

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Волгоградский государственный технический университет»

Факультет Экономики и управления

Кафедра «Информационные системы в экономике»

Специальность (направление) 09.03.03 «Прикладная информатика»

Дисциплина «Проектирование информационных систем»

Утверждаю  
Зав. кафедрой \_\_\_\_\_

«\_\_\_» \_\_\_\_\_ 202\_ г.

**ЗАДАНИЕ**  
**на курсовую работу**

Студент Афонасенкова Елизавета Олеговна  
(фамилия, имя, отчество)

Группа ЭП-461

1. Тема: Разработка приложения по выбору экскурсий

Утверждено приказом от «\_\_» \_\_\_\_\_ 20\_\_ г. №\_\_

2. Срок представления работы к защите «\_\_» \_\_\_\_\_ 202\_ г.

3. Содержание расчетно-пояснительной записки: \_\_\_\_\_

Введение; Описание модели приложения; Описание программного решения  
и примеры кода; Экранные формы приложения; Заключение.

4. Перечень графического материала: Рисунков – 15;

5. Дата выдачи задания «\_\_» \_\_\_\_\_ 20\_\_ г.

Руководитель работы \_\_\_\_\_ Н. В. Кетько  
подпись, дата инициалы и фамилия

Задание принял к исполнению \_\_\_\_\_ Е.О. Афонасенкова  
подпись, дата инициалы и фамилия

## Содержание

Введение .....	4
1. Описание модели приложения.....	6
2. Описание программного решения и примеры кода .....	8
3. Экранные формы приложения.....	12
Заключение .....	18
Список литературы .....	19
Приложение А .....	20

## Введение

Java Swing представляет собой мощный и гибкий набор библиотек и компонентов для создания графических пользовательских интерфейсов (GUI) в приложениях, разрабатываемых на языке программирования Java. Эта технология, впервые представленная в Java 1.2, стала важным инструментом для разработчиков, стремящихся создавать современные и интерактивные приложения с привлекательными пользовательскими интерфейсами.

Современные технологии в сфере разработки программного обеспечения играют важную роль в улучшении и упрощении различных сфер жизни. Одним из направлений, где применение программных решений становится все более значимым, является сфера туризма. Развитие технологий позволяет создавать удобные и функциональные приложения, способствующие более комфортному и интересному путешествию.

Целью данной курсовой работы является разработка приложения для организации и проведения экскурсий по России с использованием технологии Java Swing. Это приложение предназначено для туристов и гидов, предоставляя удобные инструменты для выбора, планирования и реализации экскурсионных маршрутов по различным регионам страны.

Актуальность темы курсовой работы определяется современными требованиями к туристической индустрии и возможностями технологий. С развитием туристической индустрии растет и спрос на новые, удобные и интерактивные способы организации экскурсий. Приложение на Java Swing может предоставить туристам удобный и привлекательный интерфейс для выбора, планирования экскурсий по городам России и записи на них.

Задачами данной курсовой работы являются:

- Описать модель приложения и сформулировать постановку задачи;
- Описать программные решения и представить части кода программы;

- Представить экранные формы работы приложения при различных выборах;

В рамках работы будет рассмотрен процесс проектирования и реализации графического интерфейса приложения, а также функциональности, необходимой для эффективной организации и управления экскурсионными маршрутами. Использование технологии Java Swing обеспечит кроссплатформенность и интуитивно понятный интерфейс, делая приложение доступным для широкого круга пользователей.

## 1. Описание модели приложения

Разработка моделей является неотъемлемой частью создания сложных систем, даже если она – небольшая. Важность этого процесса особенно возрастает, если система должна быть точной, производительной, прибыльной или отвечать важным требованиям.

Методология проектирования систем получила широкое распространение с появлением языка объектного моделирования нового поколения - унифицированного языка моделирования Unified Modeling Language (UML). Данный язык предназначен для визуального моделирования и проектирования информационных систем. Можно сказать, что UML — это набор правил, по которым нужно рисовать схемы. Зная его, можно быстро создавать универсальные графические представления сложных процессов и структур. Именно поэтому IT-специалисты во время разработки ПО часто используют UML-моделирование и проектирование процессов.

Диаграмма вариантов использования (use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

Назначение данной диаграммы состоит в следующем: проектируемая программная система представляется в форме так называемых вариантов использования, с которыми взаимодействуют внешние сущности или актеры. При этом актером или действующим лицом называется любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая служит источником воздействия на моделируемую систему так, как определит разработчик. Вариант использования служит для описания сервисов, которые система предоставляет актеру.

Другими словами каждый вариант использования определяет набор действий, совершаемый системой при диалоге с актером. На рисунке 1

представлена диаграмма вариантов использования для разрабатываемого приложения.

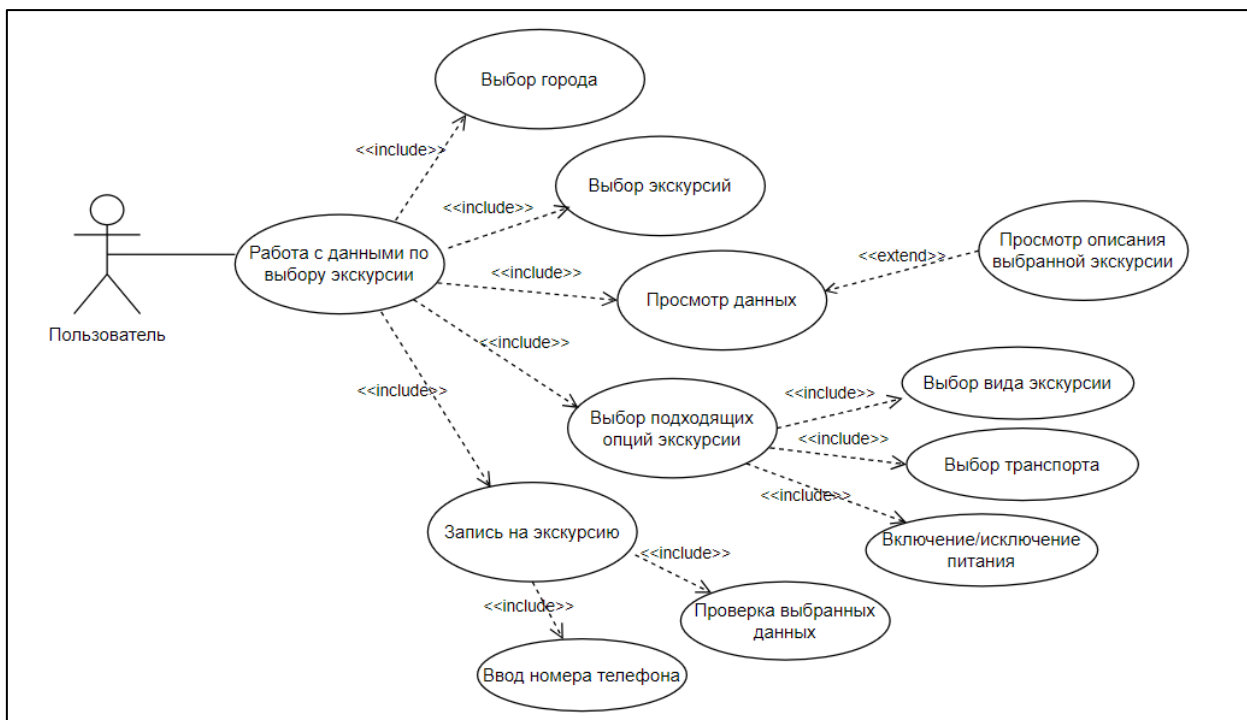


Рисунок 1 – Диаграмма вариантов использования

#### Постановка задачи:

Пользователь заходит в приложение для выбора экскурсий по городам России. Пользователь выбирает город, где бы он хотел пойти на экскурсию. Выбрав город, появляется список экскурсий в этом городе. Пользователь может ознакомиться со всеми экскурсиями, прочитав описание. Далее, пользователь может перейти на вкладку «Запись на экскурсию», где отражаются выбранные им город и экскурсия, а также просмотрев картинки с данной экскурсии. Пользователю необходимо выбрать вид экскурсии: групповая/индивидуальная, тип транспорта: автобус/маршрутка, а также необходимость включения или исключения питания во время экскурсии. Каждый выбор отличается по ценам, с которыми пользователь может ознакомиться в дополнительной информации. Выбрав все необходимые опции, пользователь может посмотреть полную цену за поездку, и при условии, что его все устраивает, может записаться на экскурсию, введя свой номер телефона. Происходит проверка выбранных данных и пользователя

записывают на выбранную экскурсию, а также уведомляют о скором звонке для выбора времени и даты. Если пользователь хочет записаться еще раз, ему необходимо все очистить и повторно провести вышеописанные действия.

## 2. Описание программного решения и примеры кода

Для разработки приложения используется встроенная в язык программирования Java библиотека Swing. Для отображения элементов графического пользовательского интерфейса необходимо создать фрейм (окно) и отобразить его. Часть кода приложения для отображения окна:

```
JFrame jf=new JFrame("Экскурсии по России");
jf.getContentPane().setLayout(new BorderLayout());
jf.setPreferredSize(new Dimension(1450,650));
jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
jf.setVisible(true);
```

Пример создания вкладок JTabbedPane и двух JLabel, на которых располагаются все компоненты первой и второй вкладок соответственно:

```
final String logo = "D:\\фото\\cart3.jpg";
ImageIcon background = new ImageIcon(logo);
JTabbedPane tappane=new JTabbedPane(JTabbedPane.TOP,
JTabbedPane.SCROLL_TAB_LAYOUT);
JLabel allLabel=new JLabel(background);
JLabel shapLabel = new JLabel(background);

allLabel.setLayout(new GridBagLayout());
allLabel.setOpaque(true);
shapLabel.setLayout(new GridBagLayout());
shapLabel.setOpaque(true);

for (int i = 1; i < colors.length; i++) {
    tappane.setBackground(colors[i - 1]);
    // Добавление вкладки
    tappane.addTab(String.format("Выбор экскурсии", i), allLabel);
    tappane.addTab(String.format("Запись на экскурсию", i), shapLabel);
    // Подключение мнемоники
    tappane.setMnemonicAt(i-1, String.valueOf(i).charAt(0));
}
jf.add(tappane);
```

Как видно из вышеописанного кода для размещения компонентов на JLabel используется диспетчер компоновки GridBagLayout, что позволяет более точно размещать компоненты в виде табличной сетки.

Пример добавления компонента JLabel и JTextField на JLabel:

```

cenLabel.setText("Цена экскурсии");
cenLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=3;
gridCons.gridy=2;
gridCons.anchor=GridBagConstraints.WEST;
gridCons.insets=new Insets(180,20,0,0);
allLabel.add(cenLabel,gridCons);

cenTextF.setPreferredSize(new Dimension(120,25));
cenTextF.setEditable(false);
cenTextF.setFont(myFont1);
cenTextF.setBackground(Color.decode("#FFefd5"));
cenTextF.setBorder(BorderFactory.createLineBorder(Color.BLACK));
gridCons=new GridBagConstraints();
gridCons.gridx=3;
gridCons.gridy=2;
gridCons.insets=new Insets(180,190,0,0);
allLabel.add(cenTextF,gridCons);

```

Пример создания выплывающего списка и его добавление на JLabel:

```

String[] items = {
    "Выберете город",
    "Владивосток",
    "Владимир",
    "Ростов Великий",
    "Санкт-Петербург",
    "Москва",
    "Ярославль"
};
JComboBox comboBox = new JComboBox(items);

comboBox.setFont(myFont1);
comboBox.setBackground(Color.decode("#98FB98"));
gridCons=new GridBagConstraints();
gridCons.gridx=1;
gridCons.gridy=2;
gridCons.insets=new Insets(-15,40,0,30);
allLabel.add(comboBox,gridCons);

```

Пример создания кнопки JButton и привязка обработчика событий:



```
JButton rezButton=new JButton();

rezButton.setText("Запись");
rezButton.setBackground(Color.decode("#FFEEFD5"));
zpLabel.setEnabled(false);
zpTextF.setEnabled(false);
gridCons=new GridBagConstraints();
gridCons.gridx=4;
gridCons.gridy=6;
gridCons.insets=new Insets(10,0,50,0);
allLabel.add(rezButton,gridCons);
rezButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        rezButtonActionPerformed(e);
    }
});
```

Пример создания кнопок JRadioButton:

```

ButtonGroup factorButtonGroup=new ButtonGroup();
JRadioButton[] factorRadioButton=new JRadioButton[4];
int x=0; int y=0;
GridBagConstraints gridCons;
uslTA.removeAll();
for (int i = 0; i < 3; i++) {
    factorRadioButton[i] = new JRadioButton();
    factorRadioButton[i].setBackground(Color.decode("#98FB98"));
    factorButtonGroup.add(factorRadioButton[i]);
    factorRadioButton[i].setFont(myFont1);
    gridCons = new GridBagConstraints();
    if (i < 2) {
        gridCons.gridx = x;
        gridCons.gridy = y;
    } else {
        gridCons.gridx = 0;
        gridCons.gridy = 0;
        gridCons.gridwidth = 2;
    }
    if (x > 0) {
        x = 0;
        y++;
    }
    x++;
    if ((String)comboBox.getSelectedItem()=="Выберете город") {
        uslTA.setText("Выберете город!");
    }else uslTA.setText("");
    if ((String)comboBox.getSelectedItem()=="Владивосток") {
        if (i == 0) factorRadioButton[i].setText("Топ 5 лучших
мест");
        if (i == 1) factorRadioButton[i].setText("Исторический
центр Владивостока");
        if (i == 2) factorRadioButton[i].setText("Остров
Русский");
        uslTA.add(factorRadioButton[i], gridCons);
    }
    if ((String)comboBox.getSelectedItem()=="Владимир") {
        if (i == 0) factorRadioButton[i].setText("Легенды
Древнего Владимира");
        if (i == 1) factorRadioButton[i].setText("Старейшие
храмы");
        if (i == 2) factorRadioButton[i].setText("Смотровые
площадки");
        uslTA.add(factorRadioButton[i], gridCons);
    }...
}
uslTA.revalidate();
uslTA.repaint();

```

Пример создания JTextArea с полосой прокрутки:

```

JTextArea textField=new JTextArea();
JPanel analysisTA=new JPanel();
...
if (zpButton.getText().equals("Выбрать")) {
    textField.setText("...");
}
...
textField.setLineWrap(true);
textField.setWrapStyleWord(true);
textField.setEditable(false);
textField.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
textField.setFont(myFont1);
textField.setCaretPosition(0);
JScrollPane scrollPane = new JScrollPane(textField);
analysisTA.add(scrollPane);

```

Создание слайд-шоу:

```
imageLabel.setPreferredSize(new Dimension(550,450));
gridCons=new GridBagConstraints();
gridCons.anchor=GridBagConstraints.CENTER;
shapLabel.add(imageLabel,gridCons);
...
private void setCurrentImage() {
    ImageIcon imageIcon = new ImageIcon((images[currentIndex]));
    Image scaledImage =
imageIcon.getImage().getScaledInstance(imageLabel.getWidth(),
imageLabel.getHeight(), Image.SCALE_DEFAULT);
    imageLabel.setIcon(new ImageIcon(scaledImage));
}
...
if ((String)comboBox.getSelectedItem()=="Владивосток") {
    images = new String[]{"D:\\фото\\vladiv1.jpg",
"D:\\фото\\vladiv2.jpg", "D:\\фото\\vladiv3.jpg"};
    setCurrentImage();
    timer = new Timer(2000, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            currentIndex = (currentIndex + 1) % images.length;
            setCurrentImage(); // Обновляем изображение
        }
    });
}
...
timer.start();
imageLabel.setBorder(BorderFactory.createLineBorder(Color.GREEN,3));
imageLabel.repaint();
```

### 3. Экранные формы приложения

Главное окно приложения при открытии представлено на рисунке 2.

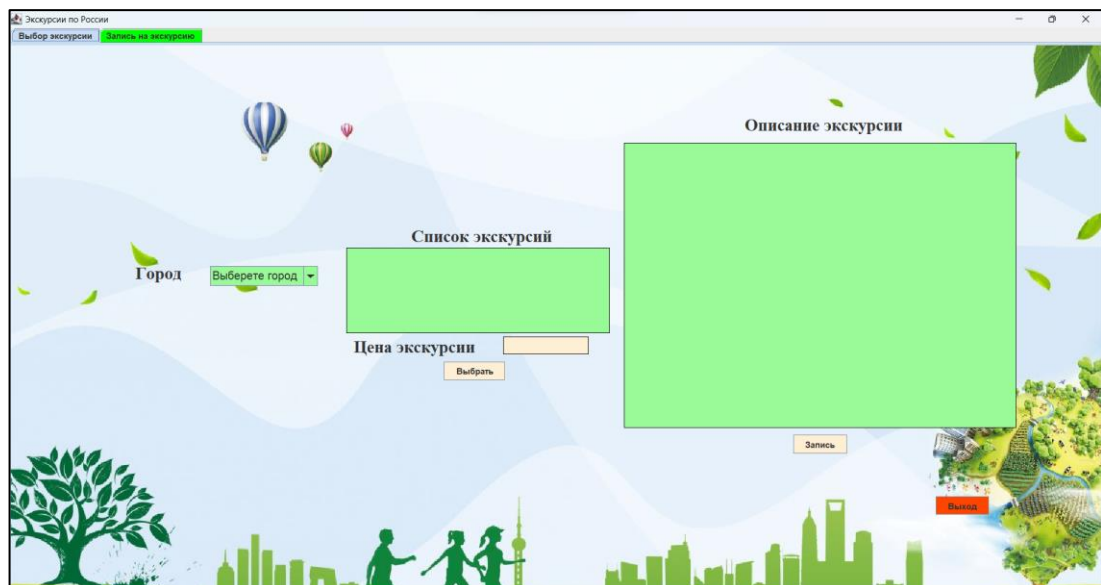


Рисунок 2 – Главное окно при открытии

Отображение списка «Выберите город» представлено на рисунке 3.



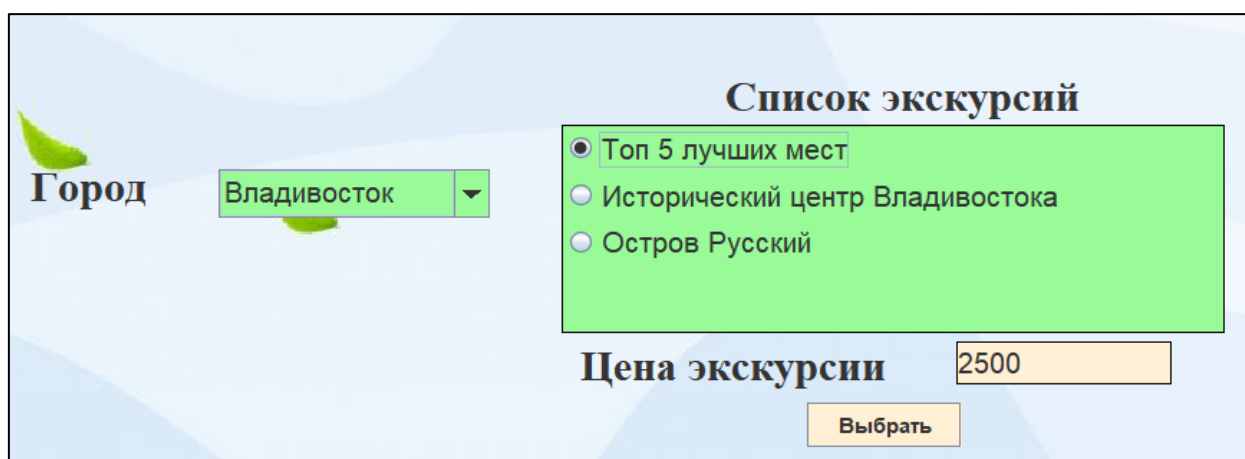
Город

Выберете город ▼

- Выберете город
- Владивосток
- Владимир
- Ростов Великий
- Санкт-Петербург
- Москва
- Ярославль

Рисунок 3 – Отображение списка

Отображение списка экскурсий при выборе разных городов представлено на рисунках 4,5.



Город: Владивосток ▼

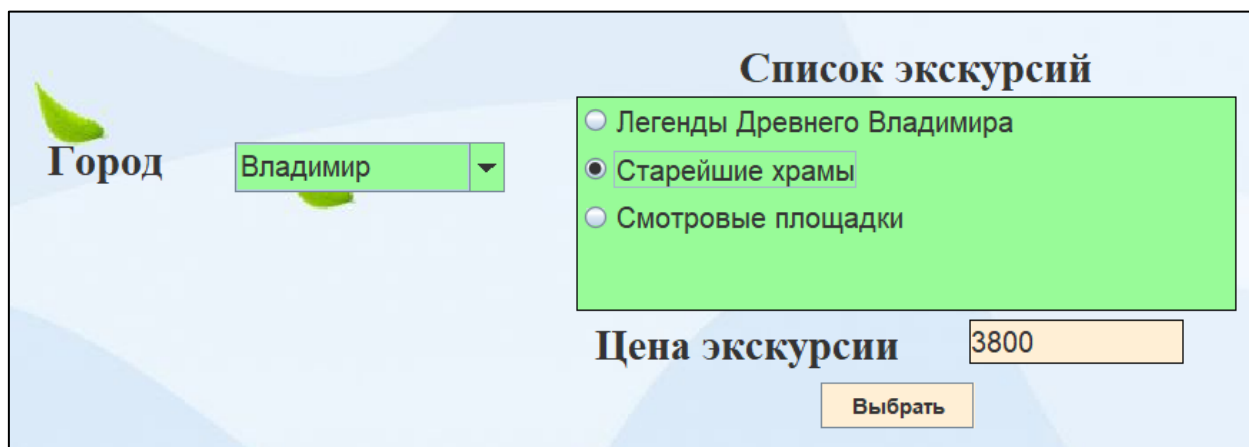
**Список экскурсий**

- ☒ **Топ 5 лучших мест**
- ☐ Исторический центр Владивостока
- ☐ Остров Русский

**Цена экскурсии** 2500

Выбрать

Рисунок 4 – Отображение списка экскурсий при выборе города  
«Владивосток»



Город: Владимир ▼

**Список экскурсий**

- ☐ Легенды Древнего Владимира
- ☒ **Старейшие храмы**
- ☐ Смотровые площадки

**Цена экскурсии** 3800

Выбрать

Рисунок 5 – Отображение списка экскурсий при выборе города «Владимир»

Как видно из рисунков 4,5 при выборе определенной экскурсии в поле «Цена экскурсии» появляется соответствующая стоимость экскурсии.

Отображение описания выбранной экскурсии появится при нажатии кнопки «Выбрать». Результат представлен на рисунке 6.

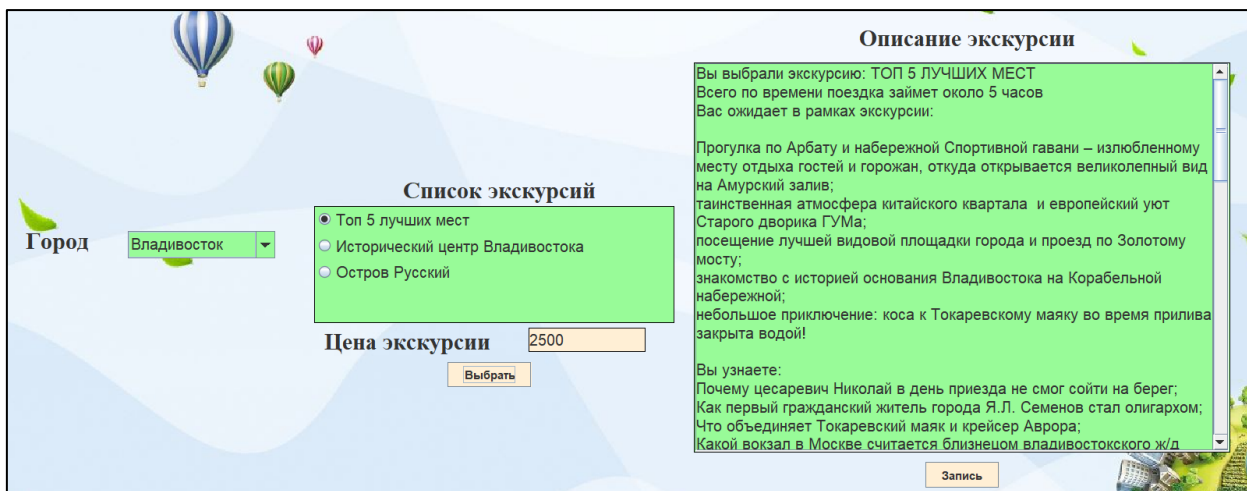


Рисунок 6 – Описание экскурсии

При нажатии кнопки «Запись» открывается вкладка «Запись на экскурсию». Результат представлен на рисунке 7.

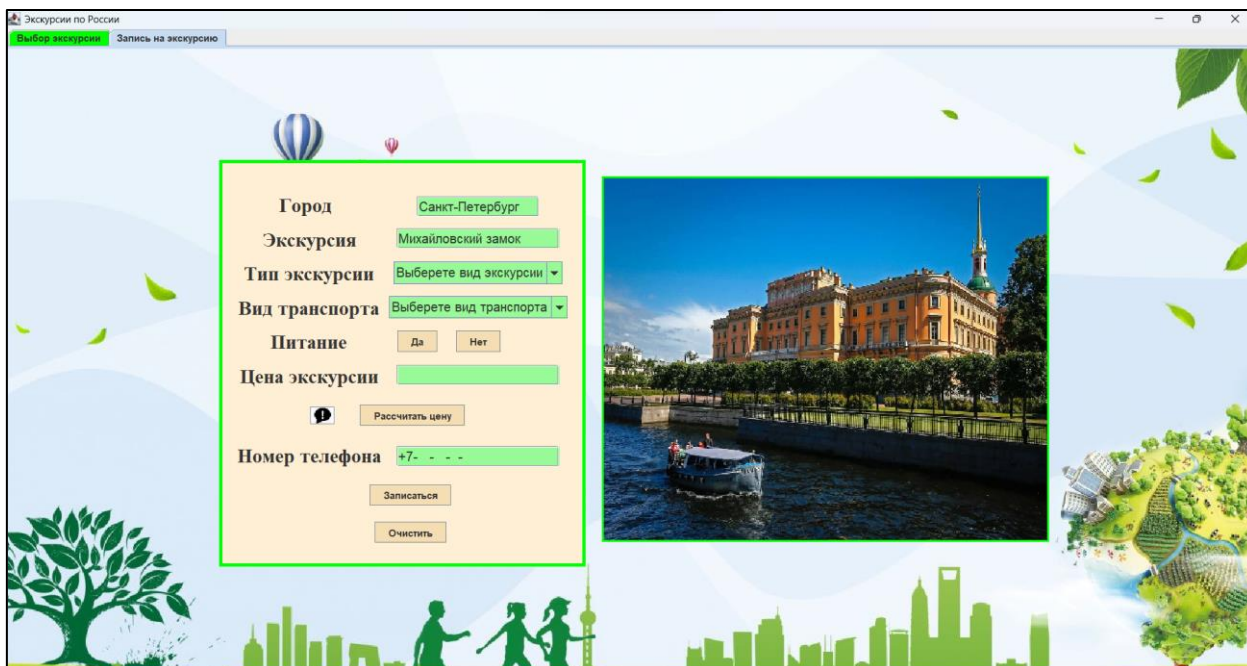


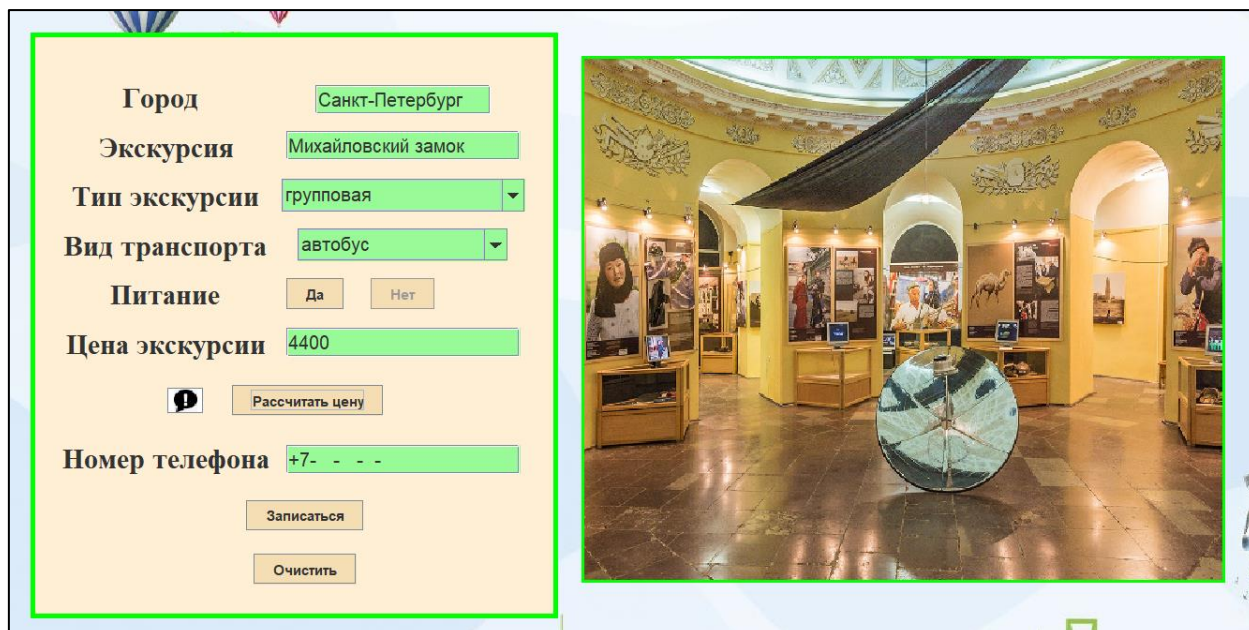
Рисунок 7 – Вкладка «Запись на экскурсию»

Как видно из рисунка 7 поля «Город» и «Экскурсия» заполнены, поскольку пользователь выбрал данный город и экскурсию во вкладке «Выбор экскурсии».

Отображение рассчитанной цены появляется, когда пользователь нажмет кнопку «Рассчитать цену», указав «Тип экскурсии», «Вид



транспорта» и включив/исключив «Питание». Пример рассчитанной цены представлен на рисунке 8.



The image shows a web form for booking a tour on the left and a photograph of a museum interior on the right. The form fields are as follows:

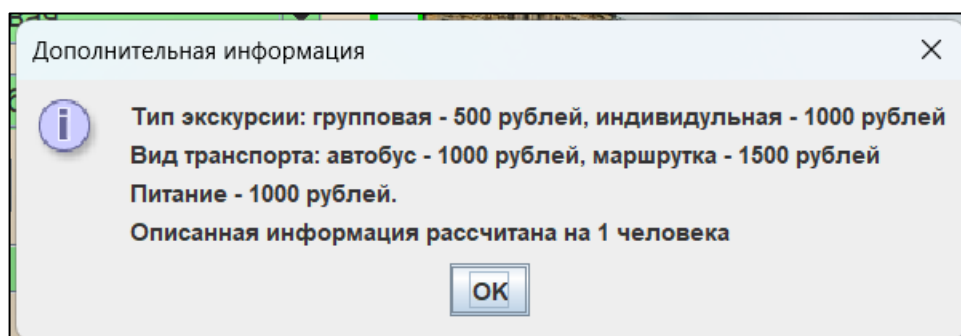
Field	Value
Город	Санкт-Петербург
Экскурсия	Михайловский замок
Тип экскурсии	групповая
Вид транспорта	автобус
Питание	Да
Цена экскурсии	4400
Номер телефона	+7- - -

Buttons: Рассчитать цену, Записаться, Очистить.

The photograph on the right shows a museum hall with yellow walls, arched doorways, and various exhibits, including a large circular object in the center.

Рисунок 8 – Цена экскурсии

Отображение дополнительной информации по ценам в диалоговом окне можно получить, нажав на кнопку с восклицательным знаком. Отображение дополнительной информации представлено на рисунке 9.



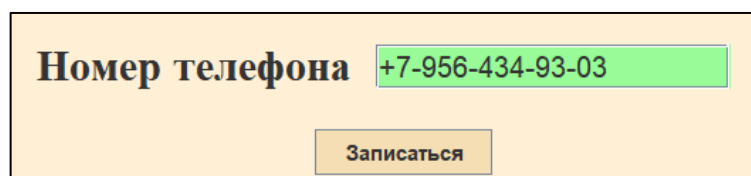
The dialog box contains the following text:

Тип экскурсии: групповая - 500 рублей, индивидуальная - 1000 рублей  
Вид транспорта: автобус - 1000 рублей, маршрутка - 1500 рублей  
Питание - 1000 рублей.  
Описанная информация рассчитана на 1 человека

Button: OK

Рисунок 9 – Диалоговое окно «Дополнительная информация»

Отображение ввода номера телефона представлено на рисунке 10.



The form shows the label "Номер телефона" and a text input field containing the number "+7-956-434-93-03". Below the input field is a button labeled "Записаться".

Рисунок 10 – Отображение ввода номера телефона

Отображение диалогового окна с проверкой введенных данных с целью записи на экскурсию появляется, нажав кнопку «Записаться». Результат представлен на рисунке 11.

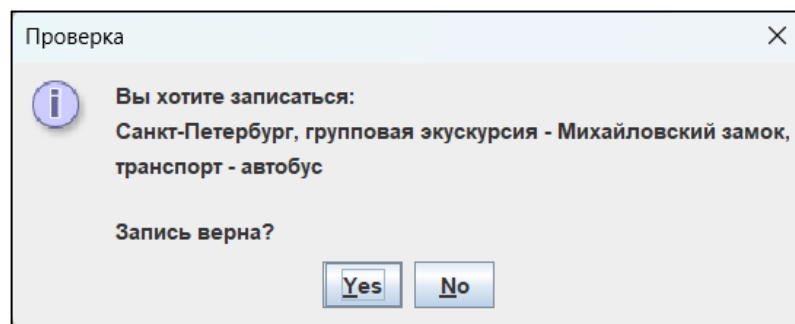


Рисунок 11 – Диалоговое окно «Проверка»

При нажатии кнопки «Yes» (рис.11) появляется диалоговое окно с подтверждением записи и уведомлении о звонке по указанному номеру телефона. Результат представлен на рисунке 12.

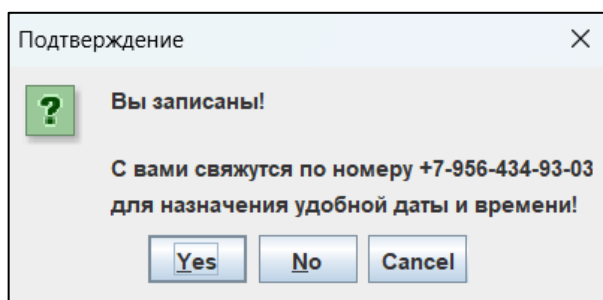


Рисунок 11 – Диалоговое окно «Подтверждение»

При нажатии кнопки «No» (рис.11) появляется диалоговое окно с соответствующим сообщением. Результат представлен на рисунке 13.

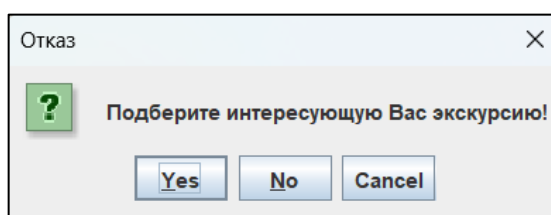


Рисунок 13 – Диалоговое окно «Отказ»

Отображение очистки всех полей происходит, когда пользователь нажимает кнопку «Очистить». Результат представлен на рисунках 14,15.

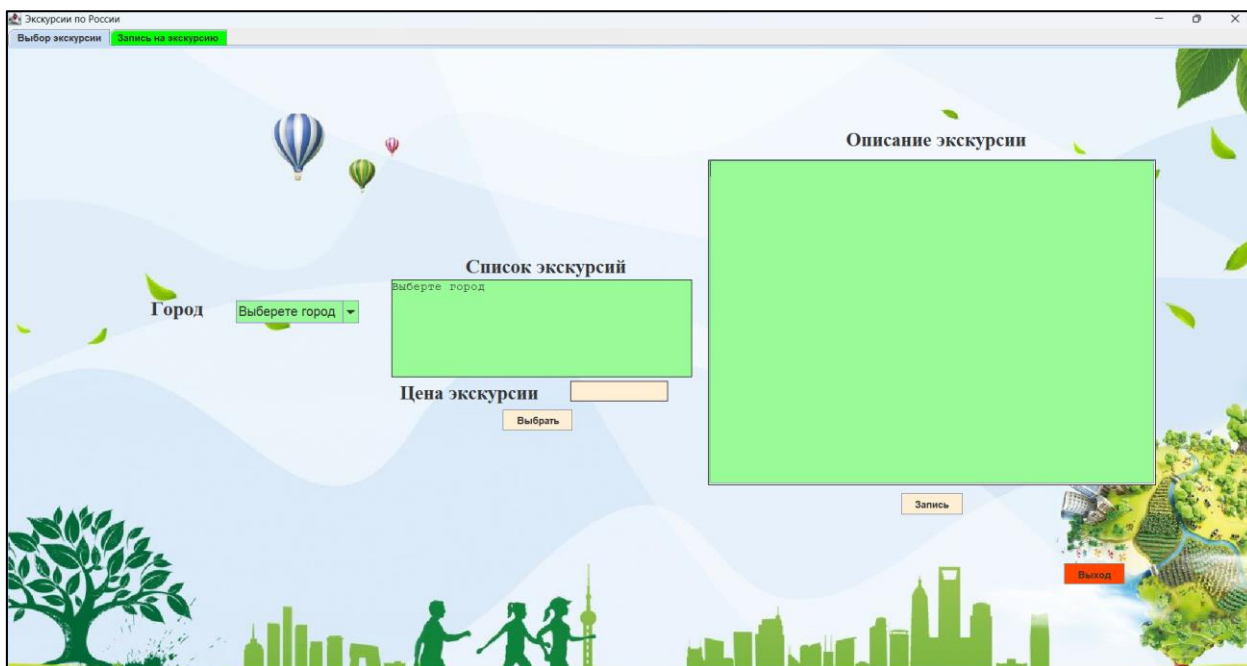


Рисунок 14 – Очистка полей на вкладке «Выбор экскурсии»



Рисунок 15 – Очистка полей на вкладке «Запись на экскурсию»



## Заключение

Разработка приложения для выбора экскурсий на языке программирования Java с использованием встроенной библиотеки Swing представляет собой важный этап в сфере туризма, обеспечивая удобство и функциональность для туристов и гидов.

Разработанное приложение состоит из двух этапов: выбора экскурсии и записи на нее. Чтобы визуально не загромождать пространство большим количеством компонентов на окне, были созданы две вкладки: выбор экскурсии и запись на экскурсию.

В данной работе было продемонстрировано создание, внедрение и использование различных компонентов, а также продемонстрирована связь между ними. Были показаны различные варианты выбора города, экскурсии, типа экскурсии, вида транспорта, что отразило работоспособность программы. Также были описаны и отображены работа каждого компонента приложения.

Исследуя сферу туризма и изучая язык программирования Java Swing, всегда можно усовершенствовать приложение путем добавления новых функциональных возможностей, улучшения визуального оформления и адаптирования под изменяющиеся требования туристической индустрии.

### Список литературы

1. Кетько, Н.В. Проектирование информационных систем: лекции/ ВолгГТУ. – 2024.
2. Давыдовский, М.А. Проектирование программной системы в UML Designer: учебное пособие / М. А. Давыдовский, М. Н. Никольская. – Москва: РУТ (МИИТ), 2019. – 129 с.
3. Флегонтов, А. В. Моделирование информационных систем: учебное пособие / А.В.Флегонтов, И. Ю. Матюшичев. – 2-е издание, стер.
4. Остроух А.В., Суркова Н.Е. Проектирование информационных систем: учебное пособие / А.В. Остроух, Н.Е Суркова. – 2-е издание, стер. – 2021. – 164 с.
5. Барков, И.А. Объектно-ориентированное программирование: учебник / И.А. Барков. – Санкт-Петербург: Лань, 2019. – 700с.

## Приложение А

```
package laba_swing;
import javax.swing.*;
import javax.swing.text.MaskFormatter;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.text.ParseException;

public class kurswork extends JFrame {
    final String logo = "D:\\фото\\cart3.jpg";
    ImageIcon background = new ImageIcon(logo);
    ImageIcon background1 = new ImageIcon("D:\\фото\\cart3.jpg");
    JLabel allLabel=new JLabel(background);
    JTabbedPane tappane=new JTabbedPane(JTabbedPane.TOP,
JTabbedPane.SCROLL_TAB_LAYOUT);
    JLabel shapLabel = new JLabel(background1);
    JPanel mainLabel = new JPanel();
    JLabel typLabel = new JLabel();
    JLabel cenLabel = new JLabel();
    JTextField cenTextF=new JTextField();
    JLabel typLabel1 = new JLabel();
    JTextField typTextF=new JTextField();
    JLabel numLabel = new JLabel();
    JLabel eksLabel = new JLabel();
    JTextField eksTextF=new JTextField();
    JLabel typEksLabel = new JLabel();
    JLabel vidTrLabel = new JLabel();
    JLabel pitLabel = new JLabel();
    JButton clearButton=new JButton();
    JButton yesButton=new JButton();
    JButton noButton=new JButton();
    JButton dopButton=new JButton(new ImageIcon("D:\\фото\\knop.png"));
    JButton zapButton=new JButton();
    JLabel sumLabel = new JLabel();
    JTextField sumTextF=new JTextField();
    Font myFont=new Font("Serif",Font.BOLD,24);
    Font myFont1=new Font("Arial",Font.PLAIN,16);
    JButton computeButton=new JButton();
    ButtonGroup factorButtonGroup=new ButtonGroup();
    JRadioButton[] factorRadioButton=new JRadioButton[4];
    JButton zpButton=new JButton();
    JButton rezButton=new JButton();
    JLabel uslLabel=new JLabel();
    JTextArea uslTA=new JTextArea();
    JLabel analysisLabel=new JLabel();
    JTextArea textField=new JTextArea();
    JButton exitBut=new JButton();
    JButton exit1But=new JButton();
    JLabel zpLabel=new JLabel();
    JTextField zpTextF=new JTextField();
    JFrame jf=new JFrame("Экскурсии по России");
    JPanel analysisTA=new JPanel();
    JLabel imageLabel= new JLabel();
    public Timer timer;
    public String[] images;
    private int currentIndex;
    String[] items = {
        "Выберете город",
        "Владивосток",
        "Владимир",
    }
```

```

        "Ростов Великий",
        "Санкт-Петербург",
        "Москва",
        "Ярославль"
    };
    String[] items1 = {
        "Выберете вид экскурсии",
        "групповая",
        "индивидуальная"
    };
    String[] items2 = {
        "Выберете вид транспорта",
        "автобус",
        "маршрутка"
    };
    JComboBox comboBox = new JComboBox(items);
    JComboBox comboBox1 = new JComboBox(items1);
    JComboBox comboBox2 = new JComboBox(items2);
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new kurswork();
            }
        });
    }
    public final Color[] colors = {Color.GREEN, Color.blue };
    public kurswork() {

        jf.getContentPane().setLayout(new BorderLayout());
        jf.setPreferredSize(new Dimension(1450, 650));
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        allLabel.setLayout(new GridBagLayout());
        allLabel.setOpaque(true);
        shapLabel.setLayout(new GridBagLayout());
        shapLabel.setOpaque(true);
        GridBagConstraints gridCons;

        mainLabel.setFont(myFont);
        mainLabel.setOpaque(true);
        mainLabel.setLayout(new GridBagLayout());
        mainLabel.setPreferredSize(new Dimension(450, 500));
        mainLabel.setBorder(BorderFactory.createLineBorder(Color.GREEN, 5));
        mainLabel.setBackground(Color.decode("#FFEFD5"));
        gridCons=new GridBagConstraints();
        gridCons.anchor=GridBagConstraints.CENTER;
        gridCons.insets=new Insets(10, 10, 0, 20);
        shapLabel.add(mainLabel, gridCons);

        typLabel1.setText("Город");
        typLabel1.setFont(myFont);
        gridCons=new GridBagConstraints();
        gridCons.gridx=0;
        gridCons.gridy=0;
        gridCons.anchor=GridBagConstraints.CENTER;
        gridCons.insets=new Insets(10, 0, 0, 0);
        mainLabel.add(typLabel1, gridCons);

        typTextF.setPreferredSize(new Dimension(150, 25));
        typTextF.setFont(myFont1);
        typTextF.setBackground(Color.decode("#98FB98"));
        gridCons=new GridBagConstraints();
        gridCons.gridx=1;
        gridCons.gridy=0;
    }

```

```

gridCons.insets=new Insets(15,10,0,10);
mainLabel.add(typTextF,gridCons);

eksLabel.setText("Экскурсия");
eksLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=0;
gridCons.gridy=1;
gridCons.anchor=GridBagConstraints.CENTER;
gridCons.insets=new Insets(10,10,0,0);
mainLabel.add(eksLabel,gridCons);
eksTextF.setPreferredSize(new Dimension(200,25));
eksTextF.setFont(myFont1);
eksTextF.setBackground(Color.decode("#98FB98"));
gridCons=new GridBagConstraints();
gridCons.gridx=1;
gridCons.gridy=1;
gridCons.insets=new Insets(10,10,0,10);
mainLabel.add(eksTextF,gridCons);

typEksLabel.setText("Тип экскурсии");
typEksLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=0;
gridCons.gridy=2;
gridCons.anchor=GridBagConstraints.CENTER;
gridCons.insets=new Insets(10,10,0,0);
mainLabel.add(typEksLabel,gridCons);

comboBox1.setEditable(true);
comboBox1.setFont(myFont1);
comboBox1.setEditable(false);
comboBox1.setBackground(Color.decode("#98FB98"));
gridCons=new GridBagConstraints();
gridCons.gridx=1;
gridCons.gridy=2;
gridCons.insets=new Insets(10,10,0,10);
mainLabel.add(comboBox1,gridCons);

comboBox1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        vybor1ActionPerformed(e);
    }
    private void vybor1ActionPerformed(ActionEvent e) {
        if ((String)comboBox1.getSelectedItem()=="индивидуальная") {
            comboBox2.removeAllItems();
            comboBox2.addItem("маршрутка");
        }else {
            comboBox2.removeAllItems();
            comboBox2.addItem("Выберете транспорт");
            comboBox2.addItem("автобус");
            comboBox2.addItem("маршрутка");
        }
        sumTextF.setText("");
        noButton.setEnabled(true);
        yesButton.setEnabled(true);
    }
});
vidTrLabel.setText("Вид транспорта");
vidTrLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=0;

```

```

gridCons.gridx=3;
gridCons.anchor=GridBagConstraints.CENTER;
gridCons.insets=new Insets(10,10,0,0);
mainLabel.add(vyborTrLabel,gridCons);

comboBox2.setEditable(true);
comboBox2.setFont(myFont1);
comboBox2.setEditable(false);
comboBox2.setBackground(Color.decode("#98FB98"));
gridCons=new GridBagConstraints();
gridCons.gridx=1;
gridCons.gridy=3;
gridCons.insets=new Insets(10,10,0,10);
mainLabel.add(comboBox2,gridCons);
comboBox2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        vybor2ActionPerformed(e);
    }
    private void vybor2ActionPerformed(ActionEvent e) {
        noButton.setEnabled(true);
        yesButton.setEnabled(true);
        sumTextF.setText("");
    }
});

pitLabel.setText("Питание");
pitLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=0;
gridCons.gridy=4;
gridCons.anchor=GridBagConstraints.CENTER;
gridCons.insets=new Insets(10,10,0,0);
mainLabel.add(pitLabel,gridCons);

dopButton.setText("");
dopButton.setOpaque(true);
dopButton.setPreferredSize(new Dimension(30,22));
dopButton.setBackground(Color.WHITE);
gridCons=new GridBagConstraints();
gridCons.gridx=0;
gridCons.gridy=6;
gridCons.anchor=GridBagConstraints.WEST;
gridCons.insets=new Insets(20,100,0,0);
mainLabel.add(dopButton,gridCons);
dopButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dopButtonActionPerformed(e);
    }
    private void dopButtonActionPerformed(ActionEvent e) {
        JOptionPane.showConfirmDialog(null,
            "<html><center>Тип экскурсии: групповая - 500 рублей,  

индивидуальная - 1000 рублей\n</center></html>" +
            "<html><center>Вид транспорта: автобус - 1000  

рублей, маршрутка - 1500 рублей\n</center></html>" +
            "<html><center>Питание - 1000  

рублей.\n\n</center></html>" +
            "<html><center>Описанная информация  

рассчитана на 1 человека</center></html>",
            "Дополнительная информация",
            JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE);
    }
});

```

```

yesButton.setText("Да");
yesButton.setOpaque(true);
yesButton.setBackground(Color.decode("#F5DEB3"));
gridCons=new GridBagConstraints();
gridCons.gridx=1;
gridCons.gridy=4;
gridCons.gridwidth=2;
gridCons.insets=new Insets(10,0,0,150);
mainLabel.add(yesButton,gridCons);
yesButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        yesButtonActionPerformed(e);
    }
    private void yesButtonActionPerformed(ActionEvent e) {
        noButton.setEnabled(false);
    }
});
noButton.setText("Нет");
noButton.setBackground(Color.decode("#F5DEB3"));
gridCons=new GridBagConstraints();
gridCons.gridx=1;
gridCons.gridy=4;
gridCons.gridwidth=2;
gridCons.insets=new Insets(10,0,0,0);
mainLabel.add(noButton,gridCons);
noButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        noActionPerformed(e);
    }
    private void noActionPerformed(ActionEvent e) {
        yesButton.setEnabled(false);
    }
});

sumLabel.setText("Цена экскурсии");
sumLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=0;
gridCons.gridy=5;
gridCons.anchor=GridBagConstraints.CENTER;
gridCons.insets=new Insets(10,10,0,0);
mainLabel.add(sumLabel,gridCons);

sumTextF.setPreferredSize(new Dimension(200,25));
sumTextF.setFont(myFont1);
sumTextF.setBackground(Color.decode("#98FB98"));
gridCons=new GridBagConstraints();
gridCons.gridx=1;
gridCons.gridy=5;
gridCons.insets=new Insets(10,10,0,10);
mainLabel.add(sumTextF,gridCons);

computeButton.setText("Рассчитать цену");
computeButton.setBackground(Color.decode("#F5DEB3"));
gridCons=new GridBagConstraints();
gridCons.gridx=0;
gridCons.gridy=6;
gridCons.gridwidth=2;
gridCons.anchor=GridBagConstraints.CENTER;
gridCons.insets=new Insets(20,25,0,0);
mainLabel.add(computeButton,gridCons);

computeButton.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            computeButtonActionPerformed(e);
        }
    });

    numLabel.setText("Номер телефона");
    numLabel.setFont(myFont);
    gridCons=new GridBagConstraints();
    gridCons.gridx=0;
    gridCons.gridy=7;
    gridCons.anchor=GridBagConstraints.CENTER;
    gridCons.insets=new Insets(20,10,0,0);
    mainLabel.add(numLabel,gridCons);

    try {
        MaskFormatter maskFormatter = new MaskFormatter("+7-###-###-##-
##");
        JFormattedTextField numTextF = new
JFormattedTextField(maskFormatter);
        numTextF.setPreferredSize(new Dimension(200,25));
        numTextF.setFont(myFont1);
        numTextF.setBackground(Color.decode("#98FB98"));
        gridCons=new GridBagConstraints();
        gridCons.gridx=1;
        gridCons.gridy=7;
        gridCons.insets=new Insets(23,10,0,10);
        mainLabel.add(numTextF,gridCons);

        zapButton.setText("Записаться");
        zapButton.setBackground(Color.decode("#F5DEB3"));
        gridCons=new GridBagConstraints();
        gridCons.gridx=0;
        gridCons.gridy=8;
        gridCons.gridwidth=2;
        gridCons.anchor=GridBagConstraints.CENTER;
        gridCons.insets=new Insets(20,20,0,0);
        mainLabel.add(zapButton,gridCons);

        zapButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                zapButtonActionPerformed(e);
            }
            private void zapButtonActionPerformed(ActionEvent e) {
                int i=JOptionPane.showConfirmDialog(null,
                    "Вы хотите записаться:\n"+typTextF.getText()+"",
                    "+comboBox1.getSelectedItem()+" экскурсия - "
                    +eksTextF.getText()+"\n" +
                    "транспорт -
                    "+comboBox2.getSelectedItem()+"\n\nЗапись верна?",
                    "Проверка", JOptionPane.YES_NO_OPTION,
                    JOptionPane.INFORMATION_MESSAGE);
                if (i==JOptionPane.YES_OPTION) {
                    JOptionPane.showConfirmDialog(null,"Вы записаны!\n\n" +
                        "С вами свяжутся по номеру
                        "+numTextF.getText()+"\nдля назначения удобной даты и
                        времени!", "Подтверждение",JOptionPane.INFORMATION_MESSAGE);
                }else if (i==JOptionPane.NO_OPTION) {
                    JOptionPane.showConfirmDialog(null,"Подберите
                    интересующую Вас экскурсию!\n", "Отказ",JOptionPane.INFORMATION_MESSAGE);
                }
            }
        });
    }
}

```



```

clearButton.setText("ОЧИСТИТЬ");
clearButton.setBackground(Color.decode("#F5DEB3"));
gridCons=new GridBagConstraints();
gridCons.gridx=0;
gridCons.gridy=9;
gridCons.gridwidth=2;
gridCons.anchor=GridBagConstraints.CENTER;
gridCons.insets=new Insets(20,20,0,0);
mainLabel.add(clearButton,gridCons);

clearButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        clearButtonActionPerformed(e);
    }
    private void clearButtonActionPerformed(ActionEvent e) {
        comboBox1.setSelectedItem("Выберете вид экскурсии");
        yesButton.setEnabled(true);
        noButton.setEnabled(true);
        cenTextF.setText("");
        numTextF.setText("");
        eksTextF.setText("");
        typTextF.setText("");
        comboBox.setSelectedItem("Выберете город");
        uslTA.setText("Выберете город");
        analysisTA.setBackground(Color.decode("#98FB98"));
        textField.setBackground(Color.decode("#98FB98"));
        textField.setText("");
        tappane.setSelectedIndex(0);
        if (clearButton.getText()=="ОЧИСТИТЬ") {
            images = new String[]{"D:\\фото\\vopr.jpg"};
            setCurrentImage();
            timer = new Timer(2000, new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    currentIndex = (currentIndex + 1) %
images.length;
                    setCurrentImage(); // Обновляем изображение
                }
            });
            timer.start();
        }
    });
} catch (ParseException exc){
}
typLabel.setText("Город");
typLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=0;
gridCons.gridy=2;
gridCons.anchor=GridBagConstraints.WEST;
gridCons.insets=new Insets(-25,65,0,0);
allLabel.add(typLabel,gridCons);

cenLabel.setText("Цена экскурсии");
cenLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=3;
gridCons.gridy=2;
gridCons.anchor=GridBagConstraints.WEST;
gridCons.insets=new Insets(180,20,0,0);
allLabel.add(cenLabel,gridCons);

```

```

cenTextF.setPreferredSize(new Dimension(120,25));
cenTextF.setEditable(false);
cenTextF.setFont(myFont1);
cenTextF.setBackground(Color.decode("#FFefd5"));
cenTextF.setBorder(BorderFactory.createLineBorder(Color.BLACK));
gridCons=new GridBagConstraints();
gridCons.gridx=3;
gridCons.gridy=2;
gridCons.insets=new Insets(180,190,0,0);
allLabel.add(cenTextF,gridCons);

comboBox.setFont(myFont1);
comboBox.setBackground(Color.decode("#98fb98"));
gridCons=new GridBagConstraints();
gridCons.gridx=1;
gridCons.gridy=2;
gridCons.insets=new Insets(-15,40,0,30);
allLabel.add(comboBox,gridCons);
comboBox.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        wyborActionPerformed(e);
    }
});

zpButton.setText("Выбрать");
zpButton.setBackground(Color.decode("#FFefd5"));
zpLabel.setEnabled(false);
zpTextF.setEnabled(false);
gridCons=new GridBagConstraints();
gridCons.gridx=3;
gridCons.gridy=2;
gridCons.insets=new Insets(290,0,40,10);
allLabel.add(zpButton,gridCons);
zpButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        vubrtorButtonActionPerformed(e);
    }
});

rezButton.setText("Запись");
rezButton.setBackground(Color.decode("#FFefd5"));
zpLabel.setEnabled(false);
zpTextF.setEnabled(false);
gridCons=new GridBagConstraints();
gridCons.gridx=4;
gridCons.gridy=6;
gridCons.insets=new Insets(10,0,50,0);
allLabel.add(rezButton,gridCons);
rezButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        rezButtonActionPerformed(e);
    }
});

uslLabel.setText("Список экскурсий");
uslLabel.setFont(myFont);
gridCons=new GridBagConstraints();
gridCons.gridx=3;
gridCons.gridy=2;
gridCons.anchor=GridBagConstraints.CENTER;
gridCons.insets=new Insets(10,10,140,0);
allLabel.add(uslLabel,gridCons);

```

```

        uslTA.setPreferredSize(new Dimension(370,120));
        uslTA.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        uslTA.setFont(new Font("Courier New",Font.PLAIN,14));
        uslTA.setBackground(Color.decode("#98FB98"));
        gridCons=new GridBagConstraints();
        gridCons.gridx=3;
        gridCons.gridy=2;
        gridCons.gridheight=4;
        gridCons.anchor=GridBagConstraints.WEST;
        gridCons.insets=new Insets(25,10,0,10);
        allLabel.add(uslTA,gridCons);

        analysisLabel.setText("Описание экскурсии");
        analysisLabel.setFont(myFont);
        gridCons=new GridBagConstraints();
        gridCons.gridx=4;
        gridCons.gridy=1;
        gridCons.anchor=GridBagConstraints.CENTER;
        gridCons.insets=new Insets(0,10,0,0);
        allLabel.add(analysisLabel,gridCons);

        analysisTA.setPreferredSize(new Dimension(550,400));
        analysisTA.setLayout(new BorderLayout());
        analysisTA.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        analysisTA.setFont(new Font("Courier New",Font.PLAIN,14));
        analysisTA.setBackground(Color.decode("#98FB98"));
        gridCons=new GridBagConstraints();
        gridCons.gridx=4;
        gridCons.gridy=2;
        gridCons.anchor=GridBagConstraints.WEST;
        gridCons.insets=new Insets(10,10,0,10);
        allLabel.add(analysisTA,gridCons);
        textField.setBackground(Color.decode("#98FB98"));

        exitBut.setText("Выход");
        exitBut.setBackground(Color.decode("#FF4500"));
        gridCons=new GridBagConstraints();
        gridCons.gridx=4;
        gridCons.gridy=8;
        gridCons.insets=new Insets(10,400,10,0);
        allLabel.add(exitBut,gridCons);
        exitBut.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                exitButActionPerformed(e);
            }
        });
        imageLabel.setPreferredSize(new Dimension(550,450));
        gridCons=new GridBagConstraints();
        gridCons.anchor=GridBagConstraints.CENTER;
        shapLabel.add(imageLabel,gridCons);

        for (int i = 1; i < colors.length; i++) {
            tappane.setBackground(colors[i - 1]);
            // Добавление вкладки
            tappane.addTab(String.format("Выбор экскурсии", i), allLabel);
            tappane.addTab(String.format("Запись на экскурсию", i),
shapLabel);
            // Подключение мнемоники
            tappane.setMnemonicAt(i-1, String.valueOf(i).charAt(0));
        }
        tappane.addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent e) {
                // Определяем индекс выделенной мышкой вкладки

```

```

        int idx =
((JTabbedPane)e.getSource()).indexOfLocation(e.getX(), e.getY());
        System.out.println("Выбрана вкладка " + idx);
    }
});
jf.add(tappane);
jf.pack();
jf.setVisible(true);
}
private void setCurrentImage() {
    ImageIcon imageIcon = new ImageIcon((images[currentIndex]));
    Image scaledImage =
imageIcon.getImage().getScaledInstance(imageLabel.getWidth(),
imageLabel.getHeight(), Image.SCALE_DEFAULT);
    imageLabel.setIcon(new ImageIcon(scaledImage));
}
public void vyborActionPerformed(ActionEvent e) {
    int x=0;
    int y=0;
    GridBagConstraints gridCons;
    uslTA.removeAll();
    uslTA.setLayout(new BoxLayout(uslTA, BoxLayout.Y_AXIS));
    for (int i = 0; i < 3; i++) {
        factorRadioButton[i] = new JRadioButton();
        factorRadioButton[i].setBackground(Color.decode("#98FB98"));
        factorButtonGroup.add(factorRadioButton[i]);
        factorRadioButton[i].setFont(myFont1);
        gridCons = new GridBagConstraints();
        if (i < 2) {
            gridCons.gridx = x;
            gridCons.gridy = y;
        } else {
            gridCons.gridx = 0;
            gridCons.gridy = 0;
            gridCons.gridwidth = 2;
        }
        if (x > 0) {
            x = 0;
            y++;
        }
        x++;
        if ((String)comboBox.getSelectedItem()=="Выберете город") {
            uslTA.setText("Выберете город!");
        } else uslTA.setText("");
        if ((String)comboBox.getSelectedItem()=="Владивосток") {
            if (i == 0) factorRadioButton[i].setText("Топ 5 лучших
мест");
            if (i == 1) factorRadioButton[i].setText("Исторический
центр Владивостока");
            if (i == 2) factorRadioButton[i].setText("Остров
Русский");
            uslTA.add(factorRadioButton[i], gridCons);
        }
        if ((String)comboBox.getSelectedItem()=="Владимир") {
            if (i == 0) factorRadioButton[i].setText("Легенды
Древнего Владимира");
            if (i == 1) factorRadioButton[i].setText("Старейшие
храмы");
            if (i == 2) factorRadioButton[i].setText("Смотровые
площадки");
            uslTA.add(factorRadioButton[i], gridCons);
        }
        if ((String)comboBox.getSelectedItem()=="Ростов Великий") {

```

```

        if (i == 0) factorRadioButton[i].setText("Святые места");
        if (i == 1) factorRadioButton[i].setText("Озеро Неро");
        if (i == 2) factorRadioButton[i].setText("Семибратово");
        uslTA.add(factorRadioButton[i], gridCons);
    }
    if ((String)comboBox.getSelectedItem()=="Санкт-Петербург") {
        if (i == 0) factorRadioButton[i].setText("Михайловский
замок");
        if (i == 1) factorRadioButton[i].setText("Обзорная
экскурсия с посещением Кунсткамеры");
        if (i == 2) factorRadioButton[i].setText("Мифы и легенды
города");
        uslTA.add(factorRadioButton[i], gridCons);
    }
    if ((String)comboBox.getSelectedItem()=="Москва") {
        if (i == 0) factorRadioButton[i].setText("Центр Москвы");
        if (i == 1) factorRadioButton[i].setText("Воробьевы
горы");
        if (i == 2) factorRadioButton[i].setText("Особняк
Рябушинского");
        uslTA.add(factorRadioButton[i], gridCons);
    }
    if ((String)comboBox.getSelectedItem()=="Ярославль") {
        if (i == 0) factorRadioButton[i].setText("Исторический
центр Ярославля");
        if (i == 1) factorRadioButton[i].setText("Монастыри
Ярославля");
        if (i == 2) factorRadioButton[i].setText("Село Вятское");
        uslTA.add(factorRadioButton[i], gridCons);
    }
    gridCons.anchor = GridBagConstraints.WEST;
    factorRadioButton[i].addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            factorRadioButtonActionPerformed(e);
        }
        private void factorRadioButtonActionPerformed(ActionEvent
e) {

            if (comboBox.getSelectedItem()=="Владивосток") {
                cenTextF.setText("");
                if (factorRadioButton[0].isSelected())
cenTextF.setText("2500");
                if (factorRadioButton[1].isSelected())
cenTextF.setText("3200");
                if (factorRadioButton[2].isSelected())
cenTextF.setText("4500");
            }
            if (comboBox.getSelectedItem()=="Владимир") {
                cenTextF.setText("");
                if (factorRadioButton[0].isSelected())
cenTextF.setText("1500");
                if (factorRadioButton[1].isSelected())
cenTextF.setText("3800");
                if (factorRadioButton[2].isSelected())
cenTextF.setText("4000");
            }
            if (comboBox.getSelectedItem()=="Ростов Великий") {
                cenTextF.setText("");
                if (factorRadioButton[0].isSelected())
cenTextF.setText("2500");
                if (factorRadioButton[1].isSelected())
cenTextF.setText("3000");
            }
        }
    });
}

```

```

        if (factorRadioButton[2].isSelected())
cenTextF.setText("4200");
    }
    if (comboBox.getSelectedItem()=="Москва") {
cenTextF.setText("");
        if (factorRadioButton[0].isSelected())
cenTextF.setText("2800");
        if (factorRadioButton[1].isSelected())
cenTextF.setText("1500");
        if (factorRadioButton[2].isSelected())
cenTextF.setText("2800");
    }
    if (comboBox.getSelectedItem()=="Санкт-Петербург") {
cenTextF.setText("");
        if (factorRadioButton[0].isSelected())
cenTextF.setText("1900");
        if (factorRadioButton[1].isSelected())
cenTextF.setText("2300");
        if (factorRadioButton[2].isSelected())
cenTextF.setText("1200");
    }
    if (comboBox.getSelectedItem()=="Ярославль") {
cenTextF.setText("");
        if (factorRadioButton[0].isSelected())
cenTextF.setText("1350");
        if (factorRadioButton[1].isSelected())
cenTextF.setText("2000");
        if (factorRadioButton[2].isSelected())
cenTextF.setText("3800");
    }
    }
    });
    }
    uslTA.revalidate();
    uslTA.repaint();
}
private void exitButActionPerformed(ActionEvent e) {
    System.exit(0);
}
private void computeButtonActionPerformed(ActionEvent e) {
    int group=500, indiv=1000;
    int av=1000, mar=1500, eda=1000;
    int sum=0;
    String comboBox1Value = (String) comboBox1.getSelectedItem();
    String comboBox2Value = (String) comboBox2.getSelectedItem();
    String yesButtonText = yesButton.getText();
    String noButtonText = noButton.getText();
    int cen = Integer.parseInt(cenTextF.getText());
    if ("Рассчитать цену".equals(computeButton.getText())) {
        if ("групповая".equals(comboBox1Value)) {
            if ("автобус".equals(comboBox2Value) &&
"Да".equals(yesButtonText)) {
                sum = cen + group + av + eda;
            } else if ("автобус".equals(comboBox2Value) &&
"Нет".equals(noButtonText)) {
                sum = cen + group + av;
            } else if ("маршрутка".equals(comboBox2Value) &&
"Да".equals(yesButtonText)) {
                sum = cen + group + mar + eda;
            } else if ("маршрутка".equals(comboBox2Value) &&
"Нет".equals(noButtonText)) {
                sum = cen + group + mar;
            }
        }
    }
}

```

```

    } else if ("индивидуальная".equals(comboBox1Value)) {
        if ("Да".equals(yesButtonText)) {
            sum = cen + indiv + mar + eda;
        } else if ("Нет".equals(noButtonText)) {
            sum = cen + indiv + mar;
        }
    }
}
sumTextF.setText(Integer.toString(sum));
}
public void actionPerformed(ActionEvent ae){
}
private void vubrtorButtonActionPerformed(ActionEvent e) {
    int k;
    String s="";
    if ((String)comboBox.getSelectedItem()=="Владивосток"){
        for (int i = 0; i < 3; i++) {
            k=i;
            if (factorRadioButton[i].isSelected() && k==0){
                if (zpButton.getText().equals("Выбрать")) {
                    textField.setText("Вы выбрали экскурсию: "+
factorRadioButton[i].getText().toUpperCase() +
"\nВсего по времени поездки займет около 5
часов\n" +
"\nВас ожидает в рамках экскурсии:\n" + "\n" +
"Прогулка по Арбату и набережной Спортивной
гавани - излюбленному месту отдыха гостей и горожан, откуда открывается
великолепный вид на Амурский залив;\n" +
"таинственная атмосфера китайского квартала
и европейский уют Старого дворика ГУМа;\n" +
"посещение лучшей видовой площадки города и
проезд по Золотому мосту;\n" +
"знакомство с историей основания Владивостока
на Корабельной набережной;\n" +
"небольшое приключение: коса к Токаревскому
маяку во время прилива закрыта водой!\n\n" +
"Вы узнаете:\n" +
"Почему цесаревич Николай в день приезда не
смог сойти на берег;\n" +
"Как первый гражданский житель города Я.Л.
Семенов стал олигархом;\n" +
"Что объединяет Токаревский маяк и крейсер
Аврора;\n" +
"Какой вокзал в Москве считается близнецом
владивостокского ж/д вокзала;\n" +
"Как экипаж подводной лодки С-56 смог
продержаться на глубине 6 часов без вентиляции;\n" +
"Почему китайский квартал называется
Миллионка;\n" +
"И много других интересных фактов из жизни
Владивостока!\n\n"+
"МАРШРУТ ЭКСКУРСИИ\n" +
"\n" +
"Набережная Спортивной гавани ~ 45мин\n" +
"Исторически это место связано с проживанием
китайской диаспоры: сохранившиеся дворики до сих пор хранят загадочную
атмосферу Чайна-тауна. Сейчас это центральная часть города, поэтому фрагмент
улицы Адмирала Фокина превращен в Арбат - пешеходную прогулочную зону с
фонтанами, множеством кафе и магазинчиков. В наше внимание попадет и главная
улица города, Светланская, с великолепной архитектурой в стиле модерн.\n" +
"\n" +
"Токаревский маяк ~ 45мин\n" +
"Визитная карточка Владивостока и один из

```

старейших маяков Приморья. Единственный действующий маяк, территория которого доступна для посещения. С мыса Токаревская кошка открывается прекрасный вид на Амурский залив, пролив Босфор Восточный, остров Русский и Русский мост.\n" +

"\n" +  
"Корабельная набережная ~ 45мин\n" +  
"Место, откуда начался Владивосток. Сейчас здесь расположен Мемориальный комплекс Боевая слава Тихоокеанского флота с подводной лодкой-музеем С-56 и кораблем-музеем Красный выпел. Рядом находится 33-й причал с боевыми кораблями ТОФ. В соседнем Адмиральском сквере красуется восстановленная Триумфальная арка. Первоначально была построена в честь посещения города цесаревичем Николаем в 1891 году, в советское время была разрушена.\n" +

"\n" +  
"Проезд по Золотому мосту ~ 15мин\n" +  
"Золотой мост, получивший свое название из-за бухты Золотой Рог, входит в пятерку крупнейших вантовых мостов мира.\n" +  
"\n" +  
"Видовая площадка Орлиное гнездо ~ 30мин\n" +  
"Сопка Орлиное гнездо является самой высокой в исторической части города, поэтому взору открываются захватывающие дух виды. Бухта Золотой Рог и раскинувшийся на ее берегах город с причалами, портами, вокзалами и высотками. Русский остров, полуостров Шкота и полуостров Черкавского, Амурский залив и пролив Босфор Восточный – весь город как на ладони.\n" +

"\n" +  
"Старый дворик ГУМа ~ 30мин\n" +  
"Атмосферное место в самом центре города, возможно, кому-то напомним Европу. И неспроста, ведь это внутренний дворик бывшего немецкого торгового дома Кунст и Альберс. Множество арт-объектов позволяет еще больше узнать об истории города.");  
}

```
    }}else if ((String)comboBox.getSelectedItemAt(0)=="Владимир") {  
for (int i = 0; i < 3; i++) {  
    k=i;  
    if (factorRadioButton[i].isSelected() && k==0) {  
        if (zpButton.getText().equals("Выбрать")) {  
            textField.setText("Вы выбрали экскурсию: "+  
factorRadioButton[i].getText().toUpperCase() +  
"\nВсего по времени поездки займет около 5  
часов\n" +  
"Вас ожидает в рамках экскурсии:\n" + "\n" +  
"Прогулка по Арбату и набережной Спортивной  
тавани – излюбленному месту отдыха гостей и горожан, откуда открывается  
великолепный вид на Амурский залив;\n" +  
"таинственная атмосфера китайского квартала  
и европейский уют Старого дворика ГУМа;\n" +  
"посещение лучшей видовой площадки города и  
проезд по Золотому мосту;\n" +  
"знакомство с историей основания Владивостока  
на Корабельной набережной;\n" +  
"небольшое приключение: коса к Токаревскому  
маяку во время прилива закрыта водой!\n\n" +  
"Вы узнаете:\n" +  
"Почему цесаревич Николай в день приезда не  
смог сойти на берег;\n" +  
"Как первый гражданский житель города Я.Л.  
Семенов стал олигархом;\n" +  
"Что объединяет Токаревский маяк и крейсер  
Аврора;\n" +  
"Какой вокзал в Москве считается близнецом  
владивостокского ж/д вокзала;\n" +
```



```

        "Как экипаж подводной лодки С-56 смог
        продержаться на глубине 6 часов без вентиляции;\n" +
        "Почему китайский квартал называется
        Миллионка;\n" +
        "И много других интересных фактов из жизни
        Владивостока!\n\n"+
        "МАРШРУТ ЭКСКУРСИИ\n" +
        "\n" +
        "Набережная Спортивной гавани ~ 45мин\n" +
        "Исторически это место связано с проживанием
        китайской диаспоры: сохранившиеся дворики до сих пор хранят загадочную
        атмосферу Чайна-тауна. Сейчас это центральная часть города, поэтому фрагмент
        улицы Адмирала Фокина превращен в Арбат – пешеходную прогулочную зону с
        фонтанами, множеством кафе и магазинчиков. В наше внимание попадет и главная
        улица города, Светланская, с великолепной архитектурой в стиле модерн.\n" +
        "\n" +
        "Токаревский маяк ~ 45мин\n" +
        "Визитная карточка Владивостока и один из
        старейших маяков Приморья. Единственный действующий маяк, территория которого
        доступна для посещения. С мыса Токаревская кошка открывается прекрасный вид
        на Амурский залив, пролив Босфор Восточный, остров Русский и Русский мост.\n"
        +
        "\n" +
        "Корабельная набережная ~ 45мин\n" +
        "Место, откуда начался Владивосток. Сейчас
        здесь расположен Мемориальный комплекс Боевая слава Тихоокеанского флота с
        подводной лодкой-музеем С-56 и кораблем-музеем Красный выпел. Рядом
        находится 33-й причал с боевыми кораблями ТОФ. В соседнем Адмиральском сквере
        красуется восстановленная Триумфальная арка. Первоначально была построена в
        честь посещения города цесаревичем Николаем в 1891 году, в советское время
        была разрушена.\n" +
        "\n" +
        "Проезд по Золотому мосту ~ 15мин\n" +
        "Золотой мост, получивший свое название из-за
        бухты Золотой Рог, входит в пятерку крупнейших вантовых мостов мира.\n" +
        "\n" +
        "Видовая площадка Орлиное гнездо ~ 30мин\n" +
        "Сопка Орлиное гнездо является самой высокой
        в исторической части города, поэтому взору открываются захватывающие дух
        виды. Бухта Золотой Рог и раскинувшийся на ее берегах город с причалами,
        портами, вокзалами и высотками. Русский остров, полуостров Шкота и полуостров
        Черкавского, Амурский залив и пролив Босфор Восточный – весь город как на
        ладони.\n" +
        "\n" +
        "Старый дворик ГУМа ~ 30мин\n" +
        "Атмосферное место в самом центре города,
        возможно, кому-то напомнит Европу. И неспроста, ведь это внутренний дворик
        бывшего немецкого торгового дома Кунст и Альберс. Множество арт-объектов
        позволяет еще больше узнать об истории города.");
    }

    }}else if (zpButton.getText().equals("Выбрать"))
    {textField.setText("Информация по экскурсии будет позже");
      textField.setBackground(Color.decode("#98FB98"));
      analysisTA.removeAll();
      JScrollPane scrollPane = new JScrollPane(textField);
      textField.setLineWrap(true);
      textField.setWrapStyleWord(true);
      textField.setEditable(false);

      textField.setComponentOrientation(ComponentOrientation.LEFT_TO_RIGHT);
      textField.setFont(myFont1);
      textField.setCaretPosition(0);

```

```

        analysisTA.add(scrollPane);
        textField.revalidate();
        textField.repaint();
        jf.setVisible(true);
    }
    private void rezButtonActionPerformed(ActionEvent e) {
        tappane.setSelectedIndex(1);
        String s="";
        typTextF.setText((String)
        comboBox.getItemAt(comboBox.getSelectedIndex()));
        for (int i=0;i<3;i++){
            if (factorRadioButton[i].isSelected()){
                eksTextF.setText(factorRadioButton[i].getText());
                eksTextF.setCaretPosition(0);
            }
            imageLabel.setHorizontalAlignment(JLabel.CENTER);
            imageLabel.setIcon(null);
            if ((String)comboBox.getSelectedItem()=="Владивосток"){
                images = new String[]{"D:\\фото\\vladiv1.jpg",
"D:\\фото\\vladiv2.jpg", "D:\\фото\\vladiv3.jpg"};
                setCurrentImage();
                timer = new Timer(2000, new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        currentIndex = (currentIndex + 1) % images.length;
                        setCurrentImage(); // Обновляем изображение
                    }
                });
            }
            if ((String)comboBox.getSelectedItem()=="Владимир"){
                images = new String[]{"D:\\фото\\mir1.jpg", "D:\\фото\\mir2.jpg",
"D:\\фото\\mir3.jpg"};
                setCurrentImage();
                timer = new Timer(2000, new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        currentIndex = (currentIndex + 1) % images.length;
                        setCurrentImage(); // Обновляем изображение
                    }
                });
            }
            if ((String)comboBox.getSelectedItem()=="Москва"){
                images = new String[]{"D:\\фото\\mos.jpg", "D:\\фото\\mos2.jpg",
"D:\\фото\\mos3.jpg"};
                setCurrentImage();
                timer = new Timer(2000, new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        currentIndex = (currentIndex + 1) % images.length;
                        setCurrentImage(); // Обновляем изображение
                    }
                });
            }
            if ((String)comboBox.getSelectedItem()=="Ростов Великий"){
                images = new String[]{"D:\\фото\\ros1.jpg", "D:\\фото\\ros2.jpg",
"D:\\фото\\ros3.jpg"};
                setCurrentImage();
                timer = new Timer(2000, new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        currentIndex = (currentIndex + 1) % images.length;
                        setCurrentImage(); // Обновляем изображение
                    }
                });
            }
            if ((String)comboBox.getSelectedItem()=="Санкт-Петербург"){
                images = new String[]{"D:\\фото\\pit1.jpg", "D:\\фото\\pit2.jpg",

```

```

"D:\\фото\\pit3.jpg");
setCurrentImage();
timer = new Timer(2000, new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        currentIndex = (currentIndex + 1) % images.length;
        setCurrentImage(); // Обновляем изображение
    }
});
}
if ((String)comboBox.getSelectedItem()=="Ярославль"){
    images = new String[]{"D:\\фото\\slav1.jpg",
"D:\\фото\\slav2.jpg", "D:\\фото\\slav3.jpg"};
setCurrentImage();
timer = new Timer(2000, new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        currentIndex = (currentIndex + 1) % images.length;
        setCurrentImage(); // Обновляем изображение
    }
});
}

timer.start();
imageLabel.setBorder(BorderFactory.createLineBorder(Color.GREEN,3));
imageLabel.repaint();
shapLabel.setVisible(true);
}
}

```