Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 19

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Запросы LINQ»

Выполнил: ст. гр. ТУУ-211

Кругликов Е.А.

Вариант №8

Проверил: к.т.н., доц. Сафронов А.И.

1. **Формулировка цели работы**

Освоить навыки подготовки наборов данных по нечётко сформулированному описанию предметной области при учёте особенностей запросов, отклики на которые необходимо предоставлять заявленным (потенциальным) пользователям программного обеспечения.

1. **Описание задачи**

Описание предметной области для подготовки данных. Пользователями являются сотрудники гостиницы и клиенты-постояльцы. В гостинице имеются номера, которые различаются по количеству мест, количеству комнат, удобствам и т.д. От всех этих параметров и их сочетания зависит цена гостиничного номера. Посетители могут занимать номера, бронировать их заранее. Постоянным посетителям и детям до 12 лет с родителями предоставляется скидка.

Реализовать запросы LINQ:

– предоставлять (и бронировать) посетителям номер в гостинице в зависимости от их потребностей (количества мест для размещения группы гостей и т.д.);

– показывать количество свободных (занятых) мест по категории номеров указанного типа;

– показывать список постоянных посетителей и предоставляемую им скидку;

– показывать информацию по указанному номеру (количество мест, комнат, наличие холодильника, телевизора, бара и т.д.).

1. **Сеть Петри запрограммированного технологического процесса**

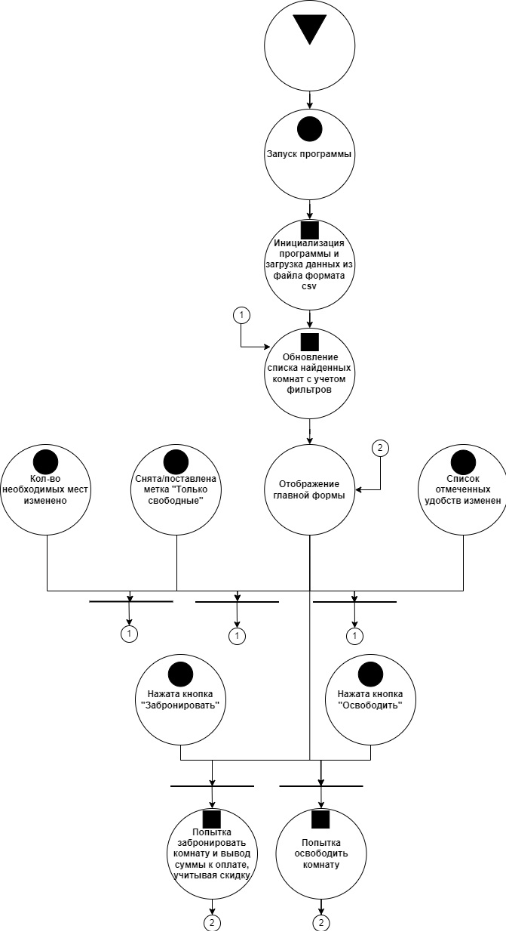


Рисунок 1 – Сеть Петри запрограммированного технологического процесса

1. **Схемы алгоритмов методов в составе решения, отмеченных на сети Петри в качестве «эффектов» (метка )**

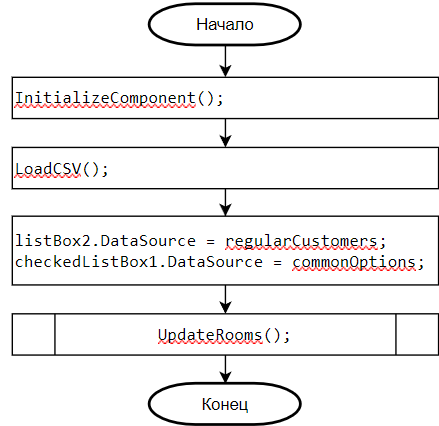


Рисунок 2 – Инициализация программы и загрузка данных из csv-файла

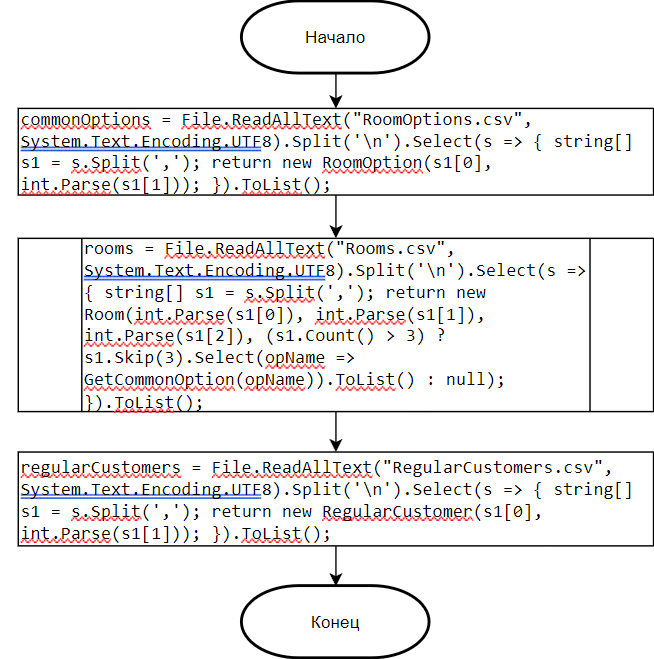


Рисунок 3 – Метод LoadCSV

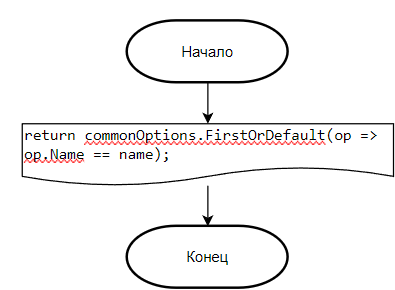


Рисунок 4 – Метод GetCommonOption

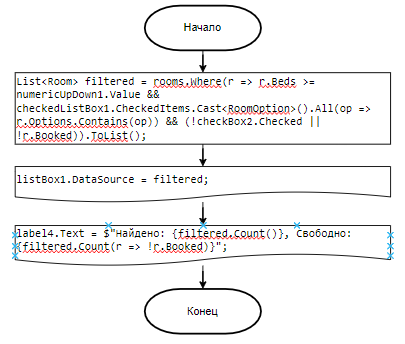


Рисунок 5 – Обновление списка найденных комнат в соответствии с фильтрами

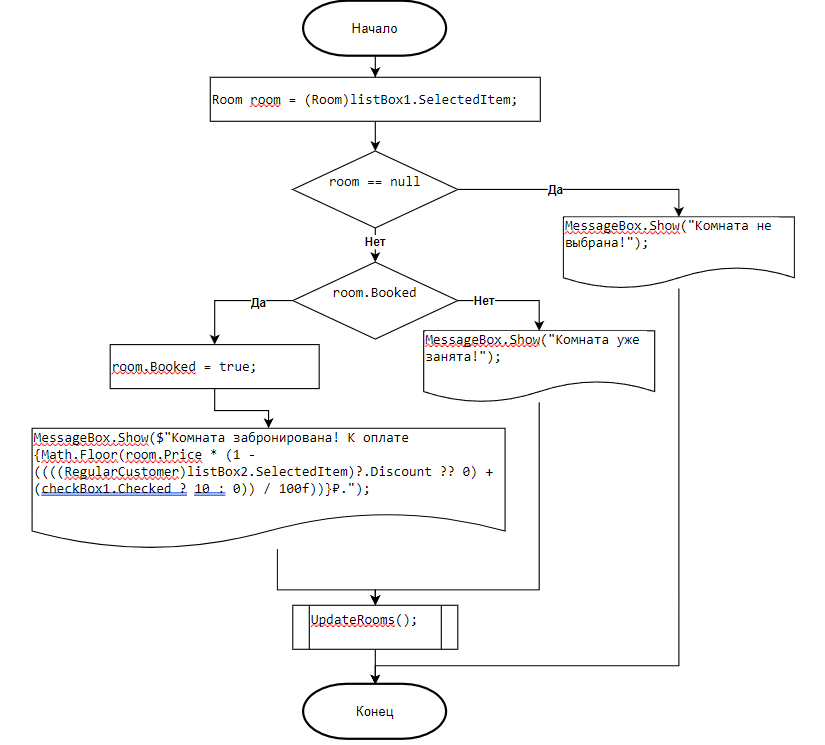


Рисунок 6 – Попытка забронировать комнату и вывод суммы к оплате с учетом скидки

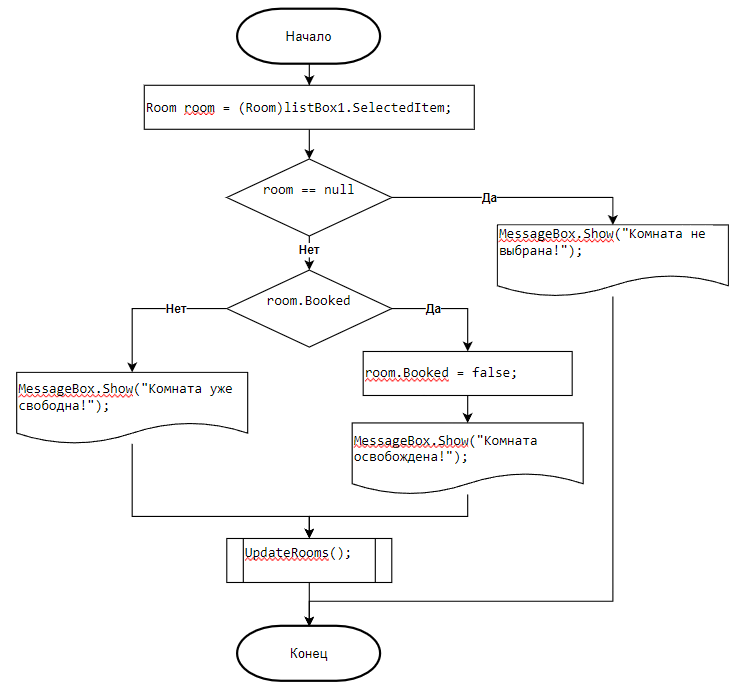


Рисунок 7 – Попытка освободить комнату

1. **Подбор тестовых примеров**

На основании задания были определены две основные сущности программы: постоянный клиент и номер отеля. В целях упрощения расчетов и работы с сущностью «номер отеля» была введена дополнительная сущность – «удобство номера». В программе были созданы соответствующие классы: RegularCustomer, Room и RoomOption. Также для хранения данных этих классов были созданы и заполнены тестовыми данными csv-файлы:

Файл RegularCustomers.csv:

нет,0

Иванов И.И.,50

Андреев Ф.Н.,15

Анатольев Ф.А.,12

Музыченко Н.Б.,15

Петренко М.О.,7

Файл Rooms.csv:

1,2,1

2,2,1

3,2,1

4,2,1,Холодильник,Балкон

5,3,2,Минибар,Балкон

6,2,1,Холодильник,Балкон,Вид на море

7,4,3,Минибар,Балкон,Джакузи

Файл RoomOptions.csv:

Холодильник,500

Минибар,1500

Балкон,2000

Джакузи,1000

Телевизор,200

Вид на море,2000

1. **Листинг (код) составленного программного обеспечения**

**Файл Form1.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Windows.Forms;

namespace LINQ\_querries

{

public partial class MainForm : Form

{

static List<RoomOption> commonOptions;

List<Room> rooms;

List<RegularCustomer> regularCustomers;

public MainForm()

{

InitializeComponent();

LoadCSV();

listBox2.DataSource = regularCustomers;

checkedListBox1.DataSource = commonOptions;

UpdateRooms();

}

public static RoomOption GetCommonOption(string name)

{

return commonOptions.FirstOrDefault(op => op.Name == name);

}

private void LoadCSV()

{

commonOptions = File.ReadAllText("RoomOptions.csv", System.Text.Encoding.UTF8).Split('\n').Select(s => { string[] s1 = s.Split(','); return new RoomOption(s1[0], int.Parse(s1[1])); }).ToList();

rooms = File.ReadAllText("Rooms.csv", System.Text.Encoding.UTF8).Split('\n').Select(s => { string[] s1 = s.Split(','); return new Room(int.Parse(s1[0]), int.Parse(s1[1]), int.Parse(s1[2]), (s1.Count() > 3) ? s1.Skip(3).Select(opName => GetCommonOption(opName)).ToList() : null); }).ToList();

regularCustomers = File.ReadAllText("RegularCustomers.csv", System.Text.Encoding.UTF8).Split('\n').Select(s => { string[] s1 = s.Split(','); return new RegularCustomer(s1[0], int.Parse(s1[1])); }).ToList();

}

private void UpdateRooms()

{

List<Room> filtered = rooms.Where(r => r.Beds >= numericUpDown1.Value && checkedListBox1.CheckedItems.Cast<RoomOption>().All(op => r.Options.Contains(op)) && (!checkBox2.Checked || !r.Booked)).ToList();

listBox1.DataSource = filtered;

label4.Text = $"Найдено: {filtered.Count()}, Свободно: {filtered.Count(r => !r.Booked)}";

}

private void checkBox2\_CheckedChanged(object sender, EventArgs e)

{

UpdateRooms();

}

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

UpdateRooms();

}

private void checkedListBox1\_MouseMove(object sender, MouseEventArgs e)

{

UpdateRooms();

}

private void button1\_Click(object sender, EventArgs e)

{

Room room = (Room)listBox1.SelectedItem;

if (room == null)

{

MessageBox.Show("Комната не выбрана!");

return;

}

if (room.Booked)

{

room.Booked = false;

MessageBox.Show("Комната освобождена!");

}

else

MessageBox.Show("Комната уже свободна!");

UpdateRooms();

}

private void button2\_Click(object sender, EventArgs e)

{

Room room = (Room)listBox1.SelectedItem;

if (room == null)

{

MessageBox.Show("Комната не выбрана!");

return;

}

if (room.Booked)

MessageBox.Show("Комната уже занята!");

else

{

room.Booked = true;

MessageBox.Show($"Комната забронирована! К оплате {Math.Floor(room.Price \* (1 - ((((RegularCustomer)listBox2.SelectedItem)?.Discount ?? 0) + (checkBox1.Checked ? 10 : 0)) / 100f))}₽.");

}

UpdateRooms();

}

}

}

**Файл RegularCustomer.cs**

namespace LINQ\_querries

{

class RegularCustomer

{

public readonly string Name;

public int Discount { get; private set; }

public RegularCustomer(string name, int discount)

{

Name = name;

Discount = discount;

}

public override string ToString()

{

return $"{Name} со скидкой {Discount}%" ;

}

}

}

**Файл Room.cs**

using System.Collections.Generic;

using System.Linq;

namespace LINQ\_querries

{

public class Room

{

private const int basePrice = 500;

public List<RoomOption> Options = new List<RoomOption>() { MainForm.GetCommonOption("Телевизор") };

public readonly int Number;

public int Beds { get; private set; }

public readonly int Rooms;

public int Price => ((basePrice + Options.Sum(op => op.Price)) \* Rooms) + basePrice \* Beds;

public bool Booked;

public Room(int number, int beds = 2, int rooms = 1, List<RoomOption> options = null)

{

Number = number;

Beds = beds;

Rooms = rooms;

if (options != null)

Options = options;

}

public override string ToString()

{

return $"№{Number}. {Beds}-местный, {Rooms}-комнатный. Цена: {Price}₽. Удобства: " + Options.Select(op => op.Name).Aggregate((n1, n2) => n1 + ", " + n2) + ". " +(Booked ? "Занят" : "Свободен");

}

}

}

**Файл RoomOption.cs**

namespace LINQ\_querries

{

public class RoomOption

{

public readonly string Name;

public int Price { get; private set; }

public RoomOption(string name, int price)

{

Name = name;

Price = price;

}

public override string ToString()

{

return $"{Name} за {Price}₽";

}

}

}

1. **Графический пользовательский интерфейс программного обеспечения и его описание**

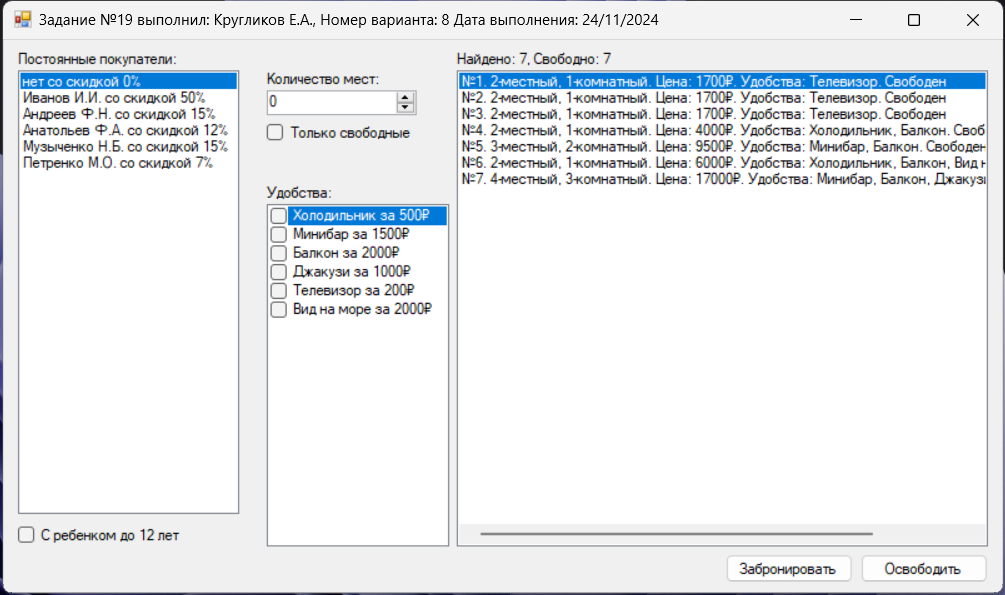


Рисунок 3 – Интерфейс главной формы

В центре формы находятся фильтры для поиска нужного номера: минимальное количество мест в номере, чекбокс для показа только свободных номеров, а также список необходимых удобств.

В правой части формы находится список всех номеров, соответствующих критериям с полным описанием номера. Также внизу справа находятся кнопки для брони/освобождения выбранного номера.

В левой части находится список постоянных покупателей и чекбокс «С ребенком до 12 лет». При выборе постоянного покупателя из списка при оплате рассчитывается соответствующая скидка. Также при отметке чекбокса есть дополнительная скидка 10%. Сумма после применения скидок выводится во всплывающем сообщении при нажатии на кнопку «Забронировать»:

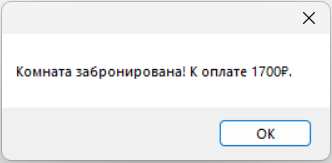


Рисунок 4 – Сообщение с суммой оплаты при выборе номера №1.

1. **Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению**

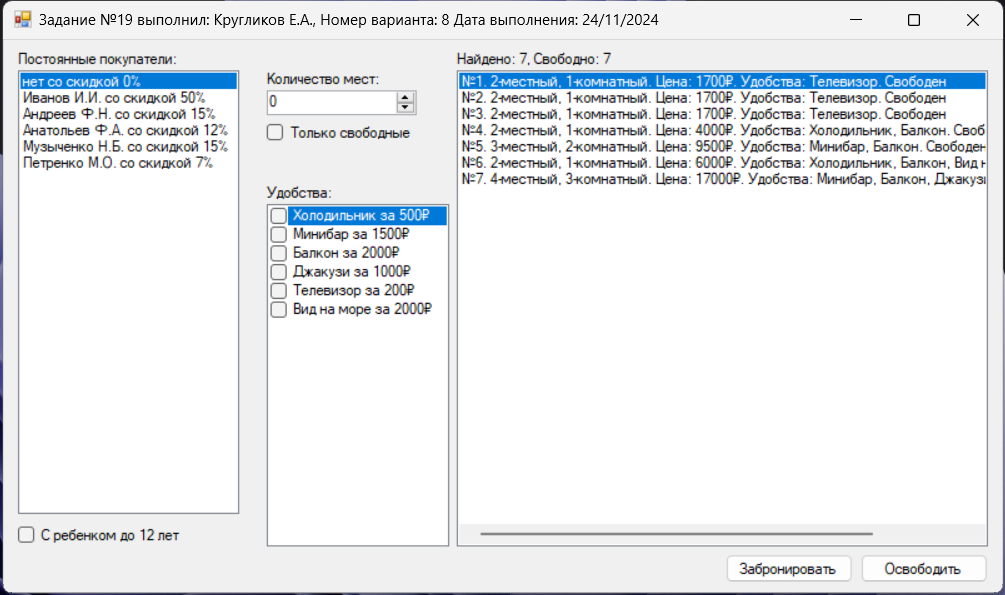


Рисунок 4 – Подтверждение соответствия графического пользовательского интерфейса требованиям к оформлению.

1. Экранный список в правой части формы и настройки фильтров позволяют предоставлять (и бронировать) посетителям номер в гостинице в зависимости от их потребностей (количества мест для размещения группы гостей и т.д.);
2. Они же позволяют показывать количество свободных (занятых) мест по категории номеров указанного типа;
3. Экранный список в левой части формы позволяет показывать список постоянных посетителей и предоставляемую им скидку;
4. В правом экранном списке выведены все номера, что позволяет показывать информацию по указанному номеру (количество мест, комнат, наличие холодильника, телевизора, бара и т.д.).

Таким образом, разработанный графический интерфейс реализует все требования, изложенные в задании к разработке.

1. **Расчёт тестовых примеров с использованием составленного программного обеспечения**

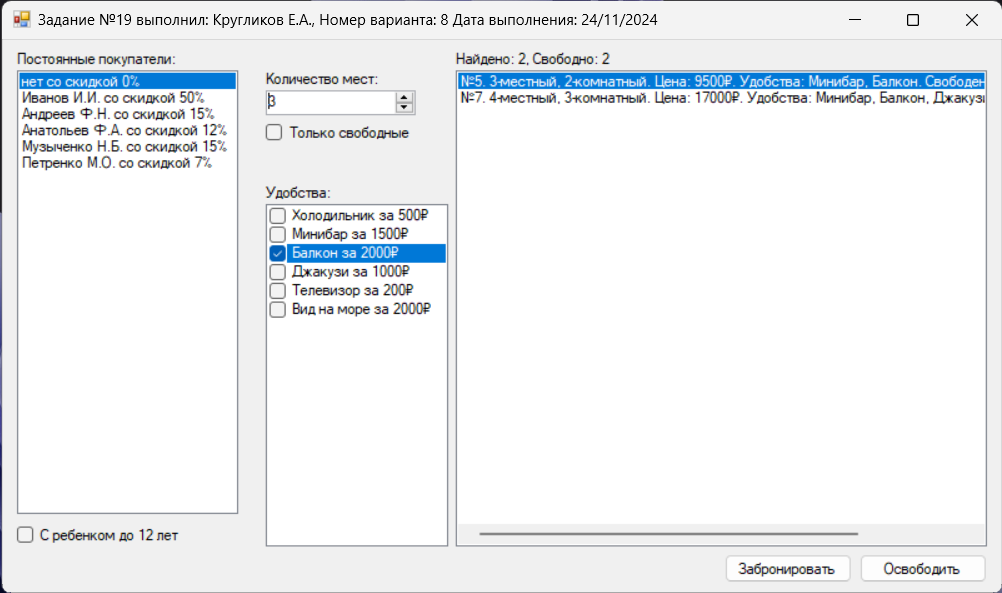


Рисунок 5 – Демонстрация работы фильтра по количеству мест и необходимым удобствам

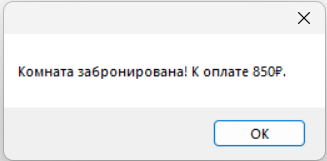
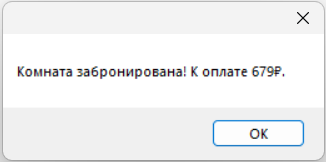
 

Рисунок 4 – Сообщение с суммой оплаты при выборе номера №1 и выборе покупателя «Иванов И.И.» (скидка 50%) с отметкой «С ребенком до 12 лет» (справа) и без (слева).

1. **Формулировка вывода о проделанной работе**

Освоил навыки подготовки наборов данных по нечётко сформулированному описанию предметной области при учёте особенностей запросов, отклики на которые необходимо предоставлять заявленным (потенциальным) пользователям программного обеспечения.

Научился выделять основные и второстепенные сущности и логику их взаимодействия в предметной области. Освоил навыки переноса этих сущностей и взаимодействий в логику программы.