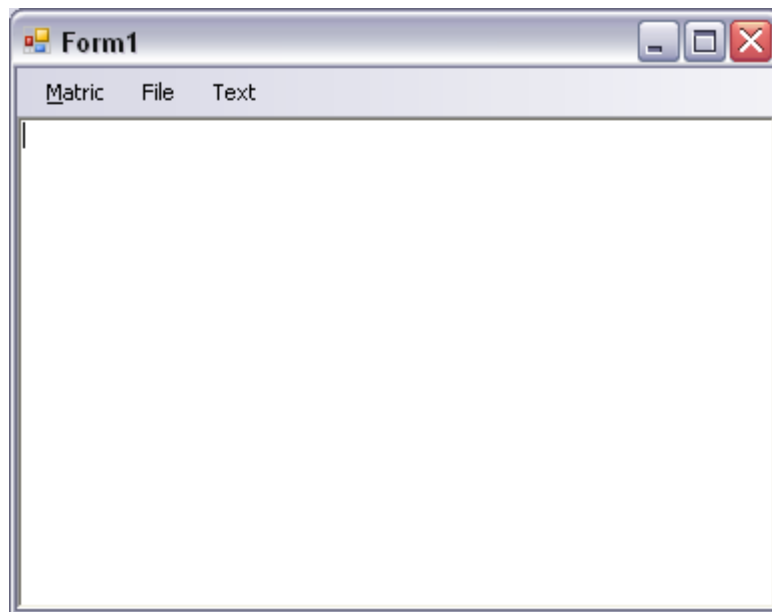


Рисунок 4.4 – Рабочее окно программы.

Обработчики событий от команд меню создаются таким же образом, что и обработчики событий от кнопок. Чтобы создать обработчик события для команды меню, ее нужно щелкнуть дважды левой клавишей мыши.

Создадим обработчики событий для всех команд, но не режимов работы программы. Все обработчики событий создаются с пустым телом, например:

```
private void sozMToolStripMenuItem_Click(object sender, EventArgs e)
{ }
```

Во многих приложениях основные команды меню программы дублируются «иконками», на инструментальных панелях. Создадим инструментальную панель для команд нашего приложения.

Чтобы добавить инструментальную панель в окно нашего приложения, перетащим мышью ее значок ToolStrip из окна элементов управления среды Visual Studio .NET в окно проектирования формы нашего приложения.

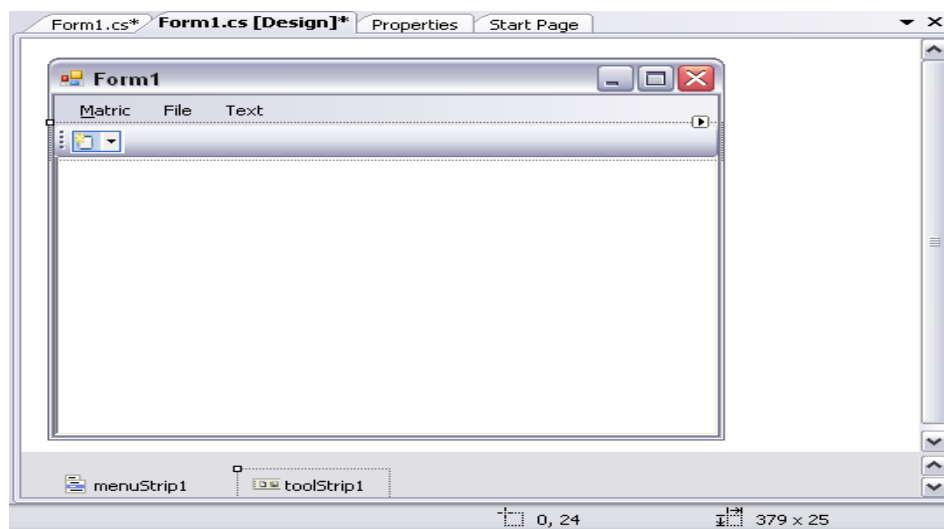


Рисунок 4.5 – Первый этап создания инструментальной панели приложения

По умолчанию окно инструментальной панели появится в верхней части формы. В только что добавленной панели нет ни одной кнопки. На рисунке 4.5 показан первый этап создания инструментальной панели.

На рисунке 4.5 видно, что окно создаваемой инструментальной панели находится внутри окна редактора текста (в верхней его части). Чтобы исправить это положение, щелкните правой кнопкой мыши окно редактора текста, а затем выберите из контекстного меню строку Bring to Front (но не Properties). В результате окна примут правильное взаимное расположение.

В панели инструментов Toolbox среды Visual Studio .NET имеется элемент управления ImageList, специально предназначенный для хранения списков изображений. Этот элемент можно использовать совместно с инструментальной панелью ToolBar.

Добавьте элемент ImageList в наш проект, перетаскив его значок в нижнюю часть окна проектирования формы.

Теперь нужно подключить пустой пока список изображений imageList1 к инструментальной панели. Для этого в окне редактирования свойств Properties присвойте свойству ImageList инструментальной панели значение imageList1.

Прежде чем приступить к наполнению списка изображениями, Вам нужно подготовить эти изображения при помощи любого графического редактора или найти необходимые изображения и поместить их в отдельную папку своего проекта (в нашем случае папка с именем ImaList). Можно запустить поиск на диске C по шаблону *.bmp и среди найденных изображений выбрать наиболее подходящие по смыслу.

После этого Вам необходимо отредактировать свойство Images элемента imageList1, выбрав свойство Collection. При этом запустится редактор набора изображений, окно которого показано на рисунке 4.6

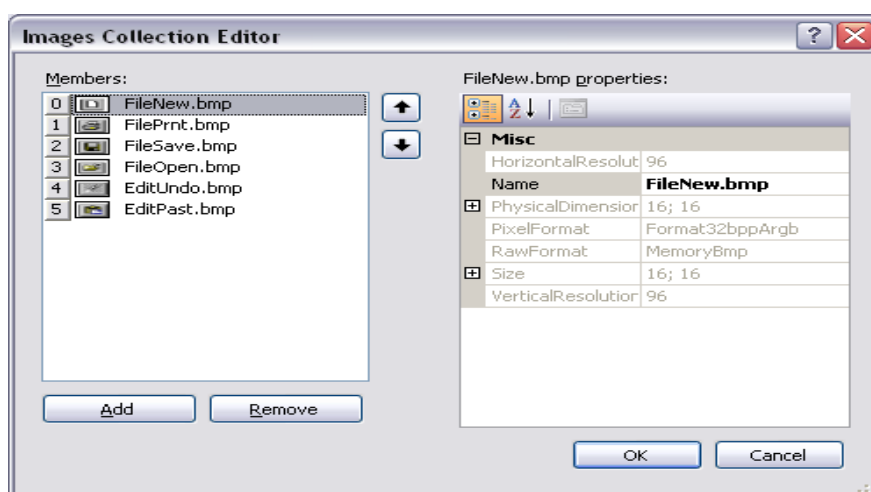


Рисунок 4.6 – Редактор списка изображений

Для добавления изображения в список щелкните кнопку Add, а затем выберите файл нужного изображения при помощи стандартного диалогового окна, которое необходимо открыть для нашей папки изображений (ImaList) смотри рисунок 4.7.

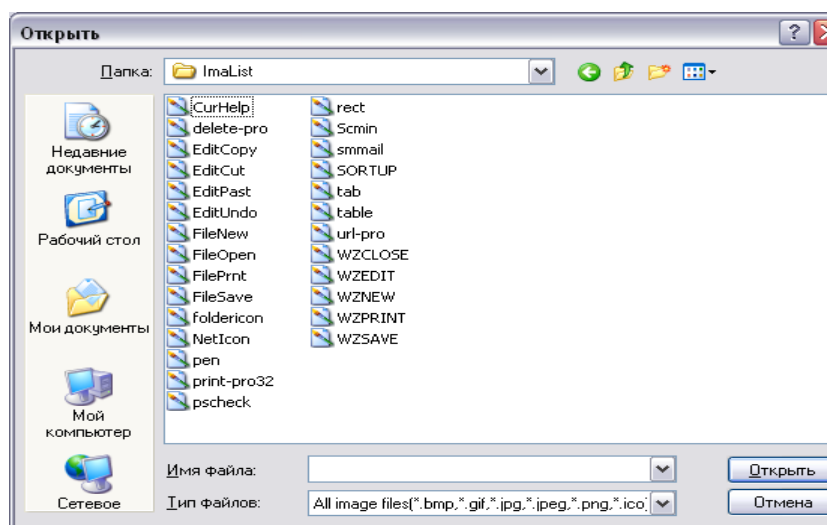


Рисунок 4.7 – Выбор файла изображения для кнопки инструментальной панели

Добавляем все изображения для наших 6 кнопок инструментальной панели. При необходимости можно изменить порядок их расположения с помощью кнопок с изображением стрелок. Можно удалять отдельные изображения из списка при помощи кнопки Remove.

Редактируя свойство ToolTipText каждой кнопки инструментальной панели, позволит нам снабдить каждую кнопку окном с поясняющим сообщением, которое появляется при наведении на кнопку курсора мыши.

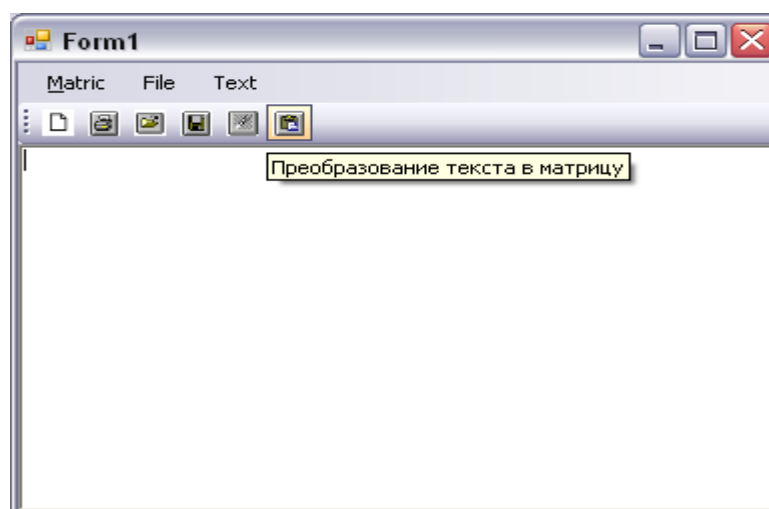


Рисунок 4.8 – Рабочее окно приложения при наведении курсора мышки на кнопку ObrabT инструментальной панели

Обработчики событий от кнопок инструментальной панели необходимо связать с обработчиками событий от команд меню. Для этого в свойствах каждой кнопки запускаем страницу «события» и выбираем событие Click (смотри рисунок 4.9).

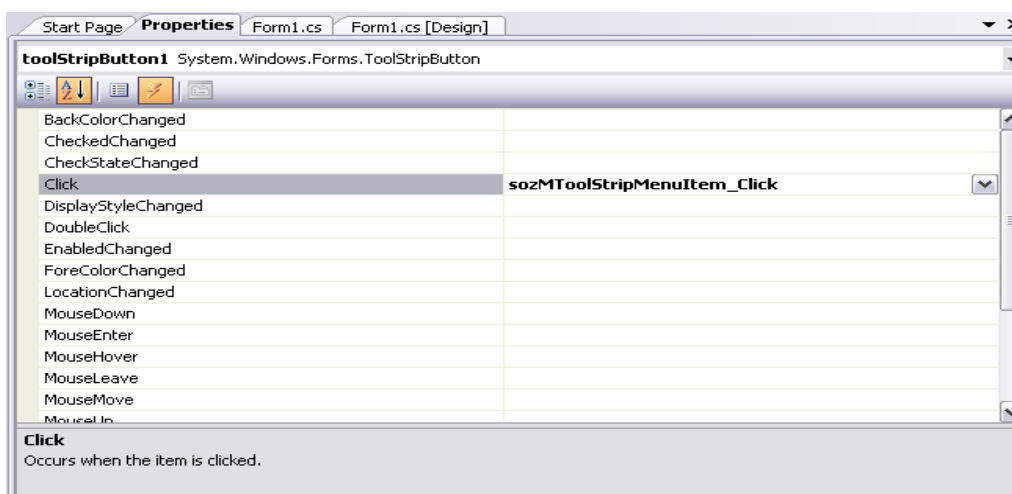


Рисунок 4.9 – Выбор обработчика событий каждой «иконки»

Дважды щелкаем по правой части таблицы события Click, при этом откроется диалоговое окно уже созданных обработчиков событий для команд нашего меню (смотри рисунок 4.10). Выбираем нужный по смыслу обработчик событий – в нашем примере это первая кнопка инструментальной панели и ей необходимо поставить в соответствие событие `sozMToolStripMenuItem_Click`.

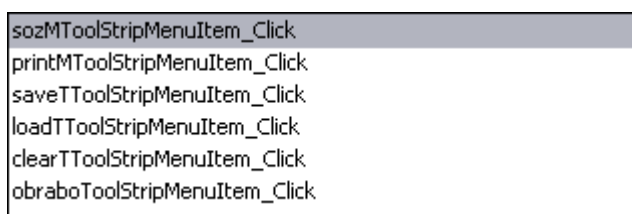


Рисунок 4.10 – Перечень обработчиков событий команд меню