

# Основы разработки приложений с использованием Windows Forms Встреча №7

Тема: Создание MDI-приложений

Домашнее задание:

**Основа интерфейса (MDI) приложения** – MDI родительская форма. Это – форма, которая содержит MDI дочерние окна. Дочерние окна являются «подокнами», с которыми пользователь взаимодействует в MDI приложении.

Для определения главного окна, как родительской формы в окне Свойств, необходимо установить свойство `IsMdiContainer` в значение `true`. Это определяет форму как MDI контейнер для дочерних форм. Для того чтобы родительское окно занимало весь экран необходимо свойству `WindowState` установить значение `Maximized`.

## Задание 1

Создание нового проекта с базовой формой

- Создайте новый проект WindowsForms, укажите имя `MdiApplication`.
- Переименуйте файл `Form1.cs` на `ParentForm.cs`
- Для свойства формы `IsMdiContainer` задайте значение `True`. Таким способом эта форма будет определена как родительская форма MDI.

**Создание меню для работы с формами.** В пространстве имен `System.Windows.Forms` предусмотрено большое количество типов для организации ниспадающих главных меню (расположенных в верхней части формы) и контекстных меню, открывающихся по щелчку правой кнопки мыши.

Элемент управления ***ToolStrip*** представляет собой контейнер, используемый для создания структур меню, панелей инструментов и строк состояний.

Элемент управления ***MenuStrip*** - это контейнер для структур меню в приложении. Система меню строится добавлением объектов `ToolStripMenu` к `menuStrip`.

Класс ***ToolStripMenuItem*** служит для построения структур меню. Каждый объект `ToolStripMenuItem` представляет отдельный пункт в системе меню.

В Windows также есть интерфейс для работы с так называемыми быстрыми клавишами, или акселераторами. Сочетание клавиш указывают из перечисления ***Shortcut***. Следует назначать стандартным пунктам общепринятые сочетания клавиш.

## Задание 2

**Создание пункта меню File.** Для создания стандартного ниспадающего меню выполните следующие действия:

- Добавьте на форму ParentForm элемент управления MenuStrip и задайте для его свойства Name значение MdiMenu.

- Выделите меню в верхней части формы и задайте имя первого пункта меню &File.

- Для свойства Name пункта меню File задайте значение FileMenuItem.

- Раскройте меню File.

- Выделите элемент, появившейся под элементом File, и задайте его как &New.

- Для свойства Name пункта меню New задайте значение NewMenuItem.

- Выделите элемент, появившийся под элементом New, и задайте его как &Exit.

- Для свойства Name пункта меню Exit задайте значение ExitMenuItem.

- Дважды кликните левой кнопкой мыши по пункту меню Exit для создания обработчика события Click.

- В обработчик события Click для пункта меню следующий код:

```
this.Close ();
```

Как вы думаете, для чего используется знак & перед буквой в названии пункта меню? Изучите особенности названия пункта меню, заданного со знаком & в любом из мест названия и без него.

### Задание 3

Создание пункта меню Window. При работе с несколькими окнами в MDI-приложениях удобно упорядочивать их на экране. Метод LayoutMdi содержит перечисление MdiLayout, содержащее четыре члена. ArrangeIcons переключает фокус на выбранную форму, в свойстве MdiList пункта меню ArrangeIcons устанавливаем также значение true. При открытии нескольких новых документов окна располагаются каскадом, их можно расположить горизонтально — значение TileHorizontal или вертикально — значение TileVertical, а затем снова вернуть каскадное расположение — Cascade.

Для упорядоченного расположения дочерних окон в главном окне выполните следующие действия:

- Для свойства Name пункта меню Window задайте значение WindowMenuItem.

- Раскройте меню Window.

- Выделите элемент, появившейся под элементом Window, и задайте для его свойства Text значение &Cascade.

- Для свойства Name пункта меню Cascade задайте значение WindowCascadeMenuItem.

- Выделите элемент, появившийся под элементом Cascade, и задайте для его свойства Text значение &Tile.

- Для свойства Name пункта меню Tile задайте значение WindowTileMenuItem.

■ Дважды кликните левой кнопкой мыши по пункту меню Cascade для создания обработчика события Click:

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

■ Вернитесь в режим конструктора и дважды кликните левой кнопкой мыши по пункту меню Tile

■ В обработчик события Click для пункта меню Tile добавьте следующий код:

```
this.LayoutMdi(System.Windows.Forms.MdiLayout.TileHorizontal);
```

■ Повторите действия по добавлению пункта меню и его обработки для задания вертикального расположения открытых окон.

#### Задание 4

Реализация списка открытых окон в меню Window. Для главного меню приложения (в нашем случае для Mdimenu) укажите в свойстве MdiWindowListItem имя пункта, созданного для отображения списка открытых окон (в нашем случае – это WindowMenuItem). При создании меню графическими средствами Visual Studio автоматически внесет необходимые изменения в служебный метод InitializeComponent и добавит переменные-члены, представляющие созданные элементы меню.

#### Задание 5

Создание дочерней формы и дочернего меню

■ Создайте еще одно окно, которое будет дочерним. Для этого выберите пункт меню Project/Add Windows Form.

■ Задайте имя формы ChildForm.cs

■ Для свойства Text формы задайте значение Child Form.

■ В панели инструментов выберите элемент RichTextBox и задайте для его свойства Name значение ChildTextBox.

■ Для свойства Dock элемента управления RichTextBox задайте значение Fill.

■ Удалите существующий текст (если он есть) для свойства Text элемента управления RichTextBox и оставьте его пустым.

■ В панели инструментов выберите элемент управления MenuStrip.

■ Для свойства Name элемента управления MenuStrip задайте значение ChildWindowMenu.

■ Выделите меню в верхней части формы и наберите текст F&ormat.

■ Для свойства Name пункта меню Format задайте значение FormatMenuItem, для свойства MergeAction установите значение Insert, а свойству MergeIndex значение 1. В этом случае меню Format будет располагаться после File при объединении базового и дочерних меню.

■ Выделите элемент, появившийся под элементом Format, и наберите текст &Toggle Foreground.

- Для свойства Name пункта меню Toggle Foreground задайте значение ToggleMenuItem.

- Дважды кликните левой кнопкой мыши по пункту меню Toggle Foreground и добавьте следующий код в обработчик события Click:

```
if (ToggleMenuItem.Checked)
{
    ToggleMenuItem.Checked = false;
    ChildTextBox.ForeColor = System.Drawing.Color.Black;
}
else
{
    ToggleMenuItem.Checked = true;
    ChildTextBox.ForeColor = System.Drawing.Color.Blue;
}
```

## Задание 6

Отображение дочерней формы

- Откройте ParentForm.cs в режиме конструктора.

- Дважды кликните левой кнопкой мыши по кнопке New в меню File для создания обработчика события Click.

- Добавьте следующий код для обработчика события Click для пункта меню New:

```
ChildForm newChild = new ChildForm();
newChild.MdiParent = this;
newChild.Show();
```

- Для проверки работы приложения постройте и запустите приложение. Когда появится родительская форма, выберите пункт меню File | New.

В родительском окне появится новая дочерняя форма. Обратите внимание на то, дочернее меню сливается с родительским и пункты меню упорядочиваются в соответствии со свойством *MergeIndex*, установленным ранее.

- Наберите какой-нибудь текст в дочернем окне и воспользуйтесь пунктом меню Format для изменения цвета шрифта текста.

- Откройте еще несколько дочерних окон.

- Выберите пункт меню Window | Tile. Обратите внимание на то, что дочерние окна выстраиваются в упорядоченном порядке.

- Закройте все дочерние окна. Обратите внимание на то, что, когда закроется последнее дочернее окно, меню родительской форм изменится, и оттуда исчезнет пункт Format.

- Для закрытия приложения выберите пункт меню File | Exit.

## Задание 7

Вырезание, копирование и вставка текстовых фрагментов Теперь приступим к созданию обработчиков для стандартных операций вырезания,

копирования и вставки. Элемент управления RichTextBox имеет свойство SelectedText, которое содержит выделенный фрагмент текста. На основании этого свойства и будут реализованы действия по работе с текстом. В коде дочерней формы объявите переменную BufferText, в которой будет храниться буферизованный фрагмент текста:

```
private String BufferText;
```

Далее создайте соответствующие методы:

```
private String BufferText;
// Вырезание текста
public void Cut() {
    this.BufferText = ChildTextBox.SelectedText;
    ChildTextBox.SelectedText = "";
}
// Копирование текста
public void Copy() {
    this.BufferText = ChildTextBox.SelectedText;
}
// Вставка
public void Paste() {
    ChildTextBox.SelectedText = this.BufferText;
}
// Выделение всего текста - используем свойство SelectAll элемента управления
// RichTextBox
public void SelectAll() {
    ChildTextBox.SelectAll();
}
// Удаление
public void Delete() {
    ChildTextBox.SelectedText = "";
    this.BufferText = "";
}
```

Для работы с текстом в дочерней форме самостоятельно добавьте пункт меню Edit и 5 подпунктов этого меню (Копирование, Вырезание, Удаление, Вставка, Выделение).

Переключитесь в режим дизайна формы и создайте обработчики для всех пунктов меню по следующему примеру:

```
private void cutMenuItem_Click(object sender, EventArgs e)
{
    this.Cut();
}

private void copyMenuItem_Click(object sender, EventArgs e)
{
    this.Copy();
}
```

## Задание 8

**Создание контекстного меню.** Контекстное меню дублирует некоторые действия основного меню. Добавьте элемент управления contextMenuStrip из окна ToolBox на форму ChildForm. Добавьте пункты контекстного меню точно так же, как сделали для главного меню Edit.

Свойство Text и Shortcut пунктов меню оставьте прежними.

Если установить для свойства ShowShortcut значение false, то сочетания клавиш будут работать, но в самом меню отображаться не будут. В обработчике пунктов для каждого подпункта вызовите методы по примеру:

```
private void cutContextMenuItem_Click(object sender, EventArgs e)
{
    Copy();
}
```

Необходимо определить, где будет появляться контекстное меню. Элемент RichTextBox, так же как и формы ParentForm и ChildForm, имеет свойство ContextMenuStrip, где мы и указываем contextMenuStrip1, поскольку нам нужно отображать меню именно в текстовом поле.

## Задание 9

**Задание нумерованного заголовка дочерним окнам.** Обратите внимание, что заголовок у дочерних окон одинаковый. При создании нескольких документов, например в Microsoft Word, они называются Документы, где N — номер документа. Реализуйте эту возможность:

■■ Откройте код родительской формы и в классе ParentForm объявите переменную openDocuments:

```
private int openDocuments = 0;
```

К свойству Text дочерней формы добавьте счетчик числа открываемых документов (в коде обработчика события Click для пункта меню New):

```
newChild.Text = newChild.Text+» «+ ++openDocuments;
```