Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЕТ

по лабораторной работе №3

Исследование разработки gui. Создание sdi-приложений обработка событий

Выполнил:

ст. гр. ИС/б-21-о

Куркчи А. Э.

Проверил:

Шишкевич В. Е.

Севастополь

2016

1. ЦЕЛЬ РАБОТЫ

В ходе выполнения данной лабораторной работы необходимо изучить принципы разработки графического интерфейса прило­жений для ОС Windows в Visual Studio .Net, освоить использование элементов графического интерфейса для управления работой приложения, освоить принципы построения иерархических меню, создания диалоговых окон, изучить модель обработки событий в языке C#.

1. ПОСТАНОВКА ЗАДАЧИ

Вариант № 12

1. Создать учебные примеры (программы 2-5) и разобрать принцип их работы. Поместите в отчет примеры работы программ, их код с комментариями.
2. Создать SDI-приложение (Single Document Interface, одно­документный интерфейс) с элементами ввода и отображения полей класса из задания к лабораторной работе 2. Для этого используйте раз­личные элементы управления: текстовые поля, списки, независимые и радиокнопки, а также панели и менеджеры компоновки.
3. Ввод новых данных осуществлять через дополнительную диало­говую форму.
4. При изменении данных запрашивать подтверждение через окно диалога. В случае неполных данных сообщать об ошибке.
5. Объекты сохранять в коллекции.
6. Реализовать просмотр всей коллекции объектов через список. Для редактирования выбранного объекта создать дополнительную форму модального диалога.
7. Добавить на форму меню, позволяющее работать с пунктами: добавить, просмотреть, удалить, редактировать, справка.
8. Дублировать основные операции панелью инструментов.
9. ТЕКСТ ПРОГРАММЫ

Файл Program.cs

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**Linq**;**

**using** System**.**Threading**.**Tasks**;**

**using** System**.**Windows**.**Forms**;**

**namespace** Lab**.**\_03

**{**

static class Program

**{**

/// <summary>

/// The main entry point for the application.

/// </summary>

**[**STAThread**]**

static void Main**()**

**{**

Application**.**EnableVisualStyles**();**

Application**.**SetCompatibleTextRenderingDefault**(false);**

Application**.**Run**(new** MainForm**());**

**}**

**}**

**}**

Файл InvalidValueException.cs

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**Linq**;**

**using** System**.**Text**;**

**using** System**.**Threading**.**Tasks**;**

**namespace** Lab**.**\_03

**{**

class InvalidValueException**:** Exception

**{**

**new** **public** string Message **{** get**;** set**;** **}**

**public** InvalidValueException**(**FieldType fieldType**)**

**{**

**switch** **(**fieldType**)**

**{**

**case** FieldType**.**Name**:**

Message **=** "Name field can`t be changed"**;**

**break;**

**case** FieldType**.**Address**:**

Message **=** "Address field can`t be empty"**;**

**break;**

**case** FieldType**.**Index**:**

Message **=** "Index field is invalid"**;**

**break;**

**default:**

Message **=** "Unknown field is invalid"**;**

**break;**

**}**

**}**

**}**

enum FieldType

**{**

Name**,**

Address**,**

Index

**}**

**}**

Файл AMuseum.cs

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**Linq**;**

**using** System**.**Text**;**

**using** System**.**Threading**.**Tasks**;**

**namespace** Lab**.**\_03

**{**

**[**Serializable**]**

**public abstract** class AMuseum

**{**

**private** string name**;**

**public** string Name

**{**

get **{** **return** name**;** **}**

set **{** **if** **(**name **==** **null)** **{** name **=** value**;** **}** **else** **{** **throw** **new** InvalidValueException**(**FieldType**.**Name**);** **}** **}**

**}**

**private** string address**;**

**public** string Address

**{**

get **{** **return** address**;** **}**

set **{** **if** **(**address **!=** ""**)** **{** address **=** value**;** **}** **else** **{** **throw** **new** InvalidValueException**(**FieldType**.**Address**);** **}** **}**

**}**

**private** int hall\_count **=** 0**;**

**public** int Hall\_count

**{**

get **{** **return** hall\_count**;** **}**

**}**

const int hall\_max\_count **=** 100**;**

**private** string**[]** halls **=** **new** string**[**hall\_max\_count**];**

**public** string **this[**int i**]**

**{**

get

**{**

**if** **(**i **>=** 0 **&&** i **<=** hall\_count**)**

**{**

**return** halls**[**i**];**

**}**

**else**

**{**

**throw** **new** InvalidValueException**(**FieldType**.**Index**);**

**}**

**}**

set

**{**

**if** **(**i **==** hall\_count**)**

**{**

halls**[**i**]** **=** value**;**

hall\_count**++;**

**}**

**else** **if** **(**i **>=** 0 **&&** i **<=** hall\_count**)**

**{**

halls**[**i**]** **=** value**;**

**}**

**else**

**{**

**throw** **new** InvalidValueException**(**FieldType**.**Index**);**

**}**

**}**

**}**

**public** AMuseum**(**string name**,** string address**)**

**{**

**this.**name **=** name**;**

**this.**address **=** address**;**

**}**

**public** **virtual** void print**()**

**{**

Console**.**WriteLine**(**"Museum \"{0}\", {1}"**,** name**,** address**);**

**}**

**public** void printHalls**()**

**{**

**for** **(**int i **=** 0**;** i **<** hall\_count**;**i**++)**

**{**

Console**.**WriteLine**(**"Hall #{0}: {1}"**,** i**+**1**,** **this[**i**]);**

**}**

**}**

**public** **abstract** void printInfo**();**

**}**

**}**

Файл IVisitable.cs

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**Linq**;**

**using** System**.**Text**;**

**using** System**.**Threading**.**Tasks**;**

**namespace** Lab**.**\_03

**{**

**interface** IVisitable

**{**

void visit**();**

int getVisits**();**

**}**

**}**

Файл Museum.cs

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**ComponentModel**;**

**using** System**.**Linq**;**

**using** System**.**Text**;**

**using** System**.**Threading**.**Tasks**;**

**namespace** Lab**.**\_03

**{**

**[**Serializable**]**

**public** class Museum **:** AMuseum**,** IVisitable**,** INotifyPropertyChanged

**{**

**private** string owner**;**

**public** string Owner

**{**

get **{** **return** owner**;** **}**

set **{** **if** **(**owner **!=** ""**)** owner **=** value**;** **}**

**}**

**private** int visits **=** 0**;**

**public** int Visits

**{**

get **{** **return** visits**;** **}**

set **{** **if** **(**visits **==** 0**)** visits **=** value**;** **}**

**}**

**public** Museum**(**string name**,** string address**,** string owner**)**

**:** **base(**name**,** address**)**

**{**

**this.**owner **=** owner**;**

**}**

**public** Museum**(**string name**,** string address**,** string owner**,** int visits**)**

**:** **base(**name**,** address**)**

**{**

**this.**owner **=** owner**;**

**this.**visits **=** visits**;**

**}**

**public** Museum**(**string name**,** string address**,** string owner**,** string visits**)**

**:** **base(**name**,** address**)**

**{**

**this.**owner **=** owner**;**

**this.**visits **=** int**.**Parse**(**visits**);**

**}**

**new** **public** void print**()**

**{**

Console**.**WriteLine**(**"Museum \"{0}\", {1}; Owner: {2}, Visits: {3}"**,** Name**,** Address**,** owner**,** visits**);**

**base.**printHalls**();**

**}**

**override** **public** void printInfo**()**

**{**

Console**.**WriteLine**(**"Owner: {0}, Visits: {1}"**,** owner**,** visits**);**

**}**

**public** void visit**()**

**{**

visits**++;**

**}**

**public** int getVisits**()**

**{**

**return** visits**;**

**}**

**public** **event** PropertyChangedEventHandler PropertyChanged**;**

**private** void NotifyPropertyChanged**(**string name**)**

**{**

**if** **(**PropertyChanged **!=** **null)**

PropertyChanged**(this,** **new** PropertyChangedEventArgs**(**name**));**

**}**

**}**

**}**

Файл MainForm.cs

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**ComponentModel**;**

**using** System**.**Data**;**

**using** System**.**Drawing**;**

**using** System**.**IO**;**

**using** System**.**Linq**;**

**using** System**.**Runtime**.**Serialization**.**Formatters**.**Binary**;**

**using** System**.**Runtime**.**Serialization**;**

**using** System**.**Text**;**

**using** System**.**Threading**.**Tasks**;**

**using** System**.**Windows**.**Forms**;**

**namespace** Lab**.**\_03

**{**

**public** partial class MainForm **:** Form

**{**

**private** BindingList**<**Museum**>** list**;**

**public** MainForm**()**

**{**

InitializeComponent**();**

list **=** **new** BindingList**<**Museum**>();**

grid**.**DataSource **=** list**;**

grid**.**Columns**[**"Name"**].**DisplayIndex **=** 0**;**

grid**.**Columns**[**"Address"**].**DisplayIndex **=** 1**;**

grid**.**Columns**[**"Owner"**].**DisplayIndex **=** 2**;**

grid**.**Columns**[**"Visits"**].**DisplayIndex **=** 3**;**

grid**.**EditMode **=** DataGridViewEditMode**.**EditProgrammatically**;**

**}**

**private** void MainForm\_Load**(object** sender**,** EventArgs e**)**

**{**

DialogResult dr **=** MessageBox**.**Show**(**"Не желаете ли загрузить данные из файла?"**,** "Загрузка"**,** MessageBoxButtons**.**YesNo**);**

**if** **(**dr **==** DialogResult**.**Yes**)**

**{**

openFile**.**ShowDialog**();**

**}**

**}**

**private** void openFile\_FileOk**(object** sender**,** CancelEventArgs e**)**

**{**

readFromFile**(**openFile**.**FileName**);**

**}**

**private** void readFromFile**(**string filename**)**

**{**

list**.**Clear**();**

Stream stream **=** **new** FileStream**(**filename**,** FileMode**.**Open**,** FileAccess**.**Read**,** FileShare**.**None**);**

IFormatter formatter **=** **new** BinaryFormatter**();**

Console**.**WriteLine**(**stream**.**Length**);**

**while(**stream**.**Position **!=** stream**.**Length**)**

**{**

Museum m **=** **(**Museum**)**formatter**.**Deserialize**(**stream**);**

list**.**Add**(**m**);**

**}**

stream**.**Close**();**

**}**

**private** void MainForm\_FormClosing**(object** sender**,** FormClosingEventArgs e**)**

**{**

DialogResult dr**;**

**if** **(**list**.**Count **>** 0**)**

**{**

dr **=** MessageBox**.**Show**(**"Сохранить данные в файл перед закрытием?"**,** "Закрытие"**,** MessageBoxButtons**.**YesNoCancel**);**

**if** **(**dr **==** DialogResult**.**Yes**)**

**{**

saveFile**.**ShowDialog**();**

e**.**Cancel **=** **false;**

**}**

**else** **if** **(**dr **==** DialogResult**.**No**)**

**{**

e**.**Cancel **=** **false;**

**}**

**else**

**{**

e**.**Cancel **=** **true;**

**}**

**}**

**else**

**{**

dr **=** MessageBox**.**Show**(**"Вы действительно хотите закрыть?"**,** "Закрытие"**,** MessageBoxButtons**.**YesNo**);**

**if** **(**dr **==** DialogResult**.**Yes**)**

**{**

e**.**Cancel **=** **false;**

**}**

**else**

**{**

e**.**Cancel **=** **true;**

**}**

**}**

**}**

**private** void saveFile\_FileOk**(object** sender**,** CancelEventArgs e**)**

**{**

saveToFile**(**saveFile**.**FileName**);**

**}**

**private** void saveToFile**(**string filename**)**

**{**

Stream stream **=** **new** FileStream**(**filename**,** FileMode**.**Create**,** FileAccess**.**Write**,** FileShare**.**None**);**

IFormatter formatter **=** **new** BinaryFormatter**();**

**foreach(**Museum m **in** list**)**

**{**

formatter**.**Serialize**(**stream**,** m**);**

**}**

stream**.**Close**();**

**}**

**private** void открытьToolStripMenuItem\_Click**(object** sender**,** EventArgs e**)**

**{**

openFile**.**ShowDialog**();**

**}**

**private** void сохранитьКакToolStripMenuItem\_Click**(object** sender**,** EventArgs e**)**

**{**

saveFile**.**ShowDialog**();**

**}**

**private** void удалитьToolStripMenuItem\_Click**(object** sender**,** EventArgs e**)**

**{**

**if** **(**grid**.**CurrentRow **==** **null)**

**{**

**return;**

**}**

DialogResult dr **=** MessageBox**.**Show**(**"Вы действительно хотите удалить выбранный музей?"**,** "Удаление"**,** MessageBoxButtons**.**YesNo**);**

**if** **(**dr **==** DialogResult**.**Yes**)**

**{**

int index **=** grid**.**CurrentRow**.**Index**;**

list**.**RemoveAt**(**index**);**

grid**.**Refresh**();**

**}**

**}**

**private** void добавитьToolStripMenuItem\_Click**(object** sender**,** EventArgs e**)**

**{**

DataForm df **=** **new** DataForm**(this,** **-**1**,** **null);**

df**.**ShowDialog**();**

**}**

**public** void addOrEdit**(**int id**,** Museum m**)**

**{**

**if(**id **>=** 0**)**

**{**

list**[**id**]** **=** m**;**

**}**

**else**

**{**

list**.**Add**(**m**);**

**}**

**}**

**private** void изменитьToolStripMenuItem\_Click**(object** sender**,** EventArgs e**)**

**{**

**if(**grid**.**CurrentRow **!=** **null)**

**{**

DataForm df **=** **new** DataForm**(this,** grid**.**CurrentRow**.**Index**,** list**[**grid**.**CurrentRow**.**Index**]);**

df**.**ShowDialog**();**

**}**

**}**

**}**

**}**

Файл DataForm.cs

**using** System**;**

**using** System**.**Collections**.**Generic**;**

**using** System**.**ComponentModel**;**

**using** System**.**Data**;**

**using** System**.**Drawing**;**

**using** System**.**Linq**;**

**using** System**.**Text**;**

**using** System**.**Threading**.**Tasks**;**

**using** System**.**Windows**.**Forms**;**

**namespace** Lab**.**\_03

**{**

**public** partial class DataForm **:** Form

**{**

**private** bool IsEdit **=** **false;**

**private** int ElementID **=** 0**;**

**private** Museum CurMuseum **=** **null;**

**private** MainForm Parent **=** **null;**

**public** DataForm**(**MainForm parent**,** int elID**,** Museum m**)**

**{**

InitializeComponent**();**

Parent **=** parent**;**

IsEdit **=** elID **>=** 0 **&&** m **!=** **null;**

ElementID **=** elID**;**

CurMuseum **=** m**;**

**if(**IsEdit**)**

**{**

actBtn**.**Text **=** "Save"**;**

nameField**.**Text **=** m**.**Name**;**

addressField**.**Text **=** m**.**Address**;**

ownerField**.**Text **=** m**.**Owner**;**

visitsField**.**Text **=** Convert**.**ToString**(**m**.**Visits**);**

**}** **else**

**{**

actBtn**.**Text **=** "Add"**;**

**}**

**}**

**private** void actBtn\_Click**(object** sender**,** EventArgs e**)**

**{**

Parent**.**addOrEdit**(**ElementID**,** **new** Museum**(**nameField**.**Text**,** addressField**.**Text**,** ownerField**.**Text**,** visitsField**.**Text**));**

Close**();**

**}**

**}**

**}**

1. ТЕСТОВЫЙ ПРИМЕР

На рисунке 1 показано основное окно программы.

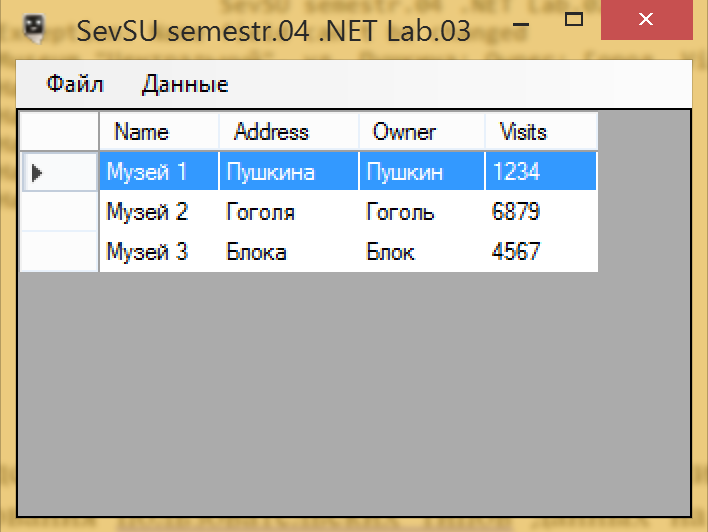


Рисунок 1 – основное окно программы

На рисунке 2 показано окно редактирования программы.

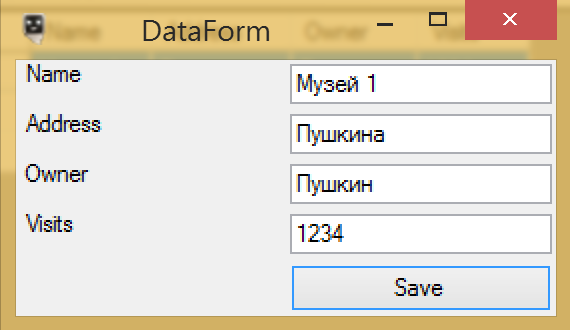


Рисунок 2 – окно редактирования программы

ВЫВОДЫ

В ходе данной лабораторной работы были изучены основы создания gui средствами C# WPF. Создано sdi приложение по заданному варианту с несколькими окнами и возможностью манипулировать данными. Также был изучен способ обработки событий в C# и в частности WPF sdi приложениях.