

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Брянский государственный технический университет

**Утверждаю**

**Ректор университета**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.Н.Федонин**

**«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2017 г.**

**Программирование в среде Microsoft .NET**

**Проектирование классов. Коллекции и обобщения**

**Методические указания**

**к выполнению лабораторной работы №5**

**для студентов очной формы обучения по направлениям подготовки 02.03.03 – «Математическое обеспечение и администрирование информационных систем», 09.03.01 – «Информатика и вычислительная техника» и 09.03.04 – «Программная инженерия»**

**Брянск 2017**

УКД 004.43

Программирование в среде Microsoft .Net. Проектирование классов. Коллекции и обобщения [Текст] + [Электронный ресурс]: Методические указания к выполнению лабораторной работы №5 для студентов очной формы обучения по направлениям подготовки 02.03.03 – «Математическое обеспечение и администрирование информационных систем», 09.03.01 – «Информатика и вычислительная техника» и 09.03.04 – «Программная инженерия». – Брянск: БГТУ, 2017. – 14 с.

Разработали:

Д.А.Коростелёв

канд. техн. наук, доц.

Рекомендовано кафедрой «Информатика и программное обеспечение» БГТУ (протокол №2 от 16.09.2016)

Научный редактор А.А.Азарченков

Редактор издательства Л.И.Афонина

Компьютерный набор Д.А.Коростелёв

Темплан 2017 г., п.273

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подписано в печать 23.09.17. Формат 60х84 1/16 Бумага офсетная. Офсетная печать. Усл.печ.л. 1,8. Уч.-изд.л. 1,8 Тираж 1 экз. Заказ Бесплатно.

Издательство Брянского государственного технического университета

241035, Брянск, бульвар 50-летия Октября, 7, БГТУ. 58-82-49.

Лаборатория оперативной полиграфии БГТУ, ул. Институтская, 16.

**Цель работы**

Целью работы является изучение принципов использования коллекций и обобщений при проектировании классов.

Продолжительность работы – 2 часа.

**Проектирование классов**

Рассмотрим использование коллекций при проектировании классов на примере организации работы гостиницы. Информация о клиентах и номерах носит справочный характер, а это значит, что для их хранения предпочтительно использовать коллекции типа «словарь». Для использования словаря необходимо определиться с ключевым полем для соответствующих классов. Для класса «Номер» таковым полем может являться свойство Number. Для класса «Клиент» такого поля нет, но им могло бы быть сочетание нескольких полей. Однако, для класса «Клиент» мы можем воспользоваться другим подходом по формированию ключевого поля – число, возрастающее при каждом обращении к нему (аналогично тому, как в БД существует автоинкремент). Для этих целей будем использовать статическое поле и свойство в классе Client:

namespace ClassLibraryHotel

{

/// <summary>

/// Клиент

/// </summary>

public class Client : IValidatable

{

/// <summary>

/// Уникальный идентификатор нового клиента (аналог автоинкремента)

/// </summary>

private static int \_newClientId;

private static int NewClientId

{

get

{

\_newClientId++;

return \_newClientId;

}

}

/// <summary>

/// Уникальный идентификатор клиента

/// </summary>

public int ClientId { get; }

/// <summary>

/// Имя

/// </summary>

public string FirstName { get; set; } = "";

/// <summary>

/// Отчество

/// </summary>

public string MiddleName { get; set; } = "";

/// <summary>

/// Фамилия

/// </summary>

public string LastName { get; set; } = "";

/// <summary>

/// Паспортные данные

/// </summary>

public PassportInfo Passport { get; set; } = new PassportInfo();

public bool IsValid

{

get

{

if (string.IsNullOrWhiteSpace(FirstName)) return false;

if (string.IsNullOrWhiteSpace(MiddleName)) return false;

if (string.IsNullOrWhiteSpace(LastName)) return false;

if (Passport == null) return false;

return true;

}

}

public Client()

{

ClientId = NewClientId;

}

public Client(string firstName, string lastName, string middleName)

{

FirstName = firstName;

LastName = lastName;

MiddleName = middleName;

ClientId = NewClientId;

}

public override string ToString()

{

return $"Фамилия: {LastName}\r\nИмя: {FirstName}\r\nОтчество: {MiddleName}\r\nПаспортные данные: {Passport}\r\n";

}

}

}

Хранение информации о заселении клиентов в номера будем осуществлять в списке. При этом в качестве классов коллекций будем использовать их обобщенные типизированные версии – Dictionary<TKey,TValue> и List<T> и все коллекции будем хранить как статические свойства в новом классе «Отель» – Hotel:

using System.Collections.Generic;

namespace ClassLibraryHotel

{

/// <summary>

/// Отель

/// </summary>

public class Hotel

{

/// <summary>

/// Словарь клиентов

/// </summary>

public static Dictionary<int, Client> Clients { get; } = new Dictionary<int, Client>();

/// <summary>

/// Словарь комнат

/// </summary>

public static Dictionary<int, Room> Rooms { get; } = new Dictionary<int, Room>();

/// <summary>

/// Спосок поселений

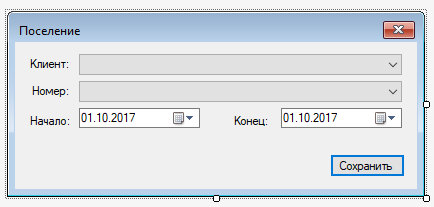
/// </summary>

public static List<Settlement> Settlements { get; } = new List<Settlement>();

}

}

Добавление и редактирование информации о поселении сделаем аналогично работе с клиентами и номерами в отдельной диалоговой форме (рис. 1).



***Рис. 1. Диалоговая форма добавления (редактирования) информации о поселении клиента в номер***

Заполнение элементов управления для выбора клиента и номера реализуем в конструкторе класса диалоговой формы, а обновление информации о поселении в обработчике нажатия кнопки «Сохранить»:

using System;

using System.Windows.Forms;

using ClassLibraryHotel;

namespace Lab05

{

public partial class FormSettlement : Form

{

public Settlement Settlement { get; }

public FormSettlement(Settlement settlement)

{

InitializeComponent();

Settlement = settlement;

foreach (var item in Hotel.Clients)

{

var client = item.Value;

comboBoxClient.Items.Add(client);

}

foreach (var item in Hotel.Rooms)

{

var room = item.Value;

comboBoxRoom.Items.Add(room);

}

comboBoxClient.SelectedItem = settlement.Client;

comboBoxRoom.SelectedItem = settlement.Room;

dateTimePickerStartDate.Value = settlement.StartDate;

dateTimePickerEndDate.Value = settlement.EndDate;

}

private void buttonSave\_Click(object sender, EventArgs e)

{

Settlement.Client = comboBoxClient.SelectedItem as Client;

Settlement.Room = comboBoxRoom.SelectedItem as Room;

Settlement.StartDate = dateTimePickerStartDate.Value;

Settlement.EndDate = dateTimePickerEndDate.Value;

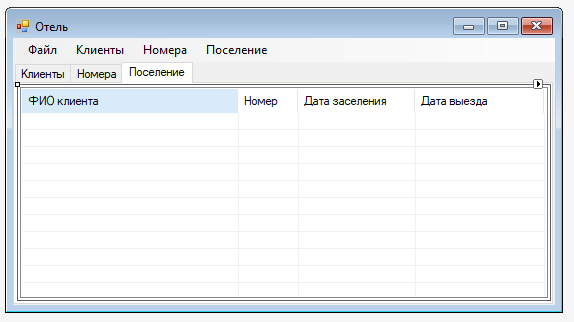
}

}

}

На главной форме программы также внесём соответствующие изменения:

1. Добавим пункты меню для работы с объектами класса «Поселение».
2. Добавим вкладки для отображения и выбора элементов коллекции (рис. 2).



***Рис. 2. Интерфейс главной формы программы***

1. Подкорректируем исходный код обработчиков пунктов главного меню с учетом того, что данные необходимо получать и добавлять в коллекции:

using System;

using System.Windows.Forms;

using ClassLibraryHotel;

namespace Lab05

{

public partial class FormMain : Form

{

public FormMain()

{

InitializeComponent();

}

private void addClientToolStripMenuItem\_Click(object sender, EventArgs e)

{

var client = new Client();

FormClient formClient = new FormClient(client);

if (formClient.ShowDialog() == DialogResult.OK)

{

Hotel.Clients.Add(client.ClientId, client);

UpdateClientsList();

}

}

private void editClientToolStripMenuItem\_Click(object sender, EventArgs e)

{

var client = listViewClients.SelectedItems[0].Tag as Client;

FormClient formClient = new FormClient(client);

if (formClient.ShowDialog() == DialogResult.OK)

{

UpdateClientsList();

}

}

private void UpdateClientsList()

{

listViewClients.Items.Clear();

foreach (var item in Hotel.Clients)

{

var client = item.Value;

var listViewItem = new ListViewItem

{

Tag = client,

Text = client.ToString()

};

listViewClients.Items.Add(listViewItem);

}

}

private void addRoomToolStripMenuItem\_Click(object sender, EventArgs e)

{

var room = new Room();

FormRoom formRoom = new FormRoom(room);

if (formRoom.ShowDialog() == DialogResult.OK)

{

Hotel.Rooms.Add(room.Number, room);

UpdateRoomsList();

}

}

private void editRoomToolStripMenuItem\_Click(object sender, EventArgs e)

{

var room = listViewRooms.SelectedItems[0].Tag as Room;

FormRoom formRoom = new FormRoom(room);

if (formRoom.ShowDialog() == DialogResult.OK)

{

UpdateRoomsList();

}

}

private void UpdateRoomsList()

{

listViewRooms.Items.Clear();

foreach (var item in Hotel.Rooms)

{

var room = item.Value;

var listViewItem = new ListViewItem

{

Tag = room,

Text = room.ToString()

};

listViewRooms.Items.Add(listViewItem);

}

}

private void addSettlementToolStripMenuItem\_Click(object sender, EventArgs e)

{

var settlement = new Settlement();

FormSettlement formSettlement = new FormSettlement(settlement);

if (formSettlement.ShowDialog() == DialogResult.OK)

{

Hotel.Settlements.Add(settlement);

UpdateSettlementList();

}

}

private void editSettlementToolStripMenuItem\_Click(object sender, EventArgs e)

{

var settlement = listViewSettlement.SelectedItems[0].Tag as Settlement;

FormSettlement formSettlement = new FormSettlement(settlement);

if (formSettlement.ShowDialog() == DialogResult.OK)

{

UpdateSettlementList();

}

}

private void UpdateSettlementList()

{

listViewSettlement.Items.Clear();

foreach (var settlement in Hotel.Settlements)

{

var listViewItem = new ListViewItem

{

Tag = settlement,

Text = settlement.Client.ToString()

};

listViewItem.SubItems.Add(settlement.Room.ToString());

listViewItem.SubItems.Add(settlement.StartDate.ToShortDateString());

listViewItem.SubItems.Add(settlement.EndDate.ToShortDateString());

listViewSettlement.Items.Add(listViewItem);

}

}

}

}

**Задание на лабораторную работу**

1. В разработанной в предыдущей лабораторной работе программе добавьте поддержку хранения коллекций объектов первых двух классов в виде словарей (Dictionary). Для этой цели разработайте новый класс, а сами коллекции хранить в виде статических свойств. При необходимости внесите изменения в первые два класса, чтобы у них были ключевые поля-свойства.
2. Добавьте в новый класс для хранения объектов 3-го класса список (тоже как статическое свойство).
3. В приложение Windows Forms добавьте диалоговую форму для создания и редактирования списка объектов 3-го класса аналогично диалоговым формам для первых двух классов.
4. На главную форму добавьте 3 вкладки, в которых будет отображаться коллекции соответствующих классов с помощью элементов управления ListView. Настройте столбцы и внешний вид элементов управления ListView по аналогии с рассмотренным примером.
5. При вызове события редактирования объектов классов сам объект необходимо определять на основе выбранной строки в соответствующем элементе управления ListView.
6. При добавлении и редактировании объектов трех классов необходимо обновлять содержимое ListView.

**Контрольные вопросы**

1. Какие существуют классы стандартных коллекций?
2. Какие методы содержит интерфейс IComparable?
3. Какие методы описывает интерфейс IList?
4. Что такое обобщение и как оно объявляется?

# Список рекомендуемой литературы

1. Нейгел, К. C# 5.0 и платформа .NET 4.5 для профессионалов / К.Нейгел, Б.Ивьен, Дж.Глинн, М.Скиннер, К.Уотсон. – М.: Диалектика, 2013. – 1440 с.
2. Неш, Т. C# 2010: ускоренный курс для профессионалов / Т.Неш – М.: Вильямс, 2010. – 592 с.
3. Снелл, М. Microsoft Visual Studio 2008 / М.Снелл, Л.Пауэрс. – СПб: БХВ-Петербург, 2009. – 1200 с.
4. Троелсен, Э. Язык программирования С# 5.0 и платформа .NET 4.5 / Э.Троелсен. – 6-е изд. – М.: Вильямс, 2015. – 1312 с.
5. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж.Рихтер. – СПб.: Питер, 2016. – 896 с.