МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет автоматики и вычислительной техники

Кафедра АСУ



**ОТЧЕТ**

по лабораторной работе №2, 3 «РЕАЛИЗАЦИЯ ПРОГРАММЫ, УПРАВЛЯЕМОЙ МЕТАДАННЫМИ», «ПРОГРАММНЫЕ МОДУЛИ МНОГОКРАТНОГО ИСПОЛЬЗОВАНИЯ»

Дисциплина «Технологии и методы программирования», 4 семестр  
  
Выполнили:

студент группы АВТ-214

Крипаков Леонид Дмитриевич  
Саушкин Никита Максимович

г. Новосибирск

2024 год

**Постановка задачи.**

1) Реализовать класс, конструктор которого создаёт меню, структура которого

определяется данными, находящимися во внешнем текстовом файле.

2) Любым текстовым редактором создать требуемые файлы. Для каждого из двух

пользователей должен быть свой набор пунктов меню.

3) Написать тестовую программу (оконное приложение) для проверки работы класса.

4) При выборе мышью пункта срабатывает обработчик пункта или появляется

подменю, если имя метода не указано.

5) Ознакомиться с приёмами создания программных модулей многократного использования, научиться применять их при проектировании и разработке программного обеспечения.

**Исходный код класса и тестирующей программы, оформленный в соответствии с приложением А.**

|  |
| --- |
| **Файл MetaDataWorker.cs** |
| using System;  using System.Collections.Generic;  using System.Text;  using System.Windows.Forms;  using System.IO;  using System.Reflection;  namespace MetaData  {  public class MetaDataWorker  {  private MenuStrip menuFromFile;  private Form form;  public MetaDataWorker(string FileName)  {  BuildMenu(FileName);  }  private void BuildMenu(string FileName)  {  string contents = File.ReadAllText(FileName);  string[] items = contents.Split(new string[1] { Environment.NewLine }, StringSplitOptions.RemoveEmptyEntries);  menuFromFile = new MenuStrip();  List<ToolStripMenuItem> lastItems = new List<ToolStripMenuItem>();  int lastNumber = -1;  for (int i = 0; i < items.Length; ++i)  {  string[] parts = items[i].Split(new string[1] { " " }, StringSplitOptions.RemoveEmptyEntries);  int number = Int32.Parse(parts[0]);  int access;  int LastPartOfName;  try  {  access = Int32.Parse(parts[parts.Length - 1]);  LastPartOfName = parts.Length - 2;  }  catch (Exception e)  {  access = Int32.Parse(parts[parts.Length - 2]);  LastPartOfName = parts.Length - 3;  }  string Text = "";  for (int j = 1; j <= LastPartOfName; ++j)  {  Text += parts[j];  if (j < LastPartOfName)  {  Text += " ";  }  }  ToolStripMenuItem item = new ToolStripMenuItem(Text);  if (access == 1)  {  item.Enabled = false;  }  if (access != 2)  {  if (number <= lastNumber)  {  int diff = lastNumber - number;  for (int k = 1; k <= diff + 1; ++k)  {  lastItems.Remove(lastItems[lastItems.Count - 1]);  }  }  if (lastItems.Count > 0)  {  lastItems[lastItems.Count - 1].DropDownItems.Add(item);  }  else  {  menuFromFile.Items.Add(item);  }  lastItems.Add(item);  lastNumber = number;  }  if (access == 0 && LastPartOfName == parts.Length - 3)  {  item.Name = parts[parts.Length - 1];  item.Click += new System.EventHandler(PointMethod);  }  }  }  private void PointMethod(object sender, EventArgs e)  {  string MethodName = ((ToolStripMenuItem)sender).Name;  MethodInfo mi = typeof(Form1).GetMethod(MethodName);  mi.Invoke(form, new object[] { });  }  public void SetForm(Form form)  {  this.form = form;  }  public MenuStrip getMenu()  {  return menuFromFile;  }    }  } |

|  |
| --- |
| **Файл Enter\_in\_system.cs** |
| using System;  using System.Collections.Generic;  using System.Text;  using System.IO;  using System.Security.Cryptography;  namespace MetaData  {  public class Enter\_in\_system  {  public static string GetHash(string password)  {  MD5 md5Hash = MD5.Create();  var sourceBytes = Encoding.UTF8.GetBytes(password);  var hashBytes = md5Hash.ComputeHash(sourceBytes);  string hash = BitConverter.ToString(hashBytes).Replace("-", string.Empty);  return hash;  }  public static int Authorisation(string login, string hash)  {  string contents = File.ReadAllText("logins.txt");  string[] parts = contents.Split(new string[1] { Environment.NewLine }, StringSplitOptions.RemoveEmptyEntries);  for (int i = 0; i < parts.Length; i++)  {  string[] lp = parts[i].Split(new string[1] { " " }, StringSplitOptions.RemoveEmptyEntries);  if (lp[0].Equals(login) && lp[1].Equals(hash))  {  return i + 1;  }  }  return 0;  }  }  } |

**Тексты управляющих файлов.**

**Menu1.txt**

0 Разное 0

0 Сотрудники 0

0 Приказы 0

0 Документы 0

0 Справочники 0

1 Отделы 0 Departs

1 Города 0 Towns

1 Должности 0 Posts

0 Окно 0

0 Справка 0

1 Оглавление 0 Content

1 О программе 0 About

**Menu2.txt**

0 Разное 0

1 Инструменты 0

2 Сортировка 0

3 По сотрудникам 0

3 По городам 0

1 Другие инструменты 0

2 Подсказки 0

3 Подсказка первая 0

2 История поиска 0

1 Инструменты новые 0

2 Обращение к администратору 0

0 Сотрудники 0

0 Приказы 0

0 Документы 0

0 Справочники 0

1 Отделы 0 Departs

1 Города 0 Towns

1 Должности 0 Posts

0 Окно 0

0 Справка 0

1 Оглавление 0 Content

1 О программе 0 About

**Скриншот экрана для каждого пользователя.**

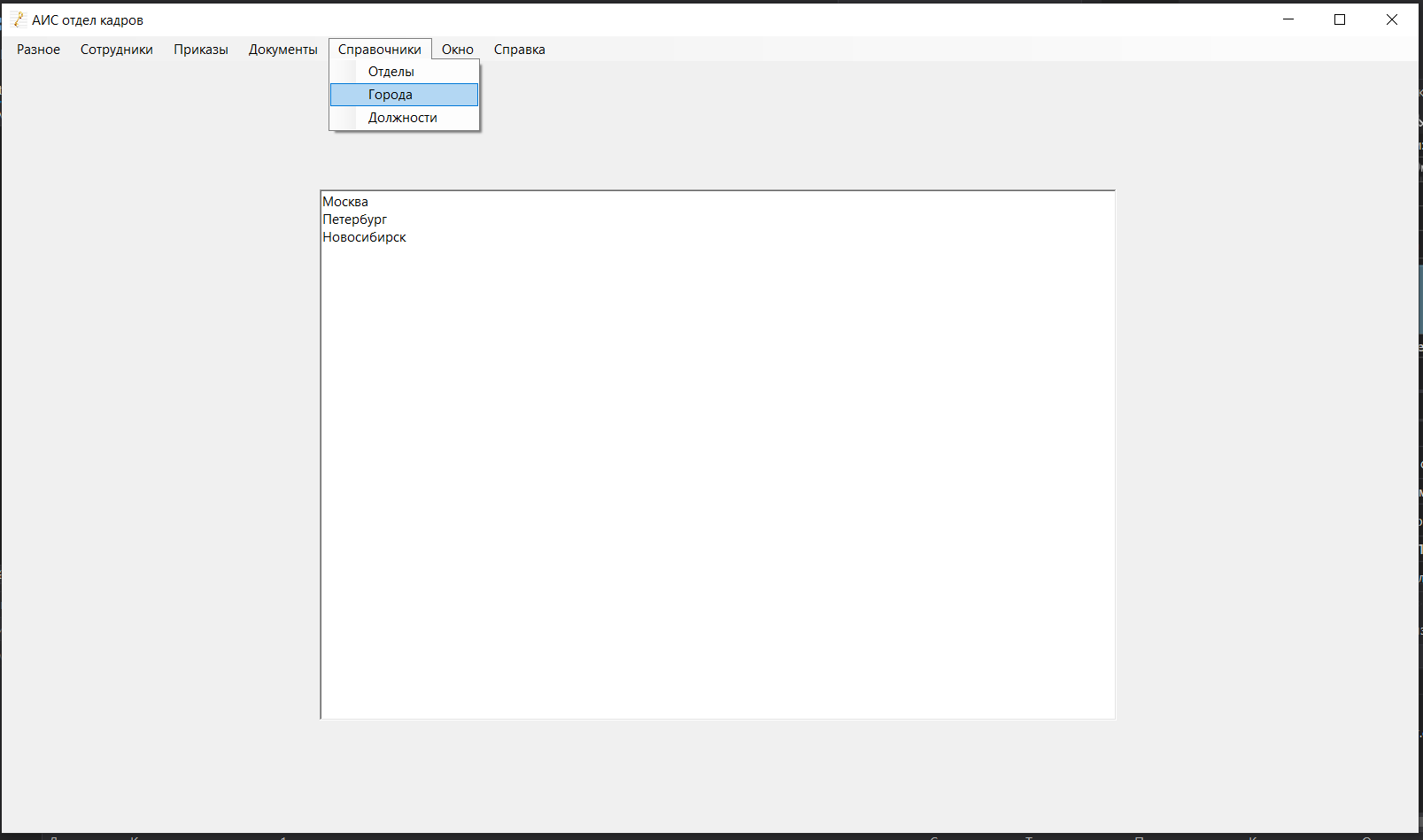
****

Рисунок 1. Экран администратора.

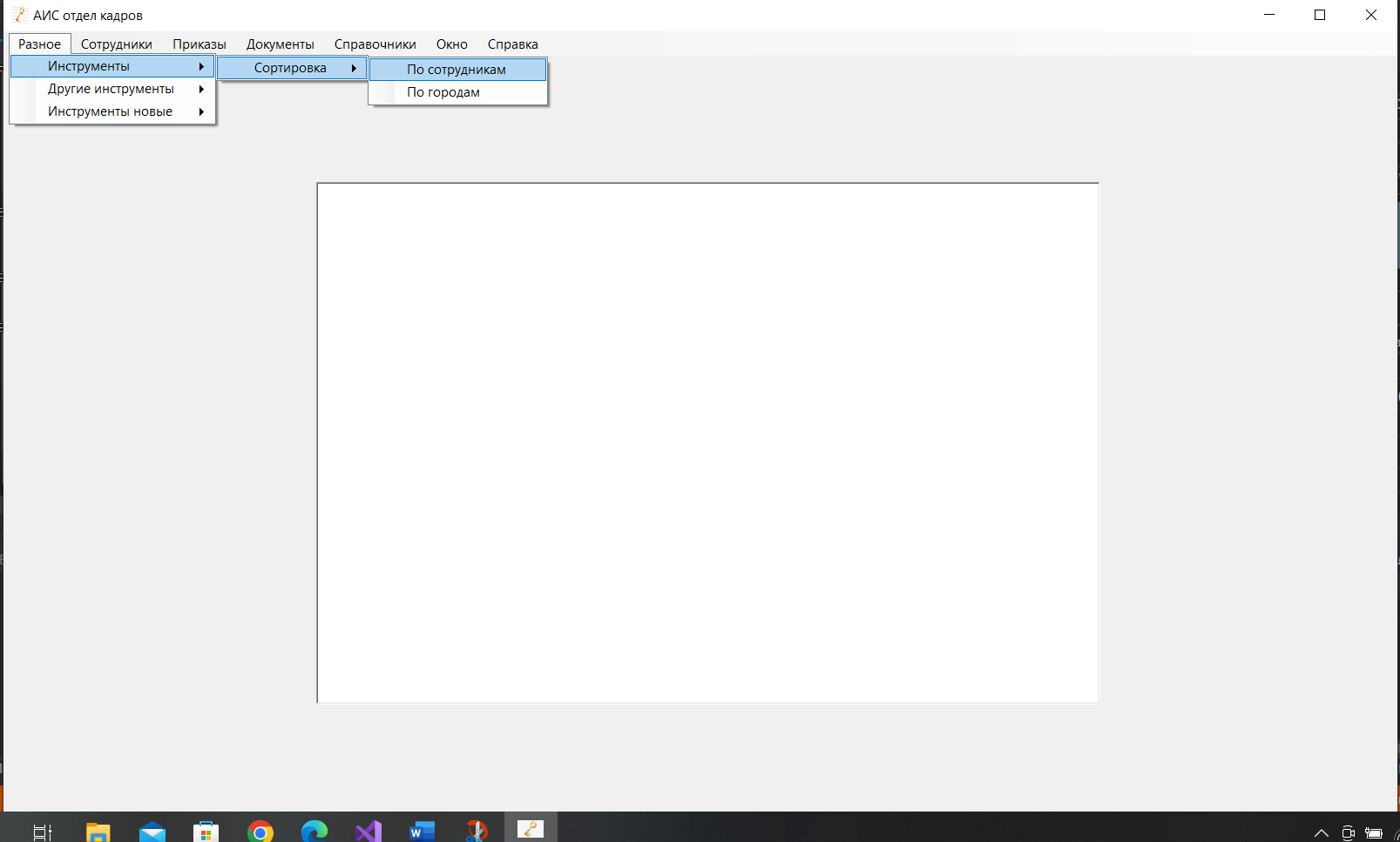
****

Рисунок 2. Экран работника.

**Тестирование программы.**

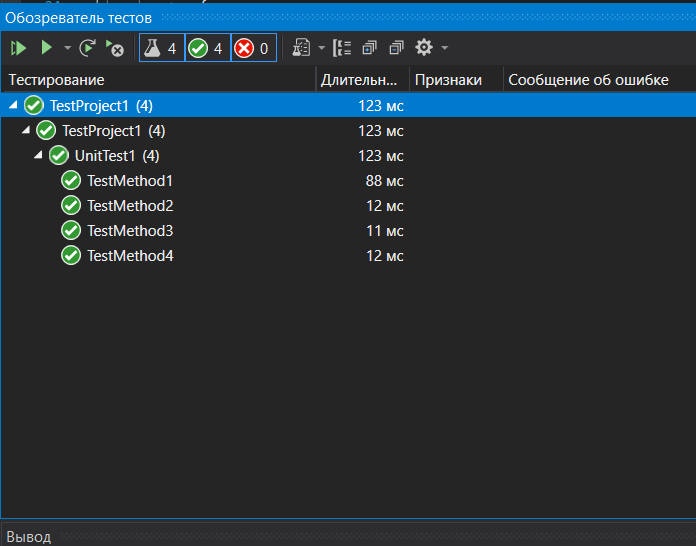
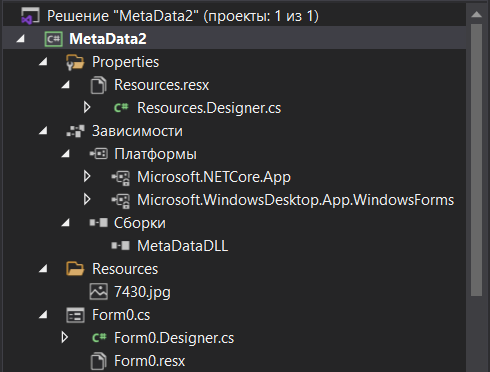
****

Рисунок 3. Тестирование программы.

**Использование библиотеки (dll)**

****