## Практическая работа №23. Задание 1.

## *Разработка приложений с многодокументным интерфейсом*

##### 1. Цель работы

Изучить теоретические принципы и получить практические навыки разработки программ на основе многооконного (многодокументного) интерфейса*.*

**2. Краткая теория** Аббревиатура MDI расшифровывается как Multiple-document interface – многооконный (многодокументный) интерфейс*.* То есть MDI-приложения позволяют отображать сразу несколько документов одновременно. При этом каждый документ будет отображаться в своем собственном окне. Обычно MDI-приложения имеют в основном меню подпункты для переключения между окнами и документами.

Механизм работы MDI-приложений немного сложнее, чем обычных приложений, базирующихся на диалогах. Основным окном MDI-приложения является *родительская форма*. Она может содержать несколько *дочерних окон*. Только одно из дочерних окон может быть активно в один момент времени.

###### 2.1. Создание родительской формы

После создания нового проекта (можно использовать уже существующий проект) необходимо выбрать форму, которая будет играть роль главной (родительской) и присвоить ее свойству **IsMdiContainer** значение **true**.

Родительская форма, как правило, имеет главное меню, в котором предусмотрены пункты для управления дочерними окнами.

###### 2.2. Создание дочерних окон

Любое окно, существующее в приложении, может быть дочерним. Разработчик сам создает *шаблон формы*, которая будет являться дочерним окном.

Для создания шаблона дочернего окна, необходимо добавить в приложение новую форму: Project  Add Windows Form  Windows Form.

Создание дочерних окон в приложении обычно происходит при выборе пункта меню «&Создать». Для этого необходимо создать обработчик этого пункта меню – обработчик события **Click**. Код обработчика:

**Form2 newMDIChild = new Form2(); newMDIChild.MdiParent = this; newMDIChild.Show();**

В функции создается экземпляр класса **Form2** с именем **newMDIChild**.

Объект **newMDIChild** – это обычное окно. Для того чтобы создаваемое окно отображалось как дочерняя форма приложения, необходимо установить его свойство **MdiParent** равным **this**. Таким образом указывается, что родительской формой создаваемого окна является главная форма приложения. Метод **Show()** позволяет отобразить форму на экране.

Ссылка на текущее дочернее окно может быть получена с помощью свойства **ActiveMdiChild** родительской формы.

##### 3. Пример создания MDI-приложения

Для создания тестового примера необходимо выполнить следующие пошаговые инструкции:

1. Создайте новый проект.
2. Проект содержит только одну форму **Form1**. Сделайте ее главной (родительской).
3. Установите свойству **WindowState** формы **Form1** значение **Maximized**. При запуске программы главное окно будет принимать максимальный размер (раскрываться на весь экран). Аналогичного эффекта можно добиться, если в конструкторе формы **Form1** дописатьследующий код:

**this.WindowState=FormWindowState.Maximized;**

1. Создайте главное меню.
2. Добавьте пункты меню «&Файл» и «&Окно». Измените их свойство **name** на **menuItemFile** и **menuItemWindow** соответственно.
3. Добавьте в меню «&Файл» пункт «&Создать». Свойство **name** установите в **menuItemNew**. Этот пункт меню будет предназначен для создания дочерних окон.
4. Добавьте в проект еще одну форму, которая будет шаблоном дочерней формы.
5. Создайте обработчик события **Click** для пункта меню «&Создать» и добавьте код из п.4.2. краткой теории лабораторной работы.
6. Добавьте в меню «&Окно» пункты «&Горизонтально», «&Вертикально» и «&Каскадом». Свойство **name** установите в **menu-**

**ItemTileHorizontal**, **menuItemTileVertical** и  **menuItemCascade** соответственно. Эти пункты меню предназначены для вариантов упорядочения дочерних окон.

1. Выберите элемент главного меню и установите его свойству

**MdiWindowListItem** значение **menuItemWindow**. Это означает, что в пункте меню «&Окно» будет отображаться список всех открытых дочерних окон.

1. Создайте один общий обработчик для подпунктов меню пункта «&Окно». Для этого выделите все три пункта меню, удерживая клавишу Ctrl, и щелкните два раза указателем мыши по событию **Click** в окне свойств. Замените имя события (свойство **name**), созданное по умолчанию, на имя **menuItemTool\_Click**. Функция **menuItemTool\_Click** будет устанавливать вариант упорядочения дочерних окон.
2. Добавьте следующий код в функцию **menuItemTool\_Click**:

**ToolStripMenuItem item = sender as ToolStripMenuItem; switch (item.Text)**

**{ case "&Горизонтально": this.LayoutMdi(MdiLayout.TileHorizontal); break; case "&Вертикально":**

**this.LayoutMdi(MdiLayout.TileVertical); break; case "Каскадом": this.LayoutMdi(MdiLayout.Cascade); break;**

**}**

1. Запустите программу. Выберите из меню пункт «Файл»/ «Создать». На экране появится дочернее окно с именем «**Form2**». Вы можете создать еще несколько дочерних окон, воспользовавшись все тем же пунктом меню. При этом пункт меню «Окно» будет содержать список всех дочерних окон, открытых на текущий момент.
2. Попробуйте различные варианты упорядочения дочерних окон, воспользовавшись пунктом меню «Окно».
3. Заголовки всех дочерних окон одинаковые – «**Form2**». Модифицируем приложение таким образом, чтобы при создании нового дочернего окна в заголовке появлялся порядковый номер этого окна. Для этого необходимо изменить код обработчика события **Click** пункта меню «&Создать» следующим образом:

**Form2 newMDIChild = new Form2(); newMDIChild.MdiParent = this; int wnd\_num = this.MdiChildren.Length-1; newMDIChild.Text = "Doc"+wnd\_num.ToString(); newMDIChild.Show();**

Свойство **MdiChildren** содержит массив дочерних форм.

1. Запустите программу. Создайте несколько дочерних окон. Посмотрите как изменяются заголовки.
2. Добавьте на форму панель инструментов (компонент **ToolStrip**) с кнопками, соответствующими пунктам меню. Рисунки на кнопки можно добавить через свойство **Image**.
3. В качестве обработчика события Click каждой кнопки панели инструментов укажите функцию-обработчик соответствующего пункта меню. После этого реакция на выбор пункта меню или нажатие соответствующей кнопки панели инструментов будет одинаковой.

##### 4. Варианты заданий для самостоятельного решения

В каждом варианте в главном окне необходимо создать меню и панель инструментов (компонент ToolStrip) с кнопками, дублирующими пункты меню.

1. Дан список фамилий (загружается из файла при создании дочернего окна). Организовать отбор во всех списках (во всех дочерних окнах) по первой букве фамилий. Результаты отображать в виде списка в отдельном окне (не дочернем).
2. Создать простейший текстовый редактор с возможностью открытия и сохранения файлов.
3. Создайте приложение «Секундомер». При создании дочернего окна должен запускаться секундомер. В окне должна быть кнопка останова, паузы и нового запуска.
4. Создать простейшее приложение для просмотра графических файлов с возможностью открытия и сохранения изображений.
5. Создать следующее приложение. В каждом дочернем окне – текстовая метка или поле ввода, которые работают как числовой счетчик. Скорость увеличения задается таймером. Скорость таймера можно менять одновременно для всех дочерних окон в отдельном диалоговом окне (не дочернем).
6. Создать приложение с двумя видами шаблонов дочерних окон. В первом случае должен осуществляться контроль ввода только цифровых данных, во втором – только текстовых. Заголовок должен содержать слово «Цифры» или «Символы».
7. Создайте приложение «Конверт». Шаблон дочернего окна должен представлять собой образец заполнения полей на почтовом конверте: в левом верхнем углу адрес отправителя, внизу справа адрес получателя. В заголовке окна должен отображаться получатель.
8. Создайте приложение «Статистика текста». При наборе текста с клавиатуры должен проводиться подсчет и осуществляться вывод количества знаков препинания, букв и цифр.
9. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения цвета линии этого объекта.
10. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения цвета линии этого объекта с помощью задания интенсивности красного, зеленого и синего цветов (модель RGB) через компоненты **TrackBar**.
11. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения стиля линии этого объекта (для этого у пера нужно изменять свойство **DashStyle**).
12. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения толщины линии этого объекта.
13. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения цвета заливки этого объекта.
14. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения цвета заливки этого объекта с помощью задания интенсивности красного, зеленого и синего цветов (модель RGB) через компоненты **TrackBar**.
15. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения стиля заливки этого объекта.
16. Создать приложение, где в каждом дочернем окне изображается движущийся графический объект (круг, прямоугольник и т.п.).
17. Разработать приложение для ввода результатов сессии. Организовать табличный ввод, используя компонент **DataGridView**. Предусмотреть запись результатов в текстовый файл.
18. Разработать приложение «Табулирование функций». Программа позволяет получить значения аргумента и функции в заданном интервале с заданным шагом. Вид функции можно задать выбором в компоненте **ListBox** или в компонентах **RadioButton**.
19. Разработать приложение, где в каждом дочернем окне должен вводиться пароль. Если пароль введен верно, то отображается какое-нибудь графическое изображение.
20. Разработать следующее приложение. В каждом дочернем окне – компонент-счетчик. В главном меню есть пункт, по нажатию которого открывается диалоговое окно, где отображается сумма чисел из счетчиков во всех открытых дочерних окнах.
21. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения координат этого объекта.
22. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность изменения размера этого объекта.
23. Создать приложение, где в каждом дочернем окне изображается какой-то графический объект (круг, прямоугольник и т.п.). Предусмотреть возможность поворота этого объекта (влево на 90, вправо на 90, на 180).
24. Создать простейший текстовый редактор с возможностью изменения шрифта и цвета текста.
25. Разработать следующее приложение. В каждом дочернем окне выбирается дата с помощью компонента **DateTimePicker**. В главном меню есть пункт, по нажатию которого открывается диалоговое окно, где выводится самая ранняя из всех дат, заданных в открытых дочерних окнах.
26. Разработать следующее приложение. В диалоговом окне (не дочернем) вводится текстовая строка. Программа должна разбить эту строку на слова и каждое слово передать в отдельное дочернее окно, например, в компонент **Label** или **TextBox** (должно открыться столько окон, сколько слов в строке).
27. Разработать следующее приложение. В диалоговом окне (не дочернем) вводится матрица чисел (в компонент **DataGridView**). Программа должна разбить эту матрицу на строки и каждую строку отобразить в отдельном дочернем окне (должно открыться столько окон, сколько строк в матрице).
28. Разработать следующее приложение. В диалоговом окне (не дочернем) вводится матрица чисел (в компонент **DataGridView**). Программа должна разбить эту матрицу на столбцы и каждый столбец отобразить в отдельном дочернем окне (должно открыться столько окон, сколько столбцов в матрице).
29. Разработать следующее приложение. В каждом дочернем окне вводится одно и то же количество чисел (например, в компонент **DataGridView** или **ListBox**). Необходимо все введенные числа из каждого дочернего окна объединить по строкам в матрицу, матрицу отобразить в отдельном диалоговом окне (не дочернем).
30. Разработать следующее приложение. В каждом дочернем окне вводится одно и то же количество чисел (например, в компонент DataGridView или ListBox). Необходимо все введенные числа из каждого дочернего окна объединить по столбцам в матрицу, матрицу отобразить в отдельном диалоговом окне (не дочернем).