**АННОТАЦИЯ**

Данный курсовой проект посвящён разработке программы для конструирования путешествий.

Проделанная работа актуальна для компаний в туристической сфере.

Курсовой проект содержит пояснительную записку объёмом 62 страницы, включая 11 рисунков, 7 приложений, библиографический список из 5 наименований.

**ABSTRACT**

This course project is dedicated to the development of a program for planning a tour.

The work done is relevant for companies in the tourism sector.

The course project consists of 62 pages, including 11 pictures, 7 appendices and 5 references.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 5](#_Toc104281999)

[1. МОДЕЛИРОВАНИЕ ПРОЕКТА ПРИЛОЖЕНИЯ 6](#_Toc104282000)

[1.1. ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ 6](#_Toc104282001)

[1.2. КЛАССОВАЯ СТРУКТУРА 6](#_Toc104282002)

[1.3. СОСТАВ ФАЙЛОВ ПРОГРАММЫ 10](#_Toc104282003)

[1.4. СОСТАВ ФАЙЛОВ ДЛЯ ИМПОРТА ДАННЫХ 11](#_Toc104282004)

[2. ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ «JAVA» 11](#_Toc104282005)

[2.1. СОДЕРЖИМОЕ ФАЙЛА «HELLOAPPLICATION.JAVA» 11](#_Toc104282006)

[2.2. СОДЕРЖИМОЕ ФАЙЛА «F1\_MAINMENU.JAVA» 11](#_Toc104282007)

[2.3. СОДЕРЖИМОЕ ФАЙЛА «F2\_PLAN.JAVA» 13](#_Toc104282008)

[2.4. СОДЕРЖИМОЕ ФАЙЛА «F3\_CHARACTOFHOTELS.JAVA» 22](#_Toc104282009)

[2.5. СОДЕРЖИМОЕ ФАЙЛА «F4\_SHOWHOTEL.JAVA» 25](#_Toc104282010)

[2.6. СОДЕРЖИМОЕ ФАЙЛА «F5\_MUSEUMSANDTHEATRES.JAVA» 28](#_Toc104282011)

[2.7. СОДЕРЖИМОЕ ФАЙЛА «F6\_FINALSUM.JAVA» 30](#_Toc104282012)

[2.8. СОДЕРЖИМОЕ ФАЙЛА «INFORMATION.JAVA» 34](#_Toc104282013)

[2.9. СОДЕРЖИМОЕ ФАЙЛА «CITY.JAVA» 35](#_Toc104282014)

[2.10. СОДЕРЖИМОЕ ФАЙЛА «CONSTRJOURNEY.JAVA» 40](#_Toc104282015)

[2.11. СОДЕРЖИМОЕ ФАЙЛА «CLASSFOROUTPUT.JAVA» 45](#_Toc104282016)

[3. ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ 46](#_Toc104282017)

[ЗАКЛЮЧЕНИЕ 52](#_Toc104282018)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 53](#_Toc104282019)

[ПРИЛОЖЕНИЯ 54](#_Toc104282020)

# ВВЕДЕНИЕ

Многие люди любят путешествовать. Составить план путешествия не всегда просто, особенно в случае составной поездки. Не всегда можно быстро и просто найти ответы: как лучше добраться до места назначения, какой выбрать отель, мероприятия для развлечений и отдыха и т.д. Проблема планирования путешествий достаточно актуальна в наши дни. Спланировать оптимальный маршрут по городам, странам может помочь конструктор путешествий. С помощью конструктора можно составить маршрут движения из пункта а в пункт б, выбрать отель по предпочтениям, запланировать походы в театр, музей, выставку.

Целью курсового проекта является разработка программы для планирования маршрута путешествия по разным городам.

Задачи курсового проекта:

1. Определение параметров путешествия
2. Определение классов программы
3. Определение функций для доступа к зарытым переменным класса
4. Создание графического интерфейса

# МОДЕЛИРОВАНИЕ ПРОЕКТА ПРИЛОЖЕНИЯ

## **ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ**

Данная программа является конструктором для путешествий. Она позволяет спланировать маршрут для поездки в такие города как Берлин, Париж и Рим. В начале пользователь выбирает город назначения, затем дату вылета (предусмотрена защита от неактуальной даты(минимальная дата которая доступна для выбора – текущая)), далее вводит количество дней, которое хочет отвести для посещения города (минимальное количество дней равно 3) (предусмотрена защита от неверного ввода данных) и количество человек, участвующих в поездке (количество человек влияет на количество номеров необходимых для бронирования в отеле (1 номер предлагается для 3 человек)) (предусмотрена защита от неверного ввода данных). Затем пользователь выбирает класс мест в самолёте, на котором хочет лететь в выбранный им город. Затем пользователь отбирает отель по характеристикам, таким как количество звёзд, наличие бассейна, наличие спортзала и типа питания. Далее пользователю предлагается лучший отель по его характеристикам с самой низкой ценой. Затем предоставляется возможность увидеть фотографию отобранного для пользователя отеля и посмотреть его характеристики, по которым данный отель был отобран. Затем предлагается выбор музеев и театров для посещения в городе. Далее формируется таблица с расходами, на которой подробно расписаны компоненты поездки и их стоимость. По завершении подсчёта итогов пользователю предлагается завершить работу программы или начать сначала.

## **КЛАССОВАЯ СТРУКТУРА**

Программа состоит из 11 классов:

Классы для хранения данных

* Класс «public class city»
* Класс «public class constrjourney»
* Класс «public class classforoutput»

Классы для взаимодействия формами

* Класс «public class f1\_mainmenu»
* Класс «public class information»
* Класс «public class f2\_plan»
* Класс «public class f3\_charactofhotels»
* Класс «public class f4\_showhotel»
* Класс «public class f5\_museumsandtheatres»
* Класс «public class f6\_finalsum»

Основной класс

Класс «public class HelloApplication extends Application»

Описание классов

Класс ««public class city»»

В данном классе созданы поля для хранения всех данных о городе.

Описание полей класса

* «static private ArrayList <String> nameofhotel = new ArrayList<String>()» – коллекция, показывающая названия отелей
* «static private ArrayList <Integer> stars = new ArrayList<Integer>()» – коллекция, показывающая количество звёзд у отеля
* «static private ArrayList <Boolean> pool = new ArrayList<Boolean>()» – коллекция, показывающая наличие бассейна
* «static private ArrayList <Boolean> gym = new ArrayList<Boolean>()» – коллекция, показывающая наличие спортзала в отеле
* «static private ArrayList<String> typeoffood = new ArrayList<String>()» – коллекция показывающая доступный тип еды в отеле
* «static private ArrayList <Integer> pricefortypeoffood = new ArrayList<Integer>()» – коллекция, показывающая стоимость еды в зависимости от её типа
* «static private ArrayList <Integer> costperday = new ArrayList<Integer>()» – коллекция, показывающая стоимость номера отеля за день
* «static private ArrayList <String> nameoftheatre= new ArrayList<String>()» – коллекция показывающая названия театров
* «static private ArrayList <Integer> ticketpriceoftheatre = new ArrayList<Integer>()» – коллекция, показывающая стоимость билетов в театр
* «static private ArrayList <String> nameofmuseum= new ArrayList<String>()» – коллекция, показывающая названия музеев
* «static private ArrayList <Integer> ticketpriceofmuseum = new ArrayList<Integer>()» – коллекция, показывающая стоимость билетов в музей
* «static private ArrayList <String> nameofclassplane= new ArrayList<String>()» – коллекция показывающая название класса мест самолёта
* «static private ArrayList <Integer> ticketpriceofplane= new ArrayList<Integer>()» – коллекция, показывающая стоимость билетов на различные классы мест самолёта
* «static private ArrayList <Integer> pricefortransfertohotel = new ArrayList<Integer>()» – коллекция, показывающая стоимость трансфера от аэропорта до отеля

Класс «public class constrjourney»

В данном классе созданы поля для хранения информации о маршруте пользователя.

Описание полей класса

* «static private String city» – город назначения пользователя
* «static private LocalDate datebegintrip» – дата начала путешествия
* «static private LocalDate dateendtrip» – дата завершения путешествия
* «static private int koldays» – количество дней поездки
* «static private int kolperson» – количество человек, участвующих в поездке
* «static private int kolrooms – необходимое количество номеров для размещения всех человек
* «static private String nameofhotel» – название отобранного отеля для пользователя
* «static private int stars» – количество звёзд у отобранного отеля
* «static private boolean pool» – наличие бассейна в отеле
* «static private boolean gym» – наличие спортзала в отеле
* «static private String typeoffood» – тип еды выбранный пользователем
* «static private int pricefortypeoffood» – стоимость отобранного пользователем типа еды
* «static private int costperday» – стоимость отобранного для пользователя номера в день
* «static private String typeofclassplane» – класс самолёта выбранный пользователем
* «static private int ticketpriceofplane» – цена места для выбранного пользователем класса
* «static private ArrayList<String> nameoftheatre= new ArrayList<String>()» – названия театров, которые пользователь собирается посетить
* «static private ArrayList <Integer> ticketpriceoftheatre = new ArrayList<Integer>()» – стоимость билетов в театр, который пользователь собирается посетить
* «static private ArrayList <String> nameofmuseum= new ArrayList<String>()» – названия музеев, которые пользователь собирается посетить
* «static private ArrayList <Integer> ticketpriceofmuseum = new ArrayList<Integer>()» – стоимость билетов в музеи, которые пользователь собирается посетить
* «static private int sumofticketsplane» – стоимость билетов на самолёт для всех человек, участвующих в поездке
* «static private int sumfortranser» – стоимость трансфера от аэропорта до отеля для всех человек, участвующих в поездке
* «static private int sumforhotel» – стоимость номеров в отеле для всех человек, участвующих в поездке
* «static private int sumforfood» – стоимость еды в отеле для всех человек, участвующих в поездке
* «tatic private ArrayList <Integer> sumeachtheatre» – коллекция, показывающая суммарную стоимость билетов для каждого театра
* «static private ArrayList <Integer> sumeachmuseum=new ArrayList<Integer>()» – коллекция, показывающая суммарную стоимость билетов для каждого музея
* «static private int sum\_full» – итоговая стоимость всей поездки

Класс «public class classforoutput»

Данный класс помогает вывести итоговую свободную таблицу компонентов маршрута.

Класс «public class f1\_mainmenu»

Класс для управления формой, в которой находится главное меню. Связан с файлом «f1\_mainmenu.fxml».

Класс «f2\_plan»

Класс для управления формой, в которой планируются организационные элементы маршрута. Связан с файлом «f2\_plan.fxml». В этом классе реализуется импорт данных отелей, музеев и театров из файлов с помощью метода «public void getdatafromfile() throws IOException». Также в этом классе задаются элементы маршрута, такие как город для поездки, дата начала поездки, дата окончания поездки, количество дней поездки, тип класса мест в самолёте с помощью метода «setdataforjourney».

Класс «public class f3\_charactofhotels»

Класс для управления формой, в которой выбираются критерии отбора отеля. Связан с файлом «f3\_charactofhotels.fxml».

Класс «public class f4\_showhotel»

Данный класс используется для показа данных об отобранном отеле. Связан с файлом «f4\_showhotels.fxml».

Класс «public class f5\_museumsandtheatres»

В данном классе пользователь выбирает музеи и театры, которые он хочет посетить. Связан с файлом «f5\_museumsandtheatres.fxml».

Класс «public class f6\_finalsum»

В данном классе реализуется расчёт необходимой суммы для оплаты путешествия и отображается информация о путешествии. Связан с файлом «f6\_finalsum.fxml».

Класс «public class information»

В данном классе отображается справка по программе. Связан с файлом «information.fxml».

Класс «public class HelloApplication extends Application»

Данный класс создаёт сцену для запуска программы и открывает файл «f1\_mainmenu.fxml», в котором содержится главное меню программы.

## **СОСТАВ ФАЙЛОВ ПРОГРАММЫ**

Программа состоит из 18 файлов: 11 классов, каждый в отдельном файле с расширением «.java» и 7 файлов с разрешением «.fxml»:

* «f1\_mainmenu.fxml» – главное меню
* «f2\_plan.fxml» – ввод организационных элементов поездки
* «f3\_charactofhotels.fxml» – выбор характеристик для отеля
* «f4\_showhotel.fxml» – демонстрация выбранного отеля
* «f5\_museumsandtheatres.fxml» – выбор музеев и театров для посещения
* «f6\_finalsum» – демонстрация итоговых расчётов и элементов поездки

## **СОСТАВ ФАЙЛОВ ДЛЯ ИМПОРТА ДАННЫХ**

Для каждого города в программе предусмотрена 4 файла.

* Файл «hotelsof”название города”.txt» – отели и их характеристики
* Файл «museumsof”название города”» – музеи и стоимость их посещения
* Файл «theatresof”название города”» – театры и стоимость их посещения
* Файл «planesticketpriceof”название города”» – тип класса мест в самолёте и их стоимость

# ТЕКСТ ПРОГРАММЫ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ «JAVA»

## **СОДЕРЖИМОЕ ФАЙЛА «HELLOAPPLICATION.JAVA»**

package com.example.kurs;  
  
import javafx.application.Application;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Alert;  
import javafx.scene.control.ButtonType;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
  
public class HelloApplication extends Application {  
 @Override  
 public void start(Stage stage) throws IOException {  
 FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("f1\_mainmenu.fxml"));  
 Scene scene = new Scene(fxmlLoader.load(), 700, 700);  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(scene);  
  
 stage.show();  
  
 }  
  
  
  
 public static void main(String[] args) {*launch*();}  
  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «F1\_MAINMENU.JAVA»**

package com.example.kurs;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.AnchorPane;  
import javafx.stage.Modality;  
import javafx.stage.Stage;  
import javafx.stage.Window;  
  
import java.io.\*;  
import java.net.URL;  
import java.util.ResourceBundle;  
  
public class f1\_mainmenu {  
  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button button\_begin;  
  
 @FXML  
 private Button button\_finish;  
  
 @FXML  
 private Button button\_inform;  
  
 @FXML  
 private ImageView field\_picture;  
  
 @FXML  
 public void initialize() throws IOException {  
 //добавление фото на форму  
 File mainpicture= new File("E:\\Курсовой проект\\mainmenu.jpg");  
 Image image = new Image(mainpicture.toURI().toString());  
 field\_picture.setImage(image);  
 }  
 @FXML  
 public void click\_button\_begin(ActionEvent event)  
 {  
  
 button\_begin.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f2\_plan.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
 @FXML  
 public void click\_button\_inform(ActionEvent event)  
 {  
  
 button\_inform.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("information.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
 @FXML  
 public void click\_button\_finish(ActionEvent event)  
 {  
  
 button\_finish.getScene().getWindow().hide();  
  
  
 }  
  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «F2\_PLAN.JAVA»**

package com.example.kurs;  
  
  
import java.io.\*;  
import java.net.URL;  
import java.time.LocalDate;  
import java.time.format.DateTimeFormatter;  
import java.util.ArrayList;  
import java.util.ResourceBundle;  
  
import javafx.event.ActionEvent;  
import javafx.event.Event;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.fxml.FXML;  
import javafx.stage.Stage;  
  
import static kotlin.io.ConsoleKt.*readLine*;  
  
public class f2\_plan {  
  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button button\_back;  
  
 @FXML  
 private Button button\_paramhotels;  
  
 @FXML  
 private ComboBox<String> field\_city;  
  
 @FXML  
 private ComboBox<String> field\_classofplane;  
  
 @FXML  
 private DatePicker field\_data;  
  
 @FXML  
 private TextField field\_days;  
  
 @FXML  
 private TextField field\_persons;  
  
  
 city citydata=new city();  
 constrjourney journey=new constrjourney();  
  
  
 @FXML  
 void initialize() {  
 //город  
 field\_city.getItems().addAll(  
 "Париж",  
 "Рим",  
 "Берлин"  
 );  
 field\_city.setEditable(false);  
 field\_city.setValue("Париж");  
  
 //класс самолёта  
 field\_classofplane.getItems().addAll(  
 "Эконом класс",  
 "Бизнес класс",  
 "Первый класс"  
 );  
 field\_classofplane.setEditable(false);  
 field\_classofplane.setValue("Эконом класс");  
  
  
 //дата  
 field\_data.setEditable(false);  
 LocalDate minDate = LocalDate.*now*();  
  
 field\_data.setDayCellFactory(d ->  
 new DateCell() {  
 @Override  
 public void updateItem(LocalDate item, boolean empty) {  
 super.updateItem(item, empty);  
 setDisable(item.isBefore(minDate));  
 }  
 });  
  
 field\_data.setValue(LocalDate.*now*()); //дата по умолчанию  
  
  
 }  
  
 public boolean check\_field\_days() {  
 int days;  
 boolean check\_days = false;  
 try {  
 days = Integer.*parseInt*(field\_days.getText());  
 System.*out*.println(days);  
 if (days > 2) {  
 check\_days = true;  
 } else {  
 Alert errorAlert = new Alert(Alert.AlertType.*ERROR*);  
 errorAlert.setHeaderText("Ошибка");  
 errorAlert.setContentText("Введено недопустимое количество дней \n" +  
 "Минимальное количество дней: 3");  
 errorAlert.showAndWait();  
 }  
 } catch (Exception e) {  
 System.*out*.println(e);  
 Alert errorAlert = new Alert(Alert.AlertType.*ERROR*);  
 errorAlert.setHeaderText("Ошибка");  
 errorAlert.setContentText("Данные в поле количество дней введены некорректно");  
 errorAlert.showAndWait();  
 }  
 return check\_days;  
 }  
  
 public boolean check\_field\_persons() {  
 int persons;  
 boolean check\_persons = false;  
 try {  
 persons = Integer.*parseInt*(field\_persons.getText());  
 System.*out*.println(persons);  
 if (persons > 0) {  
 check\_persons = true;  
 } else {  
 Alert errorAlert = new Alert(Alert.AlertType.*ERROR*);  
 errorAlert.setHeaderText("Ошибка");  
 errorAlert.setContentText("Введено недопустимое количество человек");  
 errorAlert.showAndWait();  
 }  
 } catch (Exception e) {  
 System.*out*.println(e);  
 Alert errorAlert = new Alert(Alert.AlertType.*ERROR*);  
 errorAlert.setHeaderText("Ошибка");  
 errorAlert.setContentText("Данные в поле количество человек введены некорректно");  
 errorAlert.showAndWait();  
 }  
  
 return check\_persons;  
 }  
  
 public void getdatafromfile() throws IOException {  
 citydata.*delete\_city*();  
 File filehotels, filemuseums, filetheatres, filepriceticketsofplanes;  
  
  
 // try {  
 filehotels = new File("E:\\Курсовой проект\\cities\\Paris\\hotelsofparis.txt");  
 filemuseums = new File("E:\\Курсовой проект\\cities\\Paris\\museumsofparis.txt");  
 filetheatres = new File("E:\\Курсовой проект\\cities\\Paris\\theatresofparis.txt");  
 filepriceticketsofplanes=new File("E:\\Курсовой проект\\cities\\Paris\\planesticketpriceofparis.txt");  
  
 // }  
 // catch (Exception e) {  
 // System.out.println("Файл не найден");  
 // }  
  
  
 if (field\_city.getValue() == "Берлин") {  
 try {  
 filehotels = new File("E:\\Курсовой проект\\cities\\Berlin\\hotelsofberlin.txt");  
 filemuseums = new File("E:\\Курсовой проект\\cities\\Berlin\\museumsofberlin.txt");  
 filetheatres = new File("E:\\Курсовой проект\\cities\\Berlin\\theatresofberlin.txt");  
 filepriceticketsofplanes=new File("E:\\Курсовой проект\\cities\\Berlin\\planesticketpriceofberlin.txt");  
 } catch (Exception e) {  
 System.*out*.println("Файл не найден");  
 }  
 } else if (field\_city.getValue() == "Рим") {  
 try {  
 filehotels = new File("E:\\Курсовой проект\\cities\\Rome\\hotelsofrome.txt");  
 filemuseums = new File("E:\\Курсовой проект\\cities\\Rome\\museumsofrome.txt");  
 filetheatres = new File("E:\\Курсовой проект\\cities\\Rome\\theatresofrome.txt");  
 filepriceticketsofplanes=new File("E:\\Курсовой проект\\cities\\Rome\\planesticketpriceofrome.txt");  
 } catch (Exception e) {  
 System.*out*.println("Файл не найден");  
 }  
 }  
  
  
 //импорт данных отелей  
 BufferedReader inputhotels = new BufferedReader(new InputStreamReader(new FileInputStream(filehotels), "UTF-8"));  
  
  
 char ch;  
 int count = 0;  
 int checkcount = 0;  
 String stroka = "";  
  
 while(inputhotels.ready())  
 {  
 stroka="";  
  
 while ((ch = (char) inputhotels.read()) != ';') {  
 // System.out.println(ch);  
 stroka = stroka + String.*valueOf*(ch);  
  
 }  
  
  
 count++;  
 // System.out.println("count: "+ count);  
 checkcount++;  
 // System.out.println("checkcount: "+ checkcount);  
 System.*out*.println(stroka);  
  
 if(count==1)  
 {  
 citydata.*set\_nameofhotel*(stroka);  
 }  
 else if (count==2)  
 {  
 citydata.*set\_stars*(Integer.*parseInt*(stroka));  
 }  
 else if (count==3)  
 {  
 boolean checkpool=false;  
 if (stroka.equals("Есть")) {checkpool=true;}  
 citydata.*set\_pool*(checkpool);  
 }  
 else if (count==4)  
 {  
 boolean checkgym=false;  
 if (stroka.equals("Есть")) {checkgym=true;}  
 citydata.*set\_gym*(checkgym);  
 }  
 else if (count==5)  
 {  
 citydata.*set\_costperday*(Integer.*parseInt*(stroka));  
  
 }  
 else if (count ==6)  
 {  
 citydata.*set\_typeoffood*(stroka);  
 }  
 else if (count==7)  
 {  
 citydata.*set\_pricefortypeoffood*(Integer.*parseInt*(stroka));  
  
 }  
 else if (count ==8)  
 {  
 citydata.*set\_typeoffood*(stroka);  
 }  
 else if (count==9)  
 {  
 citydata.*set\_pricefortypeoffood*(Integer.*parseInt*(stroka));  
  
 }  
 else if (count ==10)  
 {  
 citydata.*set\_typeoffood*(stroka);  
 }  
 else if (count==11)  
 {  
 citydata.*set\_pricefortypeoffood*(Integer.*parseInt*(stroka));  
  
 }  
 else if (count==12)  
 {  
 citydata.*set\_pricefortransfertohotel*(Integer.*parseInt*(stroka));  
 count=0;  
 }  
  
  
 }  
 inputhotels.close();  
  
 //импорт данных типа класса в самолёте  
 count = 0;  
 BufferedReader inputticketpriceofplane = new BufferedReader(new InputStreamReader(new FileInputStream(filepriceticketsofplanes), "UTF-8"));  
 while (inputticketpriceofplane.ready())  
 {  
 stroka = "";  
  
 while ((ch = (char) inputticketpriceofplane.read()) != ';') {  
 // System.out.println(ch);  
 stroka = stroka + String.*valueOf*(ch);  
  
 }  
 count++;  
  
  
 if (count==1)  
 {  
 citydata.*set\_nameofclassplane*(stroka);  
 System.*out*.println("Название класса самолёта");  
 System.*out*.println(stroka);  
 }  
 else if (count==2)  
 {  
  
 count=0;  
 citydata.*set\_ticketpriceofplane*(Integer.*parseInt*(stroka));  
 System.*out*.println("Стоимость билета на самолёт");  
 System.*out*.println(stroka);  
 }  
  
 }  
 inputticketpriceofplane.close();  
  
  
 //импорт данных театров  
 count = 0;  
 BufferedReader inputtheatres = new BufferedReader(new InputStreamReader(new FileInputStream(filetheatres), "UTF-8"));  
  
 while (inputtheatres.ready())  
 {  
 stroka = "";  
  
 while ((ch = (char) inputtheatres.read()) != ';') {  
 // System.out.println(ch);  
 stroka = stroka + String.*valueOf*(ch);  
  
 }  
 count++;  
  
  
 if (count==1)  
 {  
 citydata.*set\_nameoftheatre*(stroka);  
 System.*out*.println("Название театра");  
 System.*out*.println(stroka);  
 }  
 else if (count==2)  
 {  
  
 count=0;  
 citydata.*set\_ticketpriceoftheatre*(Integer.*parseInt*(stroka));  
 System.*out*.println("Стоимость билета");  
 System.*out*.println(stroka);  
 }  
  
 }  
 inputtheatres.close();  
 citydata.*get\_nameoftheatre*();  
 citydata.*get\_ticketpriceoftheatre*();  
  
 //импорт данных музеев  
 count=0;  
 BufferedReader inputmuseums = new BufferedReader(new InputStreamReader(new FileInputStream(filemuseums), "UTF-8"));  
 while(inputmuseums.ready()) {  
 stroka = "";  
  
 while ((ch = (char) inputmuseums.read()) != ';') {  
 // System.out.println(ch);  
 stroka = stroka + String.*valueOf*(ch);  
  
 }  
 count++;  
 System.*out*.println(stroka);  
 if (count==1)  
 {  
 citydata.*set\_nameofmuseum*(stroka);  
 }  
 else if (count==2)  
 {  
 citydata.*set\_ticketpriceofmuseum*(Integer.*parseInt*(stroka));  
 count=0;  
 }  
  
 }  
  
 inputmuseums.close();  
  
  
  
 }  
  
  
 public void setdataforjourney() //данные для поездки  
 {  
 LocalDate mydatebegin = field\_data.getValue();  
 // String my\_format\_datebegin = mydatebegin.format(DateTimeFormatter.ofPattern("dd.MM.YYYY"));  
 //System.out.println(my\_format\_datebegin);  
 LocalDate mydateend = mydatebegin.plusDays(Integer.*parseInt*(field\_days.getText()));  
 // String my\_format\_dateend = mydatebegin.format(DateTimeFormatter.ofPattern("dd.MM.YYYY"));  
  
 //определение необходимого количества номеров в отеле  
 int kolrooms;  
 if (Integer.*parseInt*(field\_persons.getText()) % 3==0)  
 {  
 kolrooms=Integer.*parseInt*(field\_persons.getText()) / 3;  
 }  
 else  
 {  
 kolrooms=Integer.*parseInt*(field\_persons.getText()) / 3+1;  
 }  
  
 journey = new constrjourney(field\_city.getValue(), mydatebegin, mydateend,  
 Integer.*parseInt*(field\_days.getText()),  
 Integer.*parseInt*(field\_persons.getText()), kolrooms, field\_classofplane.getValue(),  
 citydata.*get\_priceclassofplane*(field\_classofplane.getValue()));  
 }  
  
 @FXML  
 public void click\_button\_paramhotels(ActionEvent event) throws IOException {  
 //==============  
  
 LocalDate mydate = field\_data.getValue();  
 String my\_format\_date = mydate.format(DateTimeFormatter.*ofPattern*("dd.MM.YYYY"));  
 System.*out*.println(my\_format\_date);  
  
  
  
 boolean check\_days = check\_field\_days();  
 boolean check\_persons=check\_field\_persons();  
 if ((check\_days==true)&&(check\_persons==true)) {  
 //======  
 getdatafromfile();  
 citydata.*show\_city\_data*();  
 setdataforjourney();  
 journey.*show\_journey\_data*();  
  
 //=================  
  
  
  
 button\_paramhotels.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f3\_charactofhotels.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
 }  
  
 @FXML  
 public void click\_button\_back(ActionEvent event)  
 {  
  
  
 button\_back.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f1\_mainmenu.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
  
  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «F3\_CHARACTOFHOTELS.JAVA»**

package com.example.kurs;  
  
import java.io.IOException;  
import java.net.URL;  
import java.util.ResourceBundle;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.ComboBox;  
import javafx.stage.Stage;  
  
public class f3\_charactofhotels {  
  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button button\_back;  
  
 @FXML  
 private Button button\_showvariants;  
  
 @FXML  
 private ComboBox<String> field\_food;  
  
 @FXML  
 private ComboBox<String> field\_gym;  
  
 @FXML  
 private ComboBox<String> field\_pool;  
  
 @FXML  
 private ComboBox<String> field\_stars;  
  
  
 city citydata=new city();  
 constrjourney journey=new constrjourney();  
  
  
 @FXML  
 void initialize() {  
 //количество звёзд у отеля  
 field\_stars.getItems().addAll(  
 "4",  
 "5"  
 );  
 field\_stars.setEditable(false);  
 field\_stars.setValue("4");  
  
 //наличие бассейна  
 field\_pool.getItems().addAll(  
 "Не важно",  
 "Есть"  
 );  
 field\_pool.setEditable(false);  
 field\_pool.setValue("Не важно");  
  
 //наличие спортзала  
 field\_gym.getItems().addAll(  
 "Не важно",  
 "Есть"  
 );  
 field\_gym.setEditable(false);  
 field\_gym.setValue("Не важно");  
  
 //тип питания  
 field\_food.getItems().addAll(  
 "Завтрак",  
 "Завтрак, ужин",  
 "Всё включено"  
 );  
 field\_food.setEditable(false);  
 field\_food.setValue("Завтрак");  
  
  
 }  
  
  
 public void choosehotel()  
 {  
 boolean check\_pool=false;  
 boolean check\_gym=false;  
  
 if (field\_pool.getValue().equals("Есть")) {check\_pool=true;}  
 if (field\_gym.getValue().equals("Есть")) {check\_gym=true;}  
  
 int indexofhotelbycriteria=0;  
 int indexfood=0;  
 int minsumperday=citydata.*get\_costperday*().get(0);  
  
 for(int i =0; i<citydata.*get\_stars*().size(); i++)  
 {  
 if (citydata.*get\_stars*().get(i)==Integer.*parseInt*(field\_stars.getValue())) {  
 System.*out*.println("сравнение звезд");  
 if (citydata.*get\_pool*().get(i) == check\_pool) {  
 System.*out*.println("сравнение бассейна");  
 System.*out*.println(check\_pool);  
 if (citydata.*get\_gym*().get(i) == check\_gym) {  
 System.*out*.println("сравнение зала");  
 System.*out*.println(check\_gym);  
 if (citydata.*get\_costperday*().get(i)<=minsumperday)  
 {  
 System.*out*.println("сравнение цены");  
 indexofhotelbycriteria=i;  
 int razm;  
 razm=citydata.*get\_typeoffood*().size()/citydata.*get\_stars*().size();  
 for(int j=indexofhotelbycriteria\*razm;j<indexofhotelbycriteria\*razm+razm;j++)  
 {  
 if (citydata.*get\_typeoffood*().get(j).equals(field\_food.getValue()))  
 {  
 indexfood=j;  
 System.*out*.println(citydata.*get\_typeoffood*().get(j));  
 System.*out*.println(field\_food.getValue());  
  
 }  
  
 }  
 }  
 }  
 }  
 }  
 }  
  
  
 journey.*set\_hotel\_data*(citydata.get\_nameofhotel().get(indexofhotelbycriteria),  
 citydata.*get\_stars*().get(indexofhotelbycriteria),  
 citydata.*get\_pool*().get(indexofhotelbycriteria),  
 citydata.*get\_gym*().get(indexofhotelbycriteria),  
 citydata.*get\_typeoffood*().get(indexfood), citydata.*get\_pricefortypeoffood*().get(indexfood),  
 citydata.*get\_costperday*().get(indexofhotelbycriteria),  
 citydata.*get\_pricefortransfertohotel*().get(indexofhotelbycriteria));  
  
  
  
 }  
 @FXML  
 public void click\_button\_back(ActionEvent event)  
 {  
  
 button\_back.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f2\_plan.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
 @FXML  
 public void click\_button\_showvariants(ActionEvent event)  
 {  
 choosehotel();  
 System.*out*.println("запуск функции journey data из f3");  
 journey.*show\_journey\_data*();  
  
 button\_showvariants.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f4\_showhotel.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «F4\_SHOWHOTEL.JAVA»**

package com.example.kurs;  
  
import java.io.File;  
import java.io.IOException;  
import java.net.URL;  
import java.util.ResourceBundle;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
  
public class f4\_showhotel {  
  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button button\_back;  
  
 @FXML  
 private Button button\_museums\_and\_theatres;  
  
 @FXML  
 private Text field\_gym;  
  
 @FXML  
 private Text field\_typeoffood;  
  
 @FXML  
 private Text field\_hotelname;  
  
 @FXML  
 private ImageView field\_picture;  
  
 @FXML  
 private Text field\_pool;  
  
 @FXML  
 private Text field\_costperday;  
  
 constrjourney journey = new constrjourney();  
  
  
 @FXML  
 void initialize() {  
 field\_hotelname.setText(journey.*get\_nameofhotel*());  
 String pool="Нет";  
 String gym="Нет";  
 if (journey.*get\_pool*()==true) {pool="Есть";}  
 if (journey.*get\_gym*()==true) {gym="Есть";}  
 field\_pool.setText(pool);  
 field\_gym.setText(gym);  
 field\_typeoffood.setText(journey.*get\_typeoffood*());  
 field\_costperday.setText(Integer.*toString*(journey.*get\_costperday*())+" руб.");  
  
 String pathtophotoofhotel="E:\\Курсовой проект\\cities\\";  
 if (journey.*get\_city*().equals("Париж"))  
 {  
 pathtophotoofhotel=pathtophotoofhotel+"Paris\\picturesofhotelsofparis\\"+journey.*get\_nameofhotel*()+".jpg";  
 }  
 else if (journey.*get\_city*().equals("Рим"))  
 {  
 pathtophotoofhotel=pathtophotoofhotel+"Rome\\picturesofhotelsofrome\\"+journey.*get\_nameofhotel*()+".jpg";  
 }  
 else  
 {  
 pathtophotoofhotel=pathtophotoofhotel+"Berlin\\picturesofhotelsofberlin\\"+journey.*get\_nameofhotel*()+".jpg";  
 }  
 File hotelpicture= new File(pathtophotoofhotel);  
 Image image = new Image(hotelpicture.toURI().toString());  
 field\_picture.setImage(image);  
  
  
 }  
  
 @FXML  
 public void click\_button\_back(ActionEvent event)  
 {  
  
 button\_back.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f3\_charactofhotels.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
 @FXML  
 public void click\_button\_museums\_and\_theatres(ActionEvent event)  
 {  
  
 button\_museums\_and\_theatres.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f5\_museumsandtheatres.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «F5\_MUSEUMSANDTHEATRES.JAVA»**

package com.example.kurs;  
  
import java.io.IOException;  
import java.net.URL;  
import java.time.LocalDate;  
import java.util.ResourceBundle;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.CheckBox;  
import javafx.stage.Stage;  
  
public class f5\_museumsandtheatres {  
  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button button\_back;  
  
 @FXML  
 private Button button\_finalsum;  
  
 @FXML  
 private CheckBox checkbox\_1museum;  
  
 @FXML  
 private CheckBox checkbox\_1theatre;  
  
 @FXML  
 private CheckBox checkbox\_2museum;  
  
 @FXML  
 private CheckBox checkbox\_2theatre;  
  
 @FXML  
 private CheckBox checkbox\_3museum;  
  
 @FXML  
 private CheckBox checkbox\_3theatre;  
  
  
 city citydata = new city();  
 constrjourney journey = new constrjourney();  
  
 @FXML  
 void initialize() {  
 checkbox\_1museum.setText(citydata.*get\_nameofmuseum*().get(0));  
 checkbox\_2museum.setText(citydata.*get\_nameofmuseum*().get(1));  
 checkbox\_3museum.setText(citydata.*get\_nameofmuseum*().get(2));  
  
 checkbox\_1theatre.setText(citydata.*get\_nameoftheatre*().get(0));  
 checkbox\_2theatre.setText(citydata.*get\_nameoftheatre*().get(1));  
 checkbox\_3theatre.setText(citydata.*get\_nameoftheatre*().get(2));  
 }  
  
  
 @FXML  
 public void click\_button\_back(ActionEvent event)  
 {  
  
 button\_back.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f4\_showhotel.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
 @FXML  
 public void click\_button\_button\_finalsum(ActionEvent event)  
 {  
 set\_theatres\_and\_museums();  
 journey.*show\_journey\_data*();  
 button\_finalsum.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f6\_finalsum.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
 public void set\_theatres\_and\_museums()  
 {  
  
 if (checkbox\_1theatre.isSelected())  
 {  
 journey.*set\_theatre*(citydata.*get\_nameoftheatre*().get(0), citydata.*get\_ticketpriceoftheatre*().get(0),  
 journey.*get\_datebegintrip*()  
  
 );  
  
 }  
 if (checkbox\_2theatre.isSelected())  
 {  
 journey.*set\_theatre*(citydata.*get\_nameoftheatre*().get(1), citydata.*get\_ticketpriceoftheatre*().get(1),  
 journey.*get\_datebegintrip*().plusDays(1)  
 );  
 }  
 if (checkbox\_3theatre.isSelected())  
 {  
 journey.*set\_theatre*(citydata.*get\_nameoftheatre*().get(2), citydata.*get\_ticketpriceoftheatre*().get(2),  
 journey.*get\_datebegintrip*().plusDays(2)  
 );  
 }  
  
 if (checkbox\_1museum.isSelected())  
 {  
 journey.*set\_museum*(citydata.*get\_nameofmuseum*().get(0), citydata.*get\_ticketpriceofmuseum*().get(0));  
 }  
 if (checkbox\_2museum.isSelected())  
 {  
 journey.*set\_museum*(citydata.*get\_nameofmuseum*().get(1), citydata.*get\_ticketpriceofmuseum*().get(1));  
 }  
 if (checkbox\_3museum.isSelected())  
 {  
 journey.*set\_museum*(citydata.*get\_nameofmuseum*().get(2), citydata.*get\_ticketpriceofmuseum*().get(2));  
 }  
  
 }  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «F6\_FINALSUM.JAVA»**

package com.example.kurs;  
  
import java.io.IOException;  
import java.net.URL;  
import java.time.LocalDate;  
import java.time.format.DateTimeFormatter;  
import java.util.ArrayList;  
import java.util.ResourceBundle;  
  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.TableColumn;  
import javafx.scene.control.TableView;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
  
public class f6\_finalsum {  
  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button button\_finish;  
  
 @FXML  
 private Button button\_mainmenu;  
  
 @FXML  
 private TableView<classforoutput> field\_table;  
 @FXML  
 private TableColumn<classforoutput, String> field\_object;  
  
 @FXML  
 private TableColumn<classforoutput, String> field\_data;  
  
  
  
 @FXML  
 private TableColumn<classforoutput, Integer> field\_sum;  
  
  
  
 constrjourney journey = new constrjourney();  
 ObservableList<classforoutput> journey\_data = FXCollections.*observableArrayList*();  
 @FXML  
 void initialize() {  
 final\_sum();  
 init\_data();  
  
  
  
 field\_object.setCellValueFactory(new PropertyValueFactory<classforoutput, String>("ob"));  
 field\_data.setCellValueFactory(new PropertyValueFactory<classforoutput, String>("data"));  
 field\_sum.setCellValueFactory(new PropertyValueFactory<classforoutput, Integer>("sum"));  
  
 field\_table.setItems(journey\_data);  
  
  
  
  
 }  
 void init\_data()  
 {  
  
 String my\_format\_datebegin = journey.*get\_datebegintrip*().format(DateTimeFormatter.*ofPattern*("dd.MM.YYYY"));  
 String my\_format\_dateend = journey.*get\_dateendtrip*().format(DateTimeFormatter.*ofPattern*("dd.MM.YYYY"));  
  
 journey\_data.add(new classforoutput("Самолёт (СПб-"+journey.*get\_city*()+")",  
 my\_format\_datebegin, journey.*get\_sumofticketsplane*()));  
 journey\_data.add(new classforoutput("Самолёт ("+journey.*get\_city*()+"-СПб)",  
 my\_format\_dateend, journey.*get\_sumofticketsplane*()));  
 journey\_data.add(new classforoutput("Трансфер (из аэропорта в отель)",  
 my\_format\_datebegin, journey.*get\_sumfortranser*()));  
 journey\_data.add(new classforoutput("Трансфер (из отеля в аэропорт)",  
 my\_format\_dateend, journey.*get\_sumfortranser*()));  
 journey\_data.add(new classforoutput("Отель '"+journey.*get\_nameofhotel*()+"'"+  
 "(количество номеров: "+journey.*get\_kolrooms*()+")", "", journey.*get\_sumforhotel*()));  
 journey\_data.add(new classforoutput("Стоимость еды в отеле", "", journey.*get\_sumforfood*()));  
  
 if (journey.*get\_nameofmuseum*().isEmpty()==false)  
 {  
  
 for(int i =0; i<journey.*get\_nameofmuseum*().size();i++)  
 {  
 journey\_data.add(new classforoutput("Музей '"+journey.*get\_nameofmuseum*().get(i)+"'",  
 "", journey.*get\_sumeachmuseum*().get(i)));  
 }  
  
 }  
  
 if (journey.*get\_nameoftheatre*().isEmpty()==false)  
 {  
  
 for(int i =0; i<journey.*get\_nameoftheatre*().size();i++)  
 {  
 String my\_format\_datevisittheatre = journey.*get\_dateofvisittheatre*().get(i).format(DateTimeFormatter.*ofPattern*("dd.MM.YYYY"));  
 journey\_data.add(new classforoutput("Музей '"+journey.*get\_nameoftheatre*().get(i)+"'",  
 my\_format\_datevisittheatre, journey.*get\_sumeachtheatre*().get(i)));  
 }  
  
 }  
 journey\_data.add(new classforoutput("Итоговая сумма", "", journey.*get\_sum\_full*()));  
  
  
 }  
  
 public void final\_sum()  
 {  
 int csumofticketsplane, csumfortranser, csumforhotel, csumforfood, csum\_full;  
 int sumfortheatres=0;  
 int sumformuseums=0;  
 ArrayList <Integer> csumfortheatres= new ArrayList<Integer>();  
 ArrayList <Integer> csumformuseums=new ArrayList<Integer>();  
 csumofticketsplane=journey.*get\_kolperson*()\*journey.*get\_ticketpriceofplane*();  
 csumfortranser=journey.*get\_kolperson*()\*journey.*get\_pricefortransfertohotel*();  
 csumforhotel=journey.*get\_kolrooms*()\*journey.*get\_costperday*()\*journey.*get\_koldays*();  
  
 for(int i=0; i<journey.*get\_ticketpriceoftheatre*().size();i++)  
 {  
 csumfortheatres.add(journey.*get\_ticketpriceoftheatre*().get(i)\*journey.*get\_kolperson*());  
 sumfortheatres=sumfortheatres+journey.*get\_ticketpriceoftheatre*().get(i)\*journey.*get\_kolperson*();  
 }  
  
 for(int i =0; i<journey.*get\_ticketpriceofmuseum*().size();i++)  
 {  
 csumformuseums.add(journey.*get\_ticketpriceofmuseum*().get(i)\*journey.*get\_kolperson*());  
 sumformuseums=sumformuseums+journey.*get\_ticketpriceofmuseum*().get(i)\*journey.*get\_kolperson*();  
 }  
 csumforfood=journey.*get\_pricefortypeoffood*()\*journey.*get\_kolperson*()\*journey.*get\_koldays*();  
 csum\_full=csumofticketsplane\*2+csumfortranser\*2+csumforhotel+csumforfood+sumfortheatres+sumformuseums;  
  
 journey.*set\_sum*(csumofticketsplane, csumfortranser, csumforhotel,  
 csumforfood, sumformuseums, sumfortheatres, csum\_full, csumfortheatres, csumformuseums);  
 }  
  
 @FXML  
 public void click\_button\_mainmenu(ActionEvent event)  
 {  
  
 button\_mainmenu.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f1\_mainmenu.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
  
 @FXML  
 public void click\_button\_finish(ActionEvent event)  
 {  
  
 button\_finish.getScene().getWindow().hide();  
  
  
 }  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «INFORMATION.JAVA»**

package com.example.kurs;  
  
import java.io.IOException;  
import java.net.URL;  
import java.util.ResourceBundle;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.stage.Stage;  
  
public class information {  
  
 @FXML  
 private ResourceBundle resources;  
  
 @FXML  
 private URL location;  
  
 @FXML  
 private Button button\_mainmenu;  
  
  
  
 @FXML  
 void initialize() {  
  
 }  
  
 @FXML  
 public void click\_button\_mainmenu(ActionEvent event)  
 {  
  
 button\_mainmenu.getScene().getWindow().hide();  
  
 FXMLLoader loader = new FXMLLoader();  
  
 loader.setLocation(getClass().getResource("f1\_mainmenu.fxml"));  
 try {  
 loader.load();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 Parent root = loader.getRoot();  
 Stage stage = new Stage();  
 stage.setTitle("Конструктор путешествий");  
 stage.setScene(new Scene(root));  
 stage.show();  
  
 }  
  
  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «CITY.JAVA»**

package com.example.kurs;  
  
import java.util.ArrayList;  
  
public class city  
{  
 //отели  
 //private String nameofhotel; //название отеля  
 static private ArrayList <String> *nameofhotel* = new ArrayList<String>(); //название отеля  
  
 // private int stars; //количество звёзд у отеля  
 static private ArrayList <Integer> *stars* = new ArrayList<Integer>(); //количество звёзд у отеля  
  
 //private boolean pool; //наличие бассейна  
 static private ArrayList <Boolean> *pool* = new ArrayList<Boolean>(); //наличие бассейна  
  
 // private boolean gym; //наличие спортзала  
 static private ArrayList <Boolean> *gym* = new ArrayList<Boolean>(); //наличие спортзала  
  
 static private ArrayList<String> *typeoffood* = new ArrayList<String>(); //тип еды  
 static private ArrayList <Integer> *pricefortypeoffood* = new ArrayList<Integer>(); //стоимость для типа еды  
 //private int costperday; //стоимость за день  
 static private ArrayList <Integer> *costperday* = new ArrayList<Integer>(); //стоимость за день  
  
 //театры  
 static private ArrayList <String> *nameoftheatre*= new ArrayList<String>(); //название театра  
 static private ArrayList <Integer> *ticketpriceoftheatre* = new ArrayList<Integer>(); //стоимость билета в театр  
 //музеи  
 static private ArrayList <String> *nameofmuseum*= new ArrayList<String>(); //название музея  
 static private ArrayList <Integer> *ticketpriceofmuseum* = new ArrayList<Integer>(); //стоимость билета в музей  
  
 //самолёт  
 static private ArrayList <String> *nameofclassplane*= new ArrayList<String>(); //название музея  
 static private ArrayList <Integer> *ticketpriceofplane*= new ArrayList<Integer>(); //стоимость билета в музей  
  
 static private ArrayList <Integer> *pricefortransfertohotel* = new ArrayList<Integer>();  
  
 public city()  
 {  
 /\*  
 this.nameofhotel=nameofhotel;  
 this.stars=stars;  
 this.pool=pool;  
 this.gym=gym;  
 this.typeoffood=typeoffood;  
 this.costperday=costperday;  
\*/  
  
 }  
  
 static public void set\_nameofhotel(String namehotel)  
 {  
 *nameofhotel*.add(namehotel);  
 }  
  
 static public void set\_stars(int st)  
 {  
 *stars*.add(st);  
 }  
  
 static public void set\_pool(boolean p)  
 {  
 *pool*.add(p);  
 }  
  
 static public void set\_gym(boolean g)  
 {  
 *gym*.add(g);  
 }  
  
 static public void set\_costperday(int cost)  
 {  
 *costperday*.add(cost);  
 }  
  
 static public void set\_nameoftheatre(String name)  
 {  
 *nameoftheatre*.add(name);  
 }  
  
 static public void set\_ticketpriceoftheatre (int price)  
 {  
 *ticketpriceoftheatre*.add(price);  
 }  
  
 static public void set\_nameofmuseum(String name)  
 {  
 *nameofmuseum*.add(name);  
 }  
  
 static public void set\_ticketpriceofmuseum (int price)  
 {  
 *ticketpriceofmuseum*.add(price);  
 }  
  
 static public void set\_nameofclassplane(String name)  
 {  
 *nameofclassplane*.add(name);  
 }  
  
 static public void set\_ticketpriceofplane(int price)  
 {  
 *ticketpriceofplane*.add(price);  
 }  
  
 static public void set\_typeoffood (String type) {*typeoffood*.add(type);}  
 static public void set\_pricefortypeoffood (int price) {*pricefortypeoffood*.add(price);}  
 static public void set\_pricefortransfertohotel (int price) {*pricefortransfertohotel*.add(price);}  
  
 public ArrayList<String> get\_nameofhotel()  
 {  
 return *nameofhotel*;  
 }  
  
 static public ArrayList<Integer> get\_stars()  
 {  
 return *stars*;  
 }  
  
 static public ArrayList<Boolean> get\_pool()  
 {  
 return *pool*;  
 }  
  
 static public ArrayList<Boolean> get\_gym()  
 {  
 return *gym*;  
 }  
  
 static public ArrayList<Integer> get\_costperday()  
 {  
 return *costperday*;  
 }  
  
 static public ArrayList<String> get\_nameoftheatre()  
 {  
 return *nameoftheatre*;  
 }  
 static public ArrayList<Integer> get\_ticketpriceoftheatre()  
 {  
 return *ticketpriceoftheatre*;  
 }  
  
 static public ArrayList<String> get\_nameofmuseum()  
 {  
 return *nameofmuseum*;  
 }  
  
 static public ArrayList<Integer> get\_ticketpriceofmuseum()  
 {  
 return *ticketpriceofmuseum*;  
 }  
 static public ArrayList<Integer> get\_pricefortransfertohotel() {return *pricefortransfertohotel*;}  
 static public ArrayList<String> get\_typeoffood()  
 {  
 return *typeoffood*;  
 }  
  
 static public ArrayList<Integer> get\_pricefortypeoffood()  
 {  
 return *pricefortypeoffood*;  
 }  
  
 static public int get\_priceclassofplane(String classofplane)  
 {  
 int fixnumberofclassplane = 0;  
 int count =0;  
 for(String d: *nameofclassplane*)  
 {  
 if (classofplane.equals(d))  
 {  
 break;  
 }  
 fixnumberofclassplane++;  
 }  
  
 return *ticketpriceofplane*.get(fixnumberofclassplane);  
 }  
  
  
 static public void show\_city\_data()  
 {  
 System.*out*.print("Данные города");  
 System.*out*.println("Название отеля");  
 *nameofhotel*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Количество звёзд");  
 *stars*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Наличие бассейна");  
 *pool*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Наличие спортзала");  
 *gym*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Стоимость за 1 день");  
 *costperday*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Названия музея");  
 *nameofmuseum*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Стоимость билетов в музей");  
 *ticketpriceofmuseum*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
  
  
  
 System.*out*.println("Названия театра");  
 *nameoftheatre*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Стоимость билетов в театр");  
 *ticketpriceoftheatre*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Названия класса самолёта");  
 *nameofclassplane*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Стоимость билета на самолёт");  
 *ticketpriceofplane*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Тип еды");  
 *typeoffood*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
 System.*out*.println("Цена за еду");  
 *pricefortypeoffood*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
 }  
  
 static public void delete\_city() //очистка данных в случае изменения города  
 {  
 *nameofhotel*.clear();  
 *stars*.clear();  
 *pool*.clear();  
 *gym*.clear();  
 *costperday*.clear();  
 *nameoftheatre*.clear();  
 *ticketpriceoftheatre*.clear();  
 *nameofmuseum*.clear();  
 *ticketpriceofmuseum*.clear();  
 *nameofclassplane*.clear();  
 *ticketpriceofplane*.clear();  
 *nameofclassplane*.clear();  
 *ticketpriceofplane*.clear();  
 *pricefortransfertohotel*.clear();  
  
 }  
  
/\*  
 public int get\_costperday()  
 {  
 //return costperday;  
 }  
\*/  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «CONSTRJOURNEY.JAVA»**

package com.example.kurs;  
  
import java.time.LocalDate;  
import java.util.ArrayList;  
  
public class constrjourney  
{  
 //основные данные  
 static private String *city*; //город назначения  
 static private LocalDate *datebegintrip*; //дата начала поездки  
 static private LocalDate *dateendtrip*; //дата завершения поездки  
 static private int *koldays*; //количество дней поездки  
 static private int *kolperson*; //количество человек, участвующих в поездке  
 static private int *kolrooms*; //необходимое количество номеров  
  
 //данные отеля  
 static private String *nameofhotel*; //название отеля  
 static private int *stars*; //количество звёзд у отеля  
 static private boolean *pool*; //наличие бассейна  
 static private boolean *gym*; //наличие спортзала  
 static private String *typeoffood*; //тип еды  
 static private int *pricefortypeoffood*;//стоимость еды  
 static private int *costperday*; //стоимость 1 номера за день  
 static private int *pricefortransfertohotel*; //стоимость трансфера от аэропорта до отеля  
  
 //данные самолёта  
 static private String *typeofclassplane*; //тип класса  
 static private int *ticketpriceofplane*; //стоимость билета в 1 сторону за 1 человека  
  
 //театры  
 static private ArrayList<String> *nameoftheatre*= new ArrayList<String>(); //название театра  
 static private ArrayList <Integer> *ticketpriceoftheatre* = new ArrayList<Integer>(); //стоимость билета в театр  
 static private ArrayList <LocalDate> *dateofvisittheatre* = new ArrayList<LocalDate>();  
 //музеи  
 static private ArrayList <String> *nameofmuseum*= new ArrayList<String>(); //название музея  
 static private ArrayList <Integer> *ticketpriceofmuseum* = new ArrayList<Integer>(); //стоимость билета в музей  
  
 //итоги  
 static private int *sumofticketsplane*;  
 static private int *sumfortranser*;  
 static private int *sumforhotel*;  
 static private int *sumforfood*;  
 static private int *sumformuseums*;  
 static private int *sumfortheatres*;  
 static private int *sum\_full*;  
 static private ArrayList <Integer> *sumeachtheatre*=new ArrayList<Integer>();  
 static private ArrayList <Integer> *sumeachmuseum*=new ArrayList<Integer>();  
 public constrjourney(String city, LocalDate dbegin, LocalDate dend, int koldays, int kolperson, int kolrooms, String typeofclassplane, int ticketpriceofplane)  
 {  
 this.*city*=city;  
 this.*datebegintrip*=dbegin;  
 this.*dateendtrip*=dend;  
 this.*koldays*=koldays;  
 this.*kolperson*=kolperson;  
 this.*kolrooms*=kolrooms;  
 this.*typeofclassplane*=typeofclassplane;  
 this.*ticketpriceofplane*=ticketpriceofplane;  
 }  
public constrjourney()  
{  
  
}  
 static public void set\_hotel\_data(String cnameofhotel, int cstars, boolean cpool, boolean cgym,  
 String ctypeoffood, int cpricefortypeoffood, int ccostperday, int cpricefortransfertohotel)  
{  
 *nameofhotel*=cnameofhotel;  
 *stars*=cstars;  
 *pool*=cpool;  
 *gym*=cgym;  
 *typeoffood*=ctypeoffood;  
 *pricefortypeoffood*=cpricefortypeoffood;  
 *costperday*=ccostperday;  
 *pricefortransfertohotel*=cpricefortransfertohotel;  
}  
  
 static public void set\_theatre(String name, int cost, LocalDate dat)  
 {  
 *nameoftheatre*.add(name);  
 *ticketpriceoftheatre*.add(cost);  
 *dateofvisittheatre*.add(dat);  
 }  
  
 static public void set\_museum(String name, int cost)  
 {  
 *nameofmuseum*.add(name);  
 *ticketpriceofmuseum*.add(cost);  
 }  
  
 static public void set\_sum(int csumofticketsplane, int csumfortranser, int csumforhotel,  
 int csumforfood, int csumformuseums, int csumfortheatres, int csum\_full,  
 ArrayList <Integer> sumeverytheatre, ArrayList <Integer> sumeverymuseum)  
 {  
 *sumofticketsplane*=csumofticketsplane;  
 *sumfortranser*=csumfortranser;  
 *sumforhotel*=csumforhotel;  
 *sumforfood*=csumforfood;  
 *sumformuseums*=csumformuseums;  
 *sumfortheatres*=csumfortheatres;  
 *sum\_full*=csum\_full;  
 *sumeachtheatre*=sumeverytheatre;  
 *sumeachmuseum*=sumeverymuseum;  
  
 }  
 static public void show\_journey\_data()  
 {  
 System.*out*.print("Данные поездки");  
  
 System.*out*.println("Название города");  
 System.*out*.println(*city*);  
  
  
  
 System.*out*.println("количество дней");  
 System.*out*.println(*koldays*);  
  
 System.*out*.println("количество человек");  
 System.*out*.println(*kolperson*);  
  
 System.*out*.println("количество комнат");  
 System.*out*.println(*kolrooms*);  
  
 System.*out*.println("тип класса самолёта");  
 System.*out*.println(*typeofclassplane*);  
  
 System.*out*.println("Стоимость билета");  
 System.*out*.println(*ticketpriceofplane*);  
  
 System.*out*.println("Название отеля");  
 System.*out*.println(*nameofhotel*);  
  
 System.*out*.println("Количество звёзд");  
 System.*out*.println(*stars*);  
  
 System.*out*.println("Наличие бассейна");  
 System.*out*.println(*pool*);  
  
 System.*out*.println("Наличие спортзала");  
 System.*out*.println(*gym*);  
  
 System.*out*.println("Тип еды");  
 System.*out*.println(*typeoffood*);  
  
 System.*out*.println("Стоимость еды");  
 System.*out*.println(*pricefortypeoffood*);  
  
 System.*out*.println("Стоимость за день");  
 System.*out*.println(*costperday*);  
  
 System.*out*.println("Название театра");  
 *nameoftheatre*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Стоимость билетов в театр");  
 *ticketpriceoftheatre*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Название музея");  
 *nameofmuseum*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Стоимость билетов в музей");  
 *ticketpriceofmuseum*.stream().forEach(d -> {  
 System.*out*.print("'"+d+"'");  
  
 });  
  
 System.*out*.println("Стоимость трансфера до отеля");  
 System.*out*.println(*pricefortransfertohotel*);  
  
 }  
  
  
 static public String get\_city()  
 {  
 return *city*;  
 }  
 static public String get\_nameofhotel()  
 {  
 return *nameofhotel*;  
 }  
  
 static public int get\_stars()  
 {  
 return *stars*;  
 }  
  
 static public boolean get\_pool()  
 {  
 return *pool*;  
 }  
  
 static public boolean get\_gym()  
 {  
 return *gym*;  
 }  
  
 static public String get\_typeoffood()  
 {  
 return *typeoffood*;  
 }  
  
 static public int get\_costperday()  
 {  
 return *costperday*;  
 }  
  
 static public int get\_koldays()  
 {  
 return *koldays*;  
 }  
  
 static public int get\_kolperson()  
 {  
 return *kolperson*;  
 }  
 static public int get\_ticketpriceofplane() {return *ticketpriceofplane*;}  
  
 static public int get\_pricefortransfertohotel() {return *pricefortransfertohotel*;}  
 static public int get\_kolrooms () {return *kolrooms*;}  
 static public LocalDate get\_datebegintrip() {return *datebegintrip*;}  
 static public LocalDate get\_dateendtrip() {return *dateendtrip*;}  
 static public ArrayList<String> get\_nameoftheatre() {return *nameoftheatre*;}  
 static public ArrayList<Integer> get\_ticketpriceoftheatre() {return *ticketpriceoftheatre*;}  
 static public ArrayList<LocalDate> get\_dateofvisittheatre() {return *dateofvisittheatre*;}  
 static public ArrayList<String> get\_nameofmuseum() {return *nameofmuseum*;}  
  
 static public ArrayList<Integer> get\_ticketpriceofmuseum() {return *ticketpriceofmuseum*;}  
 static public int get\_pricefortypeoffood() {return *pricefortypeoffood*;}  
  
  
 static public int get\_sumofticketsplane() {return *sumofticketsplane*;}  
 static public int get\_sumfortranser() {return *sumfortranser*;}  
 static public int get\_sumforhotel() {return *sumforhotel*;}  
 static public int get\_sumforfood() {return *sumforfood*;}  
 static public int get\_sumformuseums() {return *sumformuseums*;}  
 static public int get\_sumfortheatres() {return *sumfortheatres*;}  
 static public int get\_sum\_full() {return *sum\_full*;}  
 static public ArrayList<Integer> get\_sumeachtheatre() {return *sumeachtheatre*;}  
 static public ArrayList<Integer> get\_sumeachmuseum() {return *sumeachmuseum*;}  
  
  
}

## **СОДЕРЖИМОЕ ФАЙЛА «CLASSFOROUTPUT.JAVA»**

package com.example.kurs;  
  
public class classforoutput  
{  
 private String ob;  
 private String data;  
 private int sum;  
 public classforoutput(String ob, String data, int sum)  
 {  
 this.ob=ob;  
 this.data=data;  
 this.sum=sum;  
 }  
 public classforoutput()  
 {  
  
 }  
  
 public String getOb()  
 {  
 return ob;  
 }  
 public String getData()  
 {  
 return data;  
 }  
 public int getSum()  
 {  
 return sum;  
 }  
  
}

# ДЕМОНСТРАЦИЯ РАБОТЫ ПРОГРАММЫ

На рис. 3.1-3.11 представлена демонстрация работы программы.

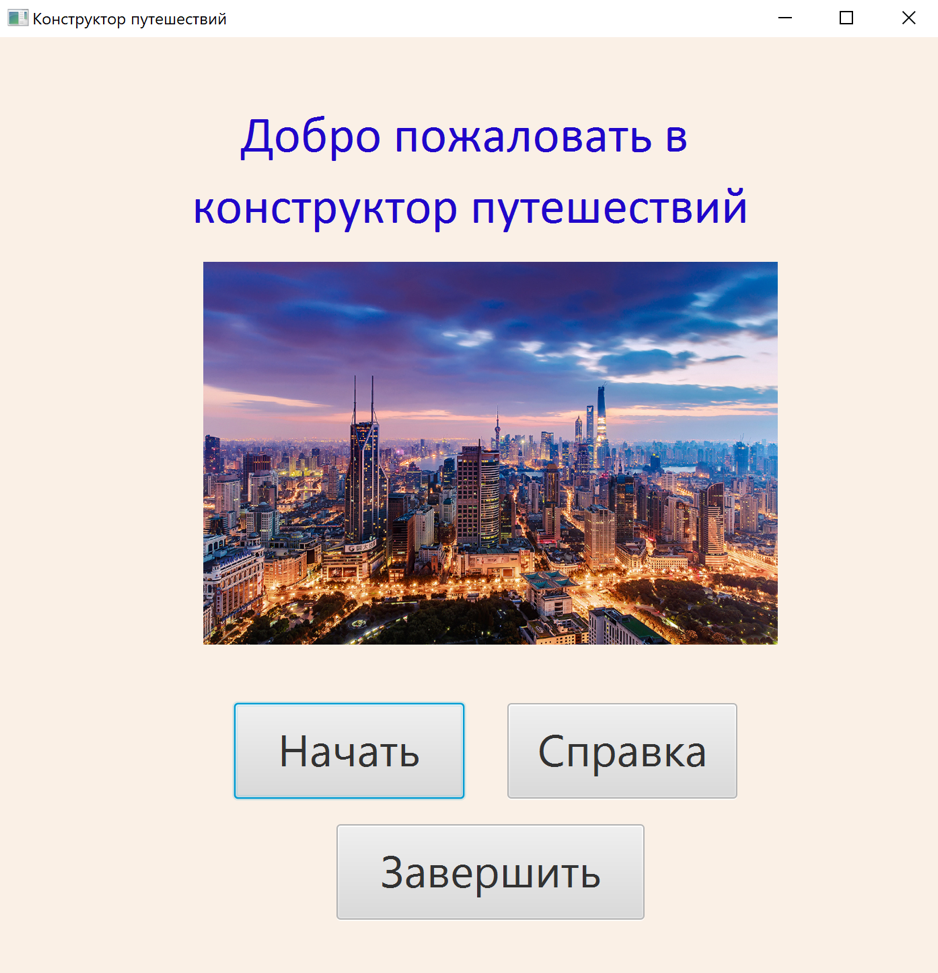


Рис. 3.1. Главное меню



Рис. 3.2. Справка

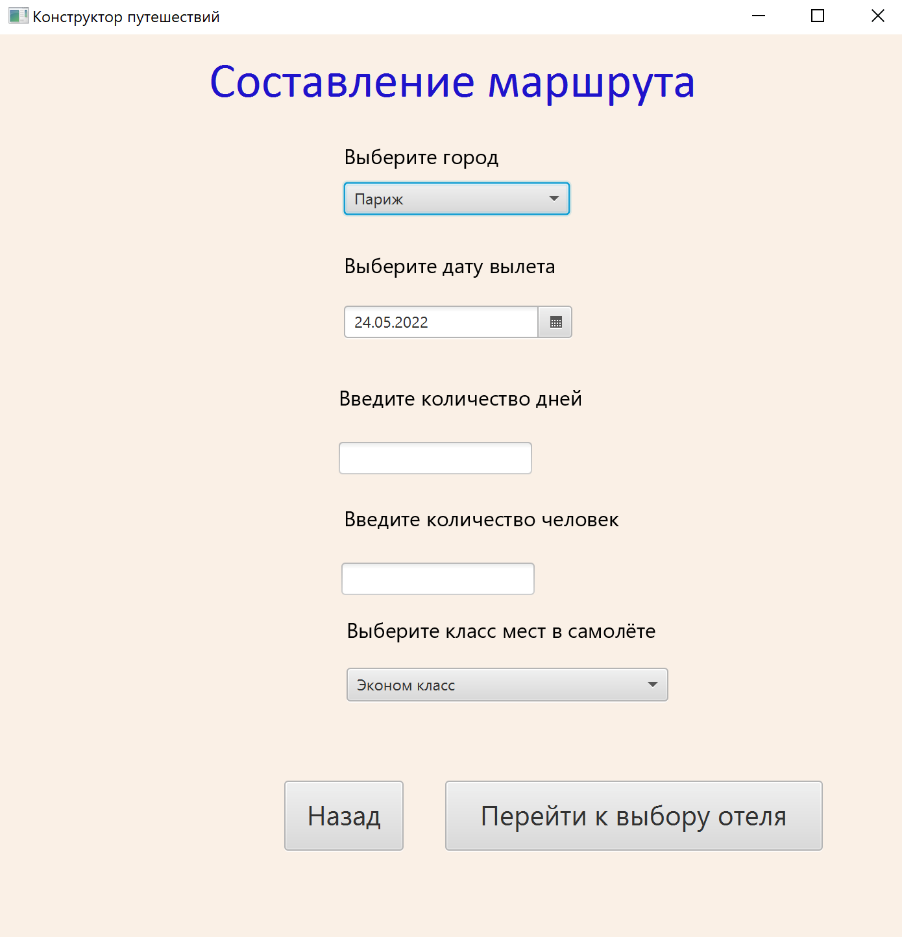


Рис. 3.3. Составление маршрута. Выбор города, даты, тип класса мест в самолёте. Ввод количество дней поездки и количества человек

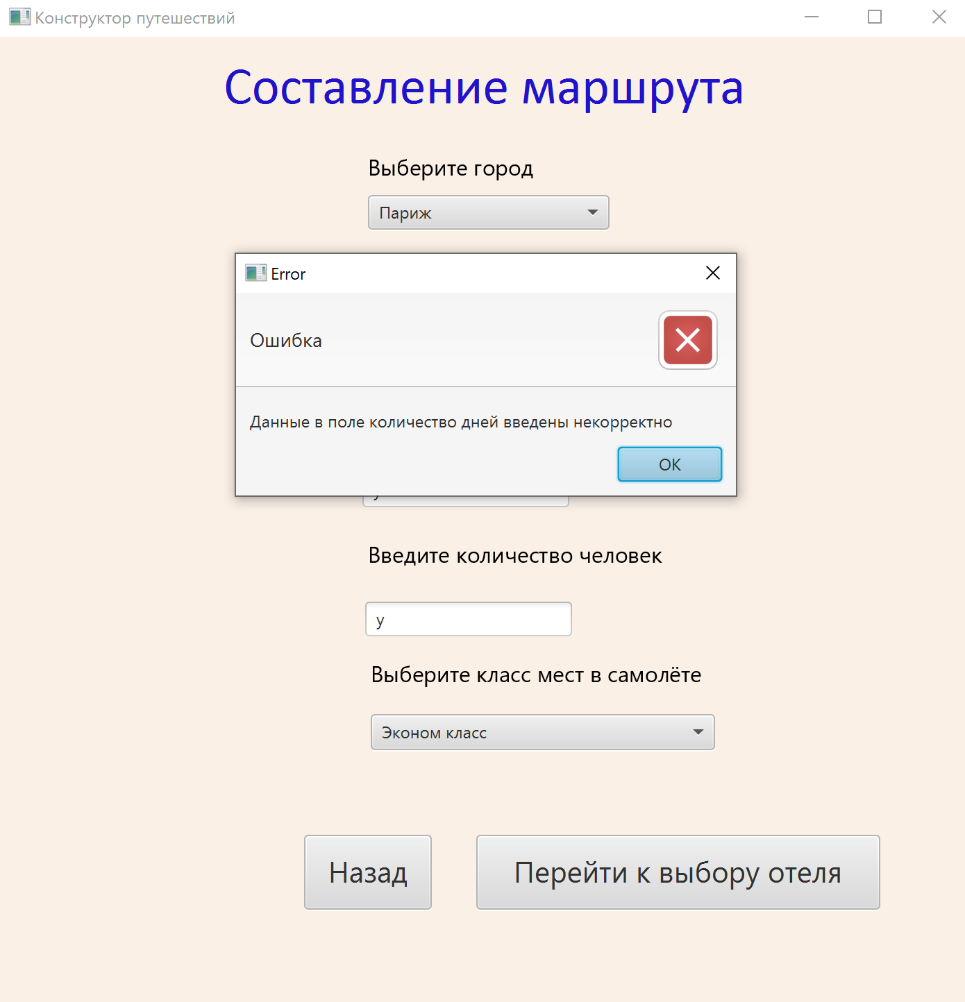


Рис. 3.4. Появление ошибки о некорректном вводе данных в поле количество дней

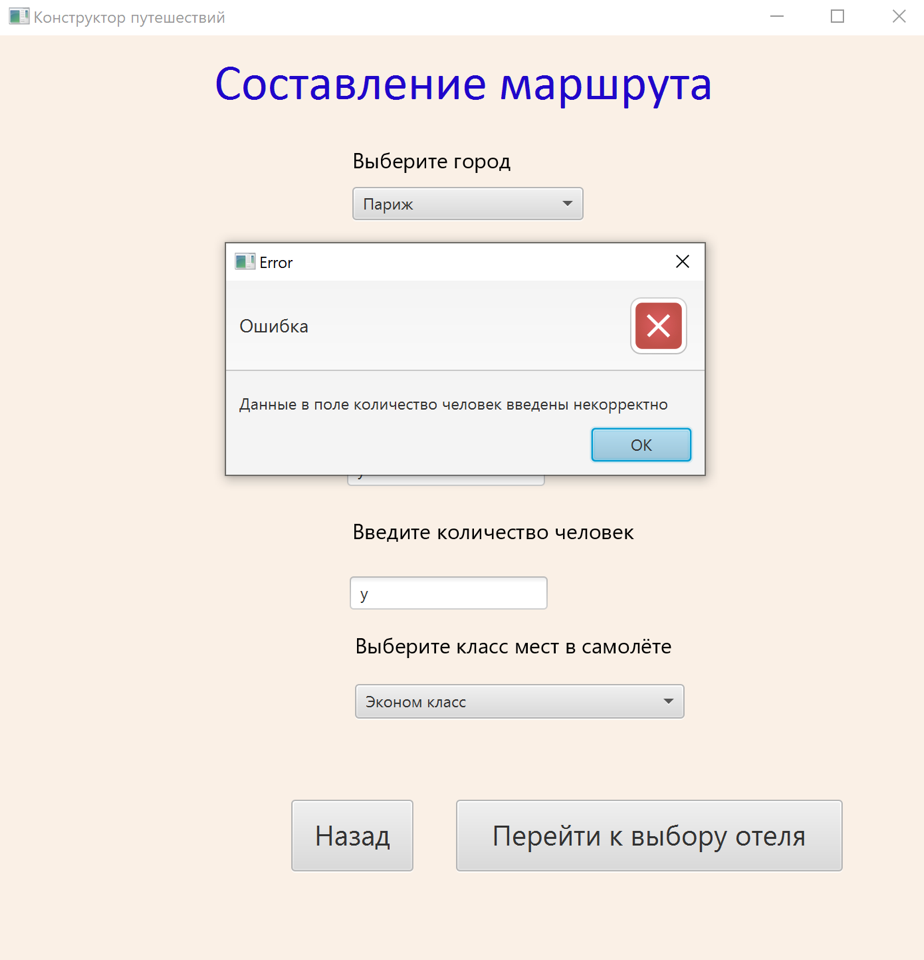


Рис. 3.5. Появление ошибки о некорректном вводе данных в поле количество человек

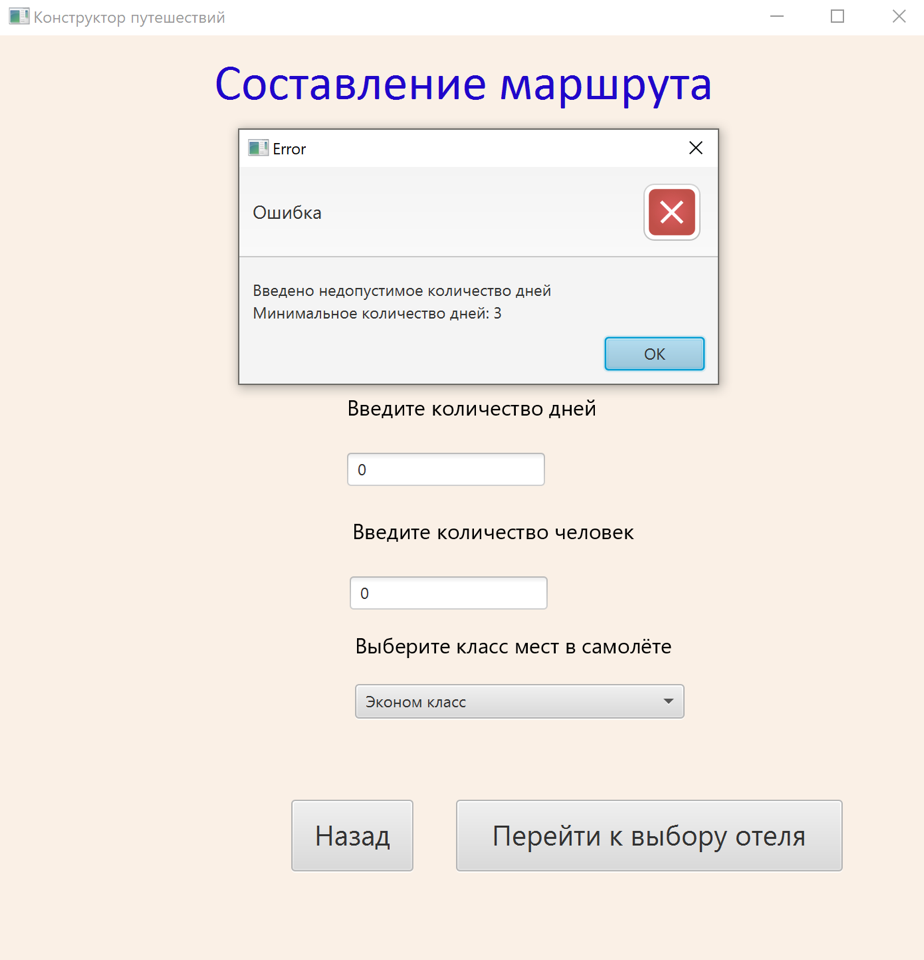


Рис. 3.6. Появление ошибки о неверном вводе данных количества дней поездки

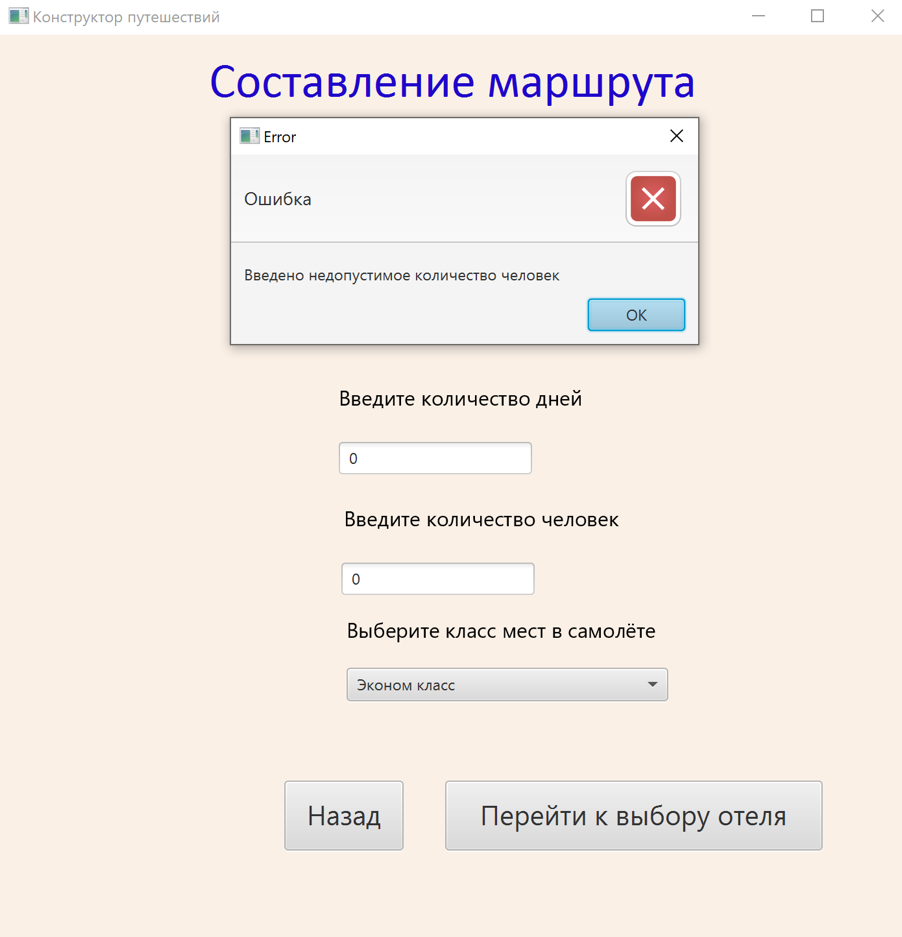


Рис. 3.7. Появление ошибки о неверном вводе данных в поле количества человек

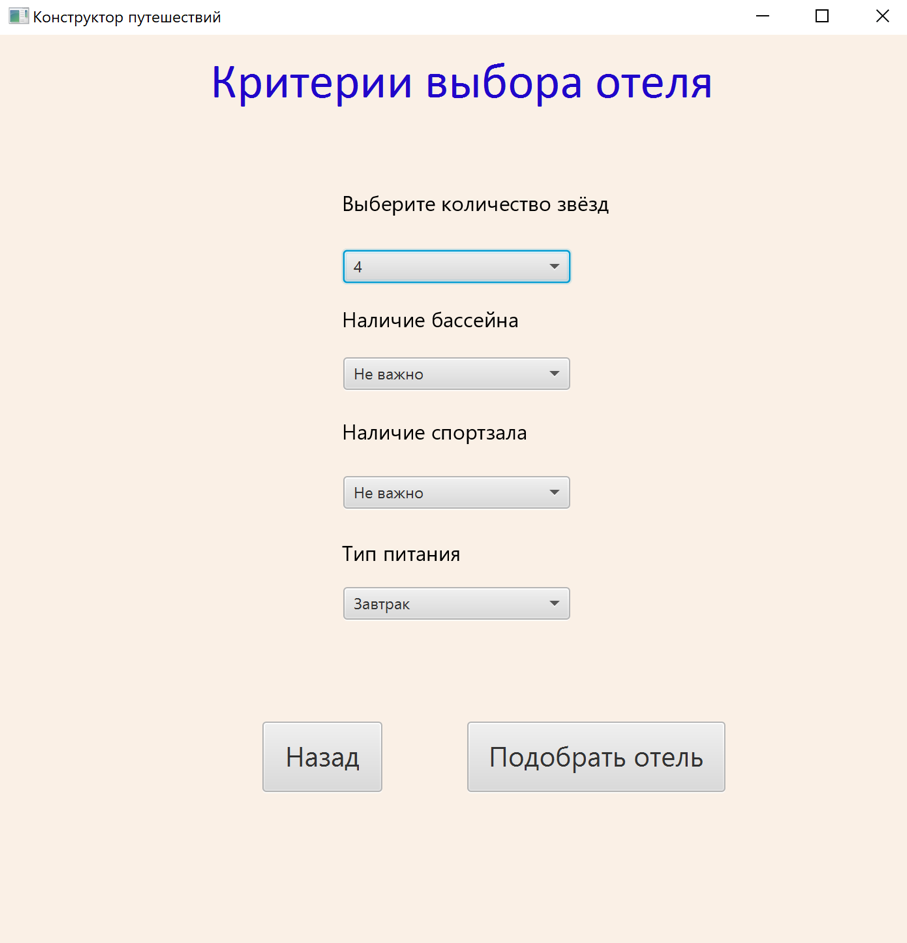


Рис. 3.8. Критерии отбора отеля

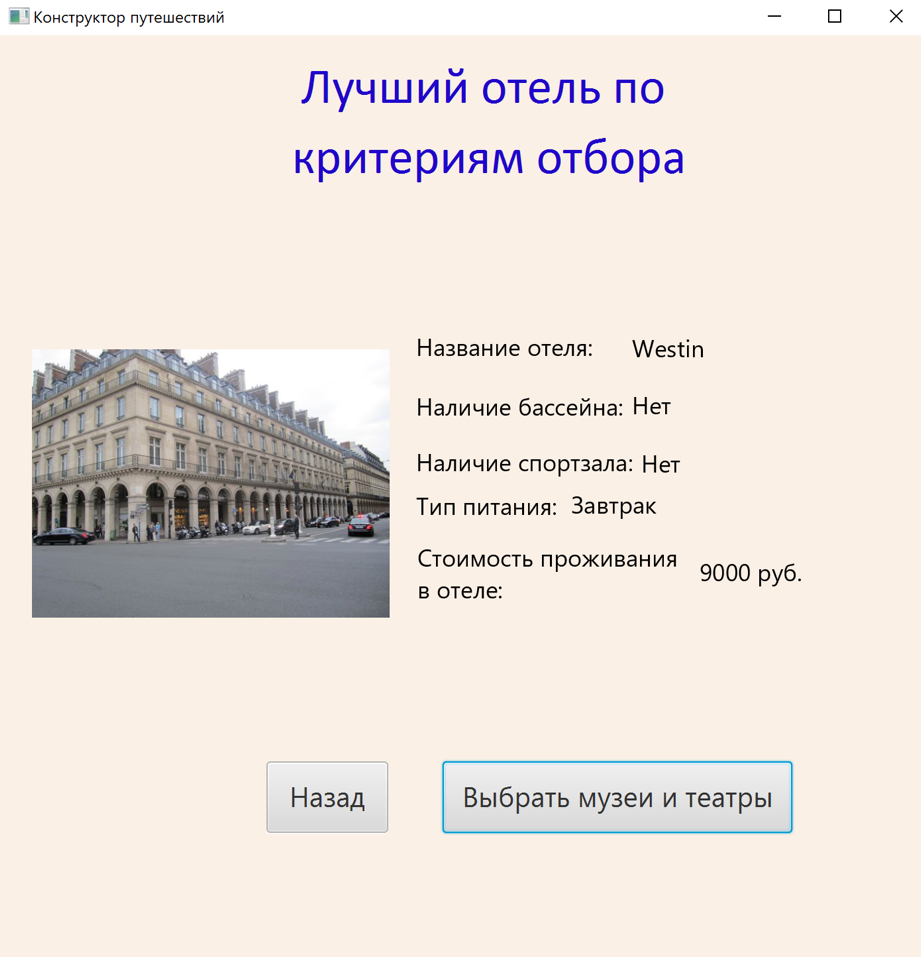


Рис. 3.9. Лучший отель отобранный по критериям отбора

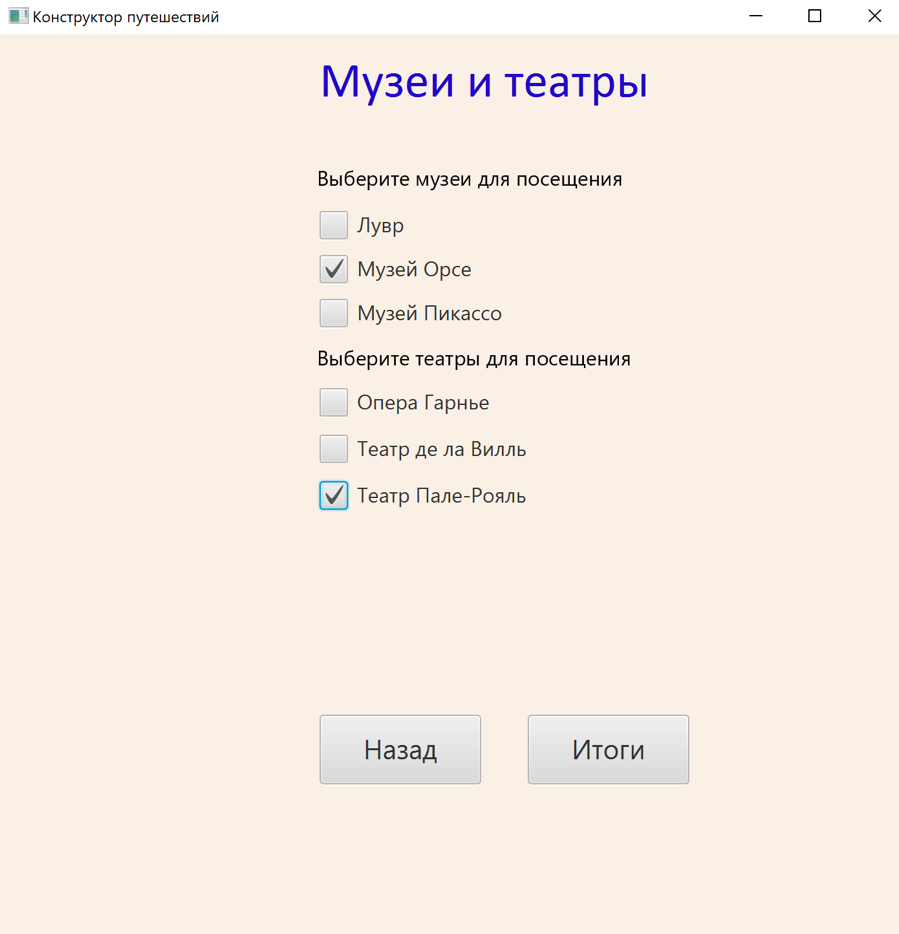


Рис. 3.10. Выбор музеев и театров для посещения

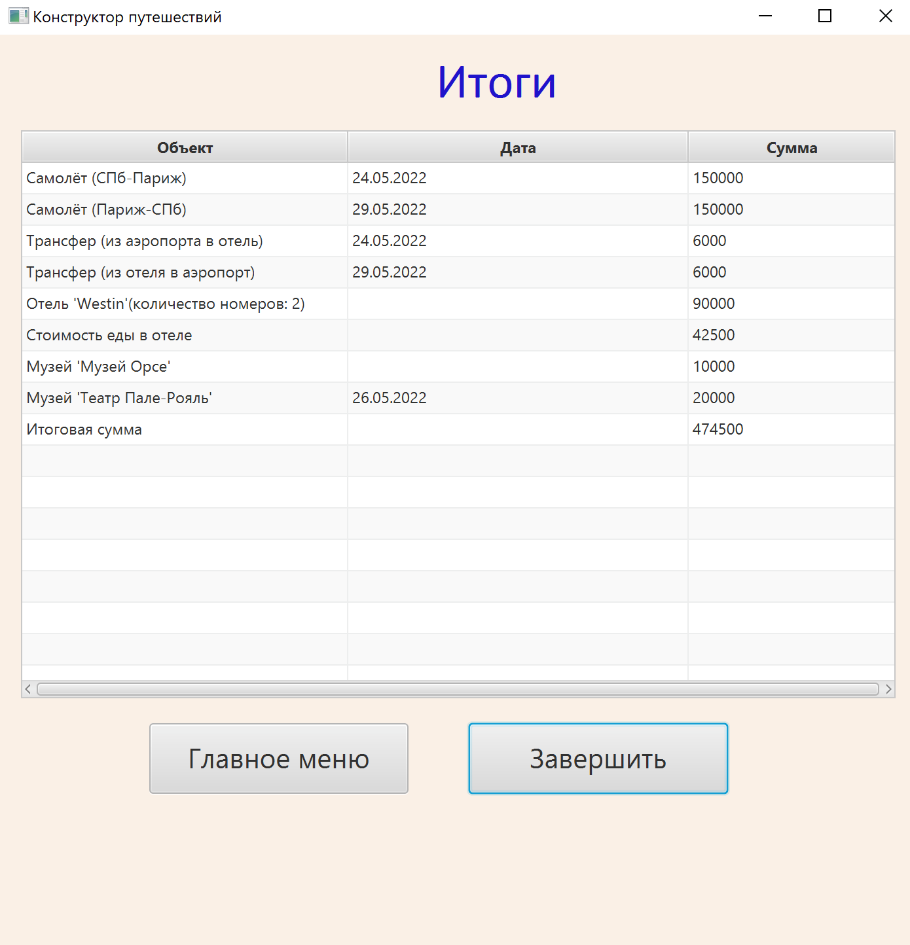


Рис. 3.11. Подсчёт итоговой суммы

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта разработана программа «Конструктор путешествий», позволяющая пользователю спланировать свою поездку. Выполнены основные задачи:

* определены параметры путешествия
* определены классы программы
* определены функции для доступа к закрытым переменным класса
* создан графический интерфейс.

В ходе выполнения практической работы были изучены и применены различные объектно-ориентированные технологии, а именно классы, наследование классов, коллекции, исключение и др. Изучена библиотека «JavaFX» для создания графического интерфейса программы.

В результате было создано 11 классов и 7 файлов формата «.fxml» для описания графических компонент программы и их свойств.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Metanit // Руководство по языку программирования Java [электронный ресурс] – Электрон. дан. Режим доступа: <https://metanit.com/java/tutorial/>, свободный (дата обращения: 14.04.2022)
2. Java // JDK 18 Documentation [электронный ресурс] – Электрон. дан. Режим доступа: <https://docs.oracle.com/en/java/javase/18/index.html>, свободный (дата обращения: 17.04.2022)
3. Java Documentation// The Java Tutorials [электронный ресурс] – Электрон. дан. Режим доступа: <https://docs.oracle.com/javase/tutorial/>, свободный (дата обращения: 19.04.2022)
4. Java FX // Documentation [электронный ресурс] – Электрон. дан. Режим доступа: <https://openjfx.io/>, свободный (дата обращения: 20.04.2022)

# ПРИЛОЖЕНИЯ

*П 1.1*

**Содержимое файла «f1\_mainmenu.fxml»**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.image.ImageView?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="700.0" prefWidth="700.0" style="-fx-background-color: FAF0E6;" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.kurs.f1\_mainmenu">  
 <children>  
 <Label layoutX="137.0" layoutY="42.0" prefHeight="115.0" prefWidth="427.0" text="Добро пожаловать в конструктор путешествий" textAlignment="CENTER" textFill="#2009cb" wrapText="true">  
 <font>  
 <Font name="Calibri" size="38.0" />  
 </font>  
 </Label>  
 <Button fx:id="button\_begin" layoutX="175.0" layoutY="495.0" mnemonicParsing="false" onAction="#click\_button\_begin" prefHeight="53.0" prefWidth="171.0" text="Начать">  
 <font>  
 <Font size="33.0" />  
 </font>  
 </Button>  
 <Button fx:id="button\_inform" layoutX="378.0" layoutY="495.0" mnemonicParsing="false" onAction="#click\_button\_inform" prefHeight="53.0" prefWidth="171.0" text="Справка">  
 <font>  
 <Font size="33.0" />  
 </font>  
 </Button>  
 <ImageView fx:id="field\_picture" fitHeight="296.0" fitWidth="427.0" layoutX="152.0" layoutY="167.0" pickOnBounds="true" preserveRatio="true" />  
 <Button fx:id="button\_finish" layoutX="251.0" layoutY="585.0" mnemonicParsing="false" onAction="#click\_button\_finish" prefHeight="71.0" prefWidth="229.0" text="Завершить">  
 <font>  
 <Font size="33.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

*П 1.2*

**Содержимое файла «f2\_plan.fxml»**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.Insets?>  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.ComboBox?>  
<?import javafx.scene.control.DatePicker?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TextField?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
<?import javafx.scene.text.Text?>  
  
<AnchorPane accessibleRole="COMBO\_BOX" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="700.0" prefWidth="700.0" style="-fx-background-color: FAF0E6;" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.kurs.f2\_plan">  
 <children>  
 <Label layoutX="161.0" layoutY="14.0" prefHeight="44.0" prefWidth="378.0" text="Составление маршрута" textAlignment="CENTER" textFill="#2009cb" wrapText="true">  
 <font>  
 <Font name="Calibri" size="38.0" />  
 </font>  
 </Label>  
 <Text layoutX="265.0" layoutY="100.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Выберите город" wrappingWidth="155.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <ComboBox fx:id="field\_city" accessibleRole="CHECK\_BOX" editable="true" layoutX="265.0" layoutY="114.0" prefHeight="25.0" prefWidth="174.0" />  
 <DatePicker fx:id="field\_data" layoutX="265.0" layoutY="209.0">  
 <opaqueInsets>  
 <Insets />  
 </opaqueInsets></DatePicker>  
 <Text layoutX="265.0" layoutY="184.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Выберите дату вылета" wrappingWidth="184.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <Text layoutX="261.0" layoutY="286.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Введите количество дней" wrappingWidth="201.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <Text layoutX="265.0" layoutY="379.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Введите количество человек" wrappingWidth="217.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <Text layoutX="267.0" layoutY="465.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Выберите класс мест в самолёте" wrappingWidth="248.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <ComboBox fx:id="field\_classofplane" accessibleRole="CHECK\_BOX" editable="true" layoutX="267.0" layoutY="488.0" prefHeight="26.0" prefWidth="248.0" />  
 <Button fx:id="button\_paramhotels" layoutX="343.0" layoutY="575.0" mnemonicParsing="false" onAction="#click\_button\_paramhotels" prefHeight="54.0" prefWidth="291.0" text="Перейти к выбору отеля" textAlignment="CENTER">  
 <font>  
 <Font size="21.0" />  
 </font>  
 </Button>  
 <Button fx:id="button\_back" layoutX="219.0" layoutY="575.0" mnemonicParsing="false" onAction="#click\_button\_back" prefHeight="54.0" prefWidth="92.0" text="Назад">  
 <font>  
 <Font size="21.0" />  
 </font></Button>  
 <TextField fx:id="field\_days" layoutX="261.0" layoutY="314.0" />  
 <TextField fx:id="field\_persons" layoutX="263.0" layoutY="407.0" />  
 </children>  
</AnchorPane>

*П 1.3*

**Содержимое файла «f3\_charactofhotels.fxml»**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.ComboBox?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
<?import javafx.scene.text.Text?>  
  
<AnchorPane accessibleRole="COMBO\_BOX" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="700.0" prefWidth="700.0" style="-fx-background-color: FAF0E6;" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.kurs.f3\_charactofhotels">  
 <children>  
 <Label layoutX="161.0" layoutY="14.0" prefHeight="44.0" prefWidth="393.0" text="Критерии выбора отеля" textAlignment="CENTER" textFill="#2009cb" wrapText="true">  
 <font>  
 <Font name="Calibri" size="38.0" />  
 </font>  
 </Label>  
 <Text layoutX="262.0" layoutY="135.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Выберите количество звёзд" wrappingWidth="225.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <ComboBox fx:id="field\_stars" accessibleRole="CHECK\_BOX" editable="true" layoutX="263.0" layoutY="165.0" prefHeight="25.0" prefWidth="174.0" />  
 <Text layoutX="262.0" layoutY="224.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Наличие бассейна" wrappingWidth="149.0">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <Button fx:id="button\_showvariants" layoutX="358.0" layoutY="526.0" mnemonicParsing="false" onAction="#click\_button\_showvariants" prefHeight="54.0" prefWidth="198.0" text="Подобрать отель" textAlignment="CENTER">  
 <font>  
 <Font size="21.0" />  
 </font>  
 </Button>  
 <Button fx:id="button\_back" layoutX="201.0" layoutY="526.0" mnemonicParsing="false" onAction="#click\_button\_back" prefHeight="54.0" prefWidth="92.0" text="Назад">  
 <font>  
 <Font size="21.0" />  
 </font></Button>  
 <ComboBox fx:id="field\_pool" accessibleRole="CHECK\_BOX" editable="true" layoutX="263.0" layoutY="247.0" prefHeight="25.0" prefWidth="174.0" />  
 <Text layoutX="262.0" layoutY="310.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Наличие спортзала" wrappingWidth="149.0">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <ComboBox fx:id="field\_gym" accessibleRole="CHECK\_BOX" editable="true" layoutX="263.0" layoutY="338.0" prefHeight="25.0" prefWidth="174.0" />  
 <Text layoutX="262.0" layoutY="403.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Тип питания" wrappingWidth="149.0">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <ComboBox fx:id="field\_food" accessibleRole="CHECK\_BOX" editable="true" layoutX="263.0" layoutY="423.0" prefHeight="25.0" prefWidth="174.0" />  
 </children>  
</AnchorPane>

*П 1.4*

**Содержимое файла «f4\_showhotel.fxml»**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.image.ImageView?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
<?import javafx.scene.text.Text?>  
  
<AnchorPane accessibleRole="COMBO\_BOX" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="700.0" prefWidth="700.0" style="-fx-background-color: FAF0E6;" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.kurs.f4\_showhotel">  
 <children>  
 <Label layoutX="208.0" layoutY="14.0" prefHeight="103.0" prefWidth="322.0" text="Лучший отель по критериям отбора" textAlignment="CENTER" textFill="#2009cb" wrapText="true">  
 <font>  
 <Font name="Calibri" size="38.0" />  
 </font>  
 </Label>  
 <Button fx:id="button\_museums\_and\_theatres" layoutX="334.0" layoutY="548.0" mnemonicParsing="false" onAction="#click\_button\_museums\_and\_theatres" prefHeight="54.0" prefWidth="264.0" text="Выбрать музеи и театры" textAlignment="CENTER">  
 <font>  
 <Font size="21.0" />  
 </font>  
 </Button>  
 <Button fx:id="button\_back" layoutX="201.0" layoutY="548.0" mnemonicParsing="false" onAction="#click\_button\_back" prefHeight="54.0" prefWidth="92.0" text="Назад">  
 <font>  
 <Font size="21.0" />  
 </font></Button>  
 <ImageView fx:id="field\_picture" fitHeight="230.0" fitWidth="270.0" layoutX="24.0" layoutY="237.0" pickOnBounds="true" preserveRatio="true" />  
 <Text layoutX="314.0" layoutY="242.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Название отеля: " wrappingWidth="159.13671875">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Text>  
 <Text layoutX="314.0" layoutY="287.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Наличие бассейна: " wrappingWidth="166.13671875">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Text>  
 <Text layoutX="314.0" layoutY="329.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Наличие спортзала: " wrappingWidth="174.13671875">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Text>  
 <Text layoutX="315.0" layoutY="401.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Стоимость проживания в отеле: " wrappingWidth="208.13671875">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Text>  
 <Text fx:id="field\_hotelname" layoutX="477.0" layoutY="243.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Text" wrappingWidth="166.13671875">  
 <font>  
 <Font size="18.0" />  
 </font></Text>  
 <Text fx:id="field\_pool" layoutX="477.0" layoutY="286.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Text" wrappingWidth="166.13671875">  
 <font>  
 <Font size="18.0" />  
 </font></Text>  
 <Text fx:id="field\_gym" layoutX="484.0" layoutY="330.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Text" wrappingWidth="166.13671875">  
 <font>  
 <Font size="18.0" />  
 </font></Text>  
 <Text fx:id="field\_costperday" layoutX="528.0" layoutY="412.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Text" wrappingWidth="137.13671875">  
 <font>  
 <Font size="18.0" />  
 </font></Text>  
 <Text layoutX="314.0" layoutY="362.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Тип питания:" wrappingWidth="137.13671875">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Text>  
 <Text fx:id="field\_typeoffood" layoutX="431.0" layoutY="361.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Text" wrappingWidth="166.13671875">  
 <font>  
 <Font size="18.0" />  
 </font>  
 </Text>  
 </children>  
</AnchorPane>

*П 1.5*

**Содержимое файла «f5\_museumsandtheatres.fxml»**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.CheckBox?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
<?import javafx.scene.text.Text?>  
  
<AnchorPane accessibleRole="COMBO\_BOX" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="700.0" prefWidth="700.0" style="-fx-background-color: FAF0E6;" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.kurs.f5\_museumsandtheatres">  
 <children>  
 <Label layoutX="247.0" layoutY="14.0" prefHeight="44.0" prefWidth="393.0" text="Музеи и театры" textAlignment="CENTER" textFill="#2009cb" wrapText="true">  
 <font>  
 <Font name="Calibri" size="38.0" />  
 </font>  
 </Label>  
 <Text layoutX="245.0" layoutY="117.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Выберите музеи для посещения" wrappingWidth="274.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <Button fx:id="button\_finalsum" layoutX="408.0" layoutY="526.0" mnemonicParsing="false" onAction="#click\_button\_button\_finalsum" prefHeight="54.0" prefWidth="125.0" text="Итоги" textAlignment="CENTER">  
 <font>  
 <Font size="21.0" />  
 </font>  
 </Button>  
 <Button fx:id="button\_back" layoutX="247.0" layoutY="526.0" mnemonicParsing="false" onAction="#click\_button\_back" prefHeight="54.0" prefWidth="125.0" text="Назад">  
 <font>  
 <Font size="21.0" />  
 </font></Button>  
 <Text layoutX="245.0" layoutY="256.0" strokeType="OUTSIDE" strokeWidth="0.0" text="Выберите театры для посещения" wrappingWidth="274.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <CheckBox fx:id="checkbox\_1museum" layoutX="247.0" layoutY="135.0" mnemonicParsing="false">  
 <font>  
 <Font size="16.0" />  
 </font></CheckBox>  
 <CheckBox fx:id="checkbox\_2museum" layoutX="247.0" layoutY="169.0" mnemonicParsing="false">  
 <font>  
 <Font size="16.0" />  
 </font></CheckBox>  
 <CheckBox fx:id="checkbox\_3museum" layoutX="247.0" layoutY="203.0" mnemonicParsing="false">  
 <font>  
 <Font size="16.0" />  
 </font></CheckBox>  
 <CheckBox fx:id="checkbox\_1theatre" layoutX="247.0" layoutY="272.0" mnemonicParsing="false">  
 <font>  
 <Font size="16.0" />  
 </font></CheckBox>  
 <CheckBox fx:id="checkbox\_2theatre" layoutX="247.0" layoutY="308.0" mnemonicParsing="false">  
 <font>  
 <Font size="16.0" />  
 </font></CheckBox>  
 <CheckBox fx:id="checkbox\_3theatre" layoutX="247.0" layoutY="344.0" mnemonicParsing="false">  
 <font>  
 <Font size="16.0" />  
 </font></CheckBox>  
 </children>  
</AnchorPane>

*П 1.6*

**Содержимое файла «f6\_finalsum.fxml»**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.control.TableColumn?>  
<?import javafx.scene.control.TableView?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
  
<AnchorPane accessibleRole="COMBO\_BOX" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="700.0" prefWidth="700.0" style="-fx-background-color: FAF0E6;" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.kurs.f6\_finalsum">  
 <children>  
 <Label layoutX="333.0" layoutY="14.0" prefHeight="44.0" prefWidth="98.0" text="Итоги" textAlignment="CENTER" textFill="#2009cb" wrapText="true">  
 <font>  
 <Font name="Calibri" size="38.0" />  
 </font>  
 </Label>  
 <Button fx:id="button\_finish" layoutX="358.0" layoutY="526.0" mnemonicParsing="false" onAction="#click\_button\_finish" prefHeight="54.0" prefWidth="198.0" text="Завершить" textAlignment="CENTER">  
 <font>  
 <Font size="21.0" />  
 </font>  
 </Button>  
 <Button fx:id="button\_mainmenu" layoutX="114.0" layoutY="526.0" mnemonicParsing="false" onAction="#click\_button\_mainmenu" prefHeight="54.0" prefWidth="198.0" text="Главное меню">  
 <font>  
 <Font size="21.0" />  
 </font></Button>  
 <TableView fx:id="field\_table" layoutX="16.0" layoutY="73.0" prefHeight="434.0" prefWidth="668.0">  
 <columns>  
 <TableColumn fx:id="field\_object" prefWidth="249.0" text="Объект" />  
 <TableColumn fx:id="field\_data" prefWidth="260.0" text="Дата" />  
 <TableColumn fx:id="field\_sum" prefWidth="158.0" text="Сумма" />  
 </columns>  
 </TableView>  
 </children>  
</AnchorPane>

*П 1.7*

**Содержимое файла «information.fxml»**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.Button?>  
<?import javafx.scene.control.Label?>  
<?import javafx.scene.layout.AnchorPane?>  
<?import javafx.scene.text.Font?>  
<?import javafx.scene.text.Text?>  
  
<AnchorPane accessibleRole="COMBO\_BOX" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="700.0" prefWidth="700.0" style="-fx-background-color: FAF0E6;" xmlns="http://javafx.com/javafx/18" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.kurs.information">  
 <children>  
 <Label layoutX="279.0" layoutY="14.0" prefHeight="44.0" prefWidth="142.0" text="Справка" textFill="#2009cb" wrapText="true">  
 <font>  
 <Font name="Calibri" size="38.0" />  
 </font>  
 </Label>  
 <Text layoutX="14.0" layoutY="80.0" strokeType="OUTSIDE" strokeWidth="0.0" text=" Данный конструктор путешествий позволяет спланировать маршрут для поездки такие города как Берлин, Париж и Рим. В начале вы выбираете город назначения, затем дату вылета, далее вводите количество дней, которое хотите отвести для посещения города (минимальное количество дней равно 3) и количество человек, участвующих в поездке (количество человек влияет на количество номеров необходимых для бронирования в отеле (1 номер предлагается для 3 человек)). Затем выбираете класс самолёта, на котором хотите лететь в выбранный вами город. Затем вы отбираете отель по характеристикам, таким как количество звёзд, наличие бассейна, наличие спортзала и типа питания. Далее вам предлагается лучший отель по вашим характеристикам и с самой низкой ценой. Затем вы можете увидеть фотографию отобранного для вас отеля и сверить характеристики, по которым данный отель был отобран для Вас. Затем вам предлагаются музеи и театры для посещения в городе. Далее производится подсчёт расходов, на котором подробно расписаны компоненты поездки и их стоимость. По заврешении подсчёта итогов вам предлагается завершить работу программы или начать сначала." wrappingWidth="664.13671875">  
 <font>  
 <Font size="16.0" />  
 </font>  
 </Text>  
 <Button fx:id="button\_mainmenu" layoutX="205.0" layoutY="518.0" mnemonicParsing="false" onAction="#click\_button\_mainmenu" prefHeight="54.0" prefWidth="291.0" text="Главное меню" textAlignment="CENTER">  
 <font>  
 <Font size="21.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>