МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Вычислительная техника»

Дисциплина «Технология программирования»

**Лабораторная работа №2.**

**WPF приложения**

Выполнил:

студент группы ИВТАПбд-21

Кондратьев П. С.

Проверил:

Фролов В. А.

Ульяновск, 2018

Что вам, как разработчику, предлагает WPF?

Использование традиционных языков .NET-платформы - C# и VB.NET для создания логики приложения

Возможность декларативного определения графического интерфейса с помощью специального языка разметки XAML, основанном на xml и представляющем альтернативу программному созданию графики и элементов управления, а также возможность комбинировать XAML и C#/VB.NET

Независимость от разрешения экрана: поскольку в WPF все элементы измеряются в независимых от устройства единицах, приложения на WPF легко масштабируются под разные экраны с разным разрешением.

Новые возможности, которых сложно было достичь в WinForms, например, создание трехмерных моделей, привязка данных, использование таких элементов, как стили, шаблоны, темы и др.

Хорошее взаимодействие с WinForms, благодаря чему, например, в приложениях WPF можно использовать традиционные элементы управления из WinForms.

Богатые возможности по созданию различных приложений: это и мультимедиа, и двухмерная и трехмерная графика, и богатый набор встроенных элементов управления, а также возможность самим создавать новые элементы, создание анимаций, привязка данных, стили, шаблоны, темы и многое другое

Аппаратное ускорение графики - вне зависимости от того, работаете ли вы с 2D или 3D, графикой или текстом, все компоненты приложения транслируются в объекты, понятные Direct3D, и затем визуализируются с помощью процессора на видеокарте, что повышает производительность, делает графику более плавной.

Создание приложений под множество ОС семейства Windows - от Windows XP до Windows 10

**Техническое задание**

* 2 коллекции из 1 л/р в виде WPF приложения
* добавление, изменение, удаление элементов коллекций
* для добавления сделать диалоговое окно, с анимацией

Исходный код

namespace test

{

public static class Lists

{

public static List<Actor> people = new List<Actor>();

public static List<Test> tst = new List<Test>();

public static string ListActor()

{

string st = null;

foreach (Actor item in people)

{

st += item.ToString();

}

return st;

}

public static void EditActor(int id, int cost, string firstName, string lastName)

{

var subset = from s in people where (s.id == id) select s;

foreach (var s in subset)

{

s.cost = cost;

s.firstName = firstName;

s.lastName = lastName;

}

subset = null;

}

public static void EditTest(int id, int yearPopular)

{

var subset = from s in tst where (s.id == id) select s;

foreach (var s in subset)

{

s.yearPopular = yearPopular;

}

subset = null;

}

public static void AddActor(int id, int cost, string firstName, string lastName)

{

Actor p = new Actor(id, cost, firstName, lastName);

people.Add(p);

p = null;

}

public static string ListTest()

{

string st = null;

foreach (Test item in tst)

{

st += item.ToString();

}

return st;

}

public static void AddTest(int id, int yearPopular)

{

Test p1 = new Test(id, yearPopular);

tst.Add(p1);

}

public static string DeleteActor(int id)

{

people.RemoveAt(id);

return "delete object Actor, where id = " + id;

}

public static string DeleteTest(int id)

{

tst.RemoveAt(id);

return "delete object Test, where id = " + id;

}

}

public class Actor

{

public int id;

public int cost;

public string firstName;

public string lastName;

public Actor(int id, int cost, string firstName, string lastName)

{

this.id = id;

this.cost = cost;

this.firstName = firstName;

this.lastName = lastName;

}

public Actor(){}

public override string ToString()

{

return string.Format("id: {0} cost: {1} firstName: {2} lastName: {3}\n", id, cost, firstName, lastName);

}

}

public class Test

{

public int id;

public int yearPopular;

public Test(int id, int yearPopular)

{

this.id = id;

this.yearPopular = yearPopular;

}

public Test() { }

public override string ToString()

{

return string.Format("id: {0} yearPopular: {1}\n", id, yearPopular);

}

}

}

