

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Институт информационных технологий и  
анализа данных

---

наименование института

**Отчет**

к лабораторной работе №5  
по дисциплине «ООП»

«Разработка приложения с графическим пользовательским интерфейсом»

---

наименование темы

Вариант №20

Выполнил студент

ИСТб-21-1

П.А. Наумов

---

шифр

---

подпись

---

И.О. Фамилия

Проверил

Т.В. Маланова

---

должность

---

подпись

---

И.О. Фамилия

Работа защищена с оценкой

---

Иркутск 2022 г.

## Содержание

1 Постановка задачи .....	3
2 Проектирование графического интерфейса .....	4
3. Проектирование тестов .....	5
4 Проектирование тестов .....	6
5 Листинг программы на Java.....	7
5.1 Главный класс .....	7
5.2 Класс XXXX.....	7
6. Список использованных источников.....	7

## **1 Постановка задачи**

**Цель работы:** Применение объектно-ориентированного подхода в разработке приложений с графическим интерфейсом.

### **Задачи:**

- Получение общего представления о шаблоне программирования Модель/Вид/Контроллер и его практического применения;
- Получение практического опыта работы с базовыми графическими библиотеками java.

### **Содержание задания**

Необходимо разработать графический пользовательский интерфейс средствами java для работы с классами, реализованными при выполнении 4-ой лабораторной работы.

Графический пользовательский интерфейс должен предоставлять возможности работы со всеми возможными функциями программы (ввод и вывод данных в соответствующей форме, изменение условий выполнения задания, работа с файлами и т.д.). Состав используемых компонентов определяется самостоятельно, но для них должны быть предусмотрены соответствующие планировщики раскладки компонентов в основных контейнерах.

Классы, реализующие графический пользовательский интерфейс, должны содержать только обращения к методам классов для выполнения заданий. Для этого может понадобиться модифицировать программный код из предыдущих лабораторных работ.

## 2 Проектирование графического интерфейса

На главном окне с помощью компоновщика FlowLayout размещены кнопки, предназначенные для навигации по главному меню приложения. Внешний вид окна представлен на рисунке 1.1.

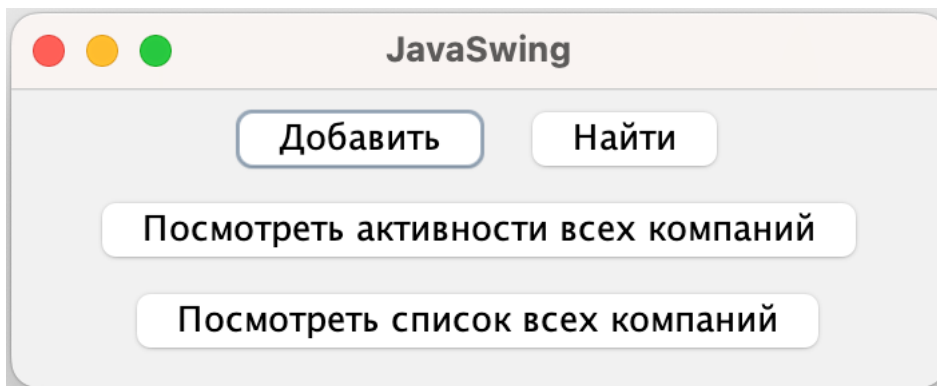


Рисунок 1.1 – Главное окно программы

На модальном окне AddCompanyView расположены Radio Button, TextField и Button. Модальное окно представлено на рисунке 1.2.

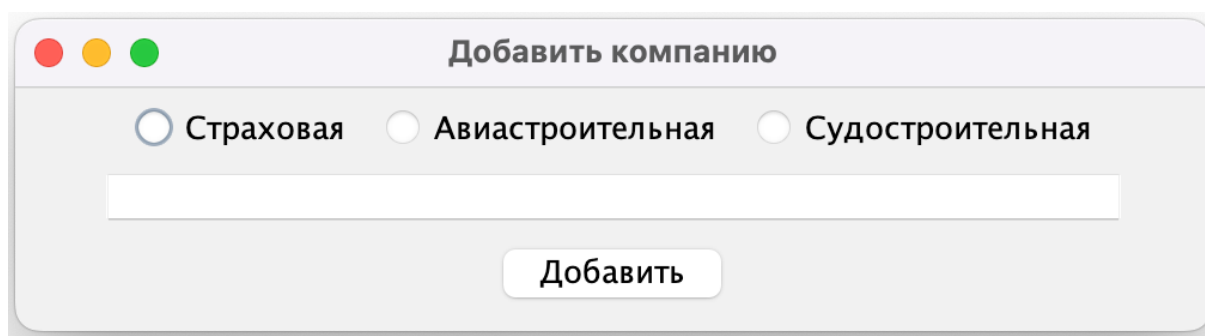


Рисунок 1.2 – Окно AddCompanyView

На модальном окне FindCompanyView расположены TextField и Button. Модальное окно FindCompanyView представлено на рисунке 1.3.

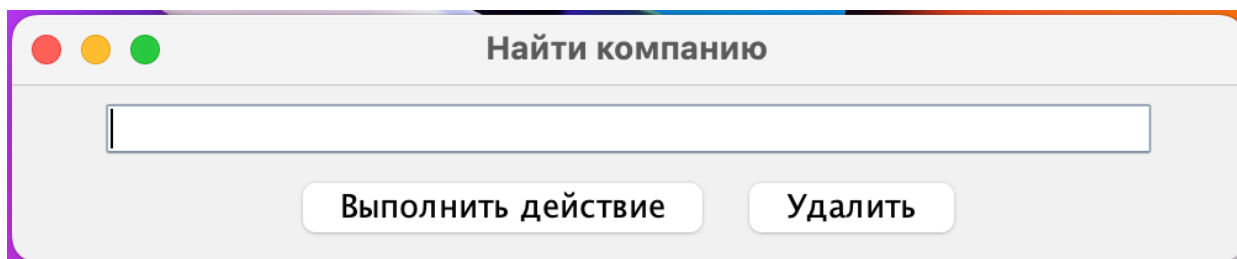


Рисунок 1.3 – Окно FindCompanyView

Из модального окна можно вызвать диалоговое окно с информацией о активности выбранной компании. Диалоговое окно представлено на рисунке 1.4

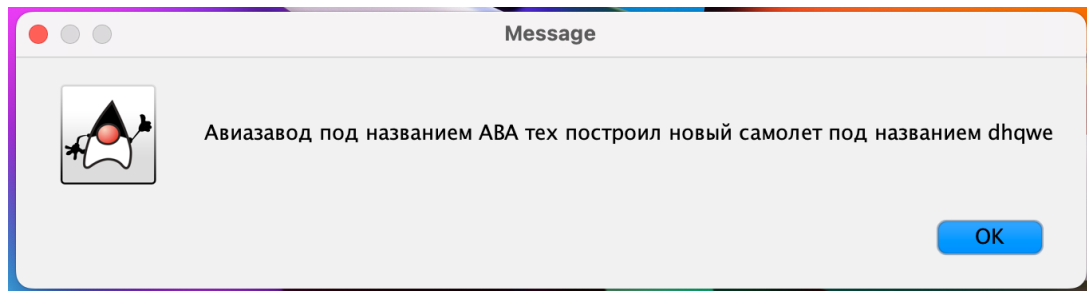
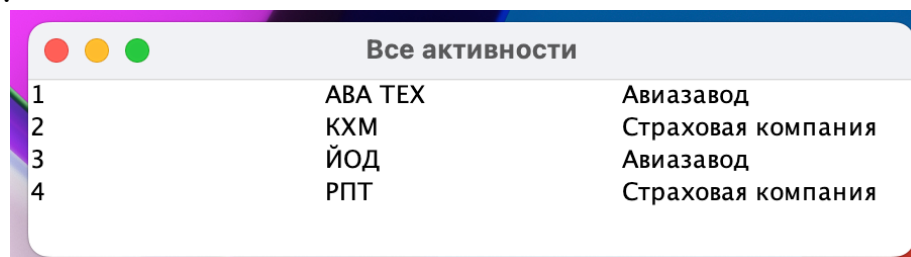


Рисунок 1.4 – Диалоговое окно

На модальном окне AllActivitiesView находится один компонент JTable. Содержание таблицы зависит от того, нажатие какой кнопки вызывает ее открытие. Если нажать в главном меню на кнопку «Посмотреть активности всех компаний», то содержание таблицы будет представлять из себя таблицу с колонками: Номер, компания, действие. Если нажать на кнопку «Посмотреть список всех компаний», то содержание таблицы будет представлять из себя таблицу с колонками: Номер, компания, название. Представление таблицы со списком всех компаний представлено на рисунке 1.5.



	Все активности	
1	АВА ТЕХ	Авиазаовод
2	КХМ	Страховая компания
3	ЙОД	Авиазаовод
4	РПТ	Страховая компания

Рисунок 1.5 – Окно AllActivitiesView

### 3 Проектирование классов

На рисунке 3 представлена UML-диаграмма проекта.

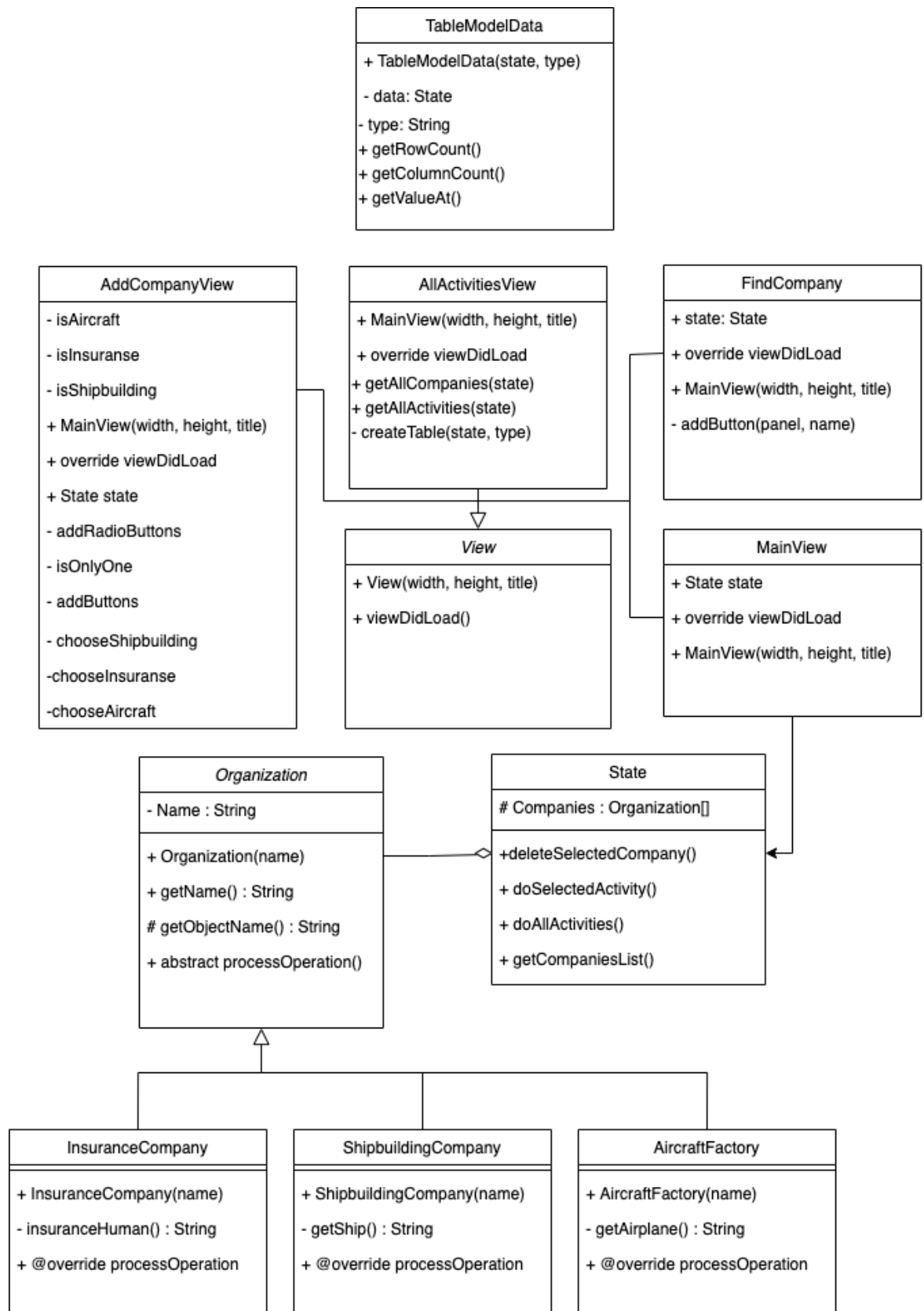


Рисунок 3 – UML-диаграмма проекта

#### 4 Проектирование тестов

В ходе разработки было проведено тестирование программы, результаты которого представлены в табл. 1.

Таблица 1 – Тестирование программы

№	Действие	Результат действия
1	Первоначальный запуск программы	При старте программы появляется главная форма.
2	Попытка посмотреть список компаний, не добавляя их заранее.	Приложение покажет пустую таблицу
3	Попытка выполнить действие компании, которой не существует.	Приложение показывает диалоговое окно с ошибкой.
4	Попытка посмотреть активности компаний, не добавляя их заранее.	Приложение покажет пустую таблицу
5	Создание нового объекта и поиск его в списке объектов.	При добавлении объекта приложение не дает пользователю выбрать сразу несколько типов компаний. А после добавления компания добавляется в таблицу.

## Результаты тестирования

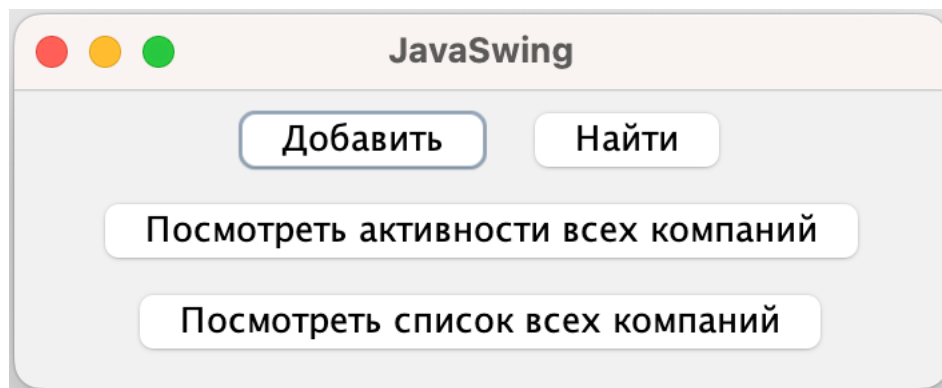


Рисунок 4.1 – Результат теста 1



Рисунок 4.2 – результат теста 2

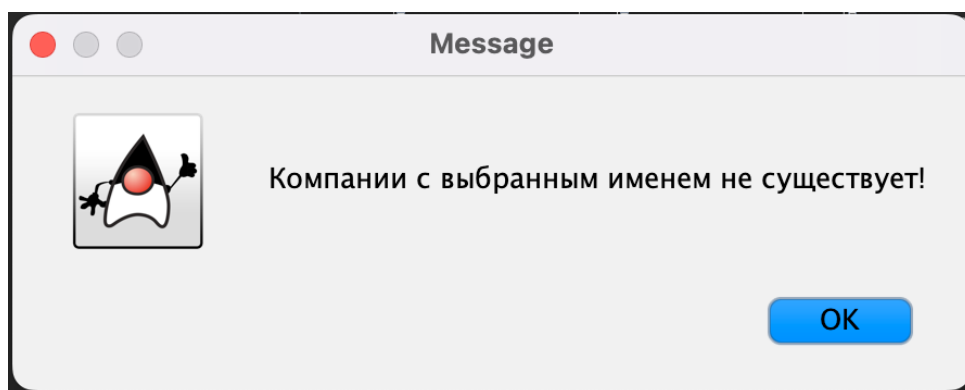


Рисунок 4.3 – результат теста 3



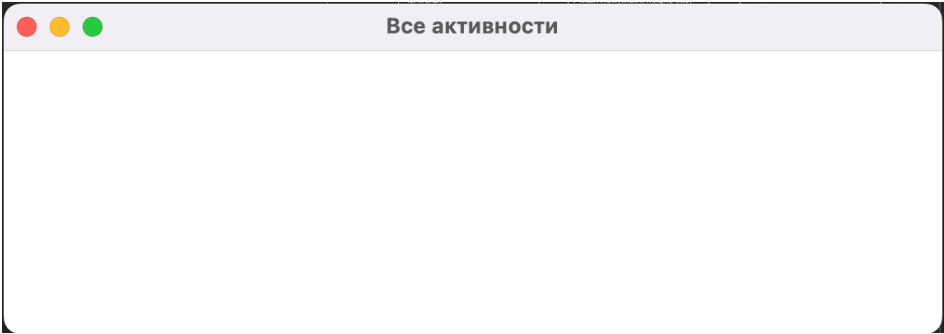


Рисунок 4.4 – результат теста 4

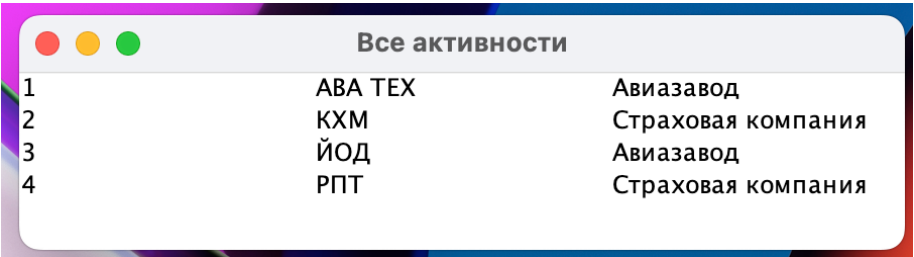


Рисунок 4.5 – результат теста 5

## 5 Листинг программы на Java

### 5.1 Главный класс

```
package malinatrash.App;

import malinatrash.ConsoleInterface.Menu;
import malinatrash.Views.MainView;

public class Main {

    public static void main(String[] args) {
        MainView view = new MainView(350, 140, "JavaSwing");
        view.invalidate();
        view.validate();
        view.repaint();
        Menu.start();
    }
}
```

### 5.2 Класс View

```
package malinatrash.Views;

import javax.swing.*.*;
import java.awt.*.*;

interface IView {
    void viewDidLoad();
}

public class View extends JFrame implements IView {
    public View(int width, int height, String title) {
        setTitle(title);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        int screenWidth =
Toolkit.getDefaultToolkit().getScreenSize().width;
        int screenHeight =
Toolkit.getDefaultToolkit().getScreenSize().height;
        setBounds((screenWidth - width) / 2, (screenHeight - height) /
2, width, height);
        setVisible(true);
        viewDidLoad();
    }
    @Override
    public void viewDidLoad() {
    }
}
```

### 5.3 Класс MainView

```
package malinatrash.Views;

import malinatrash.Model.State;

import javax.swing.*.*;
import java.awt.*.*;

public class MainView extends View {
    State state = new State();
    public MainView(int width, int height, String title) {
        super(width, height, title);
    }

    @Override
    public void viewDidLoad() {
        super.viewDidLoad();
        JPanel buttonsPanel = new JPanel();
        buttonsPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
        add(buttonsPanel);

        JButton addButton = new JButton("Добавить");
        buttonsPanel.add(addButton);

        addButton.addActionListener(e -> createAddCompanyView());

        JButton findButton = new JButton("Найти");
        buttonsPanel.add(findButton);
        findButton.addActionListener(e -> createFindCompanyView());

        JButton doAllActivitiesButton = new JButton("Посмотреть
активности всех компаний");
        buttonsPanel.add(doAllActivitiesButton);
        doAllActivitiesButton.addActionListener(e ->
getAllActivities());

        JButton allCompaniesButton = new JButton("Посмотреть список всех
компаний");
        buttonsPanel.add(allCompaniesButton);
        allCompaniesButton.addActionListener(e -> getAllCompanies());
    }
    private void createAddCompanyView() {
        AddCompanyView addCompanyView = new AddCompanyView(500, 130,
"Добавить компанию");
        addCompanyView.state = this.state;
    }
    private void createFindCompanyView() {
        FindCompanyView findCompanyView = new FindCompanyView(500, 100,
"Найти компанию");
        findCompanyView.state = this.state;
    }
    private void getAllActivities() {
        AllActivitiesView allActivitiesView = new AllActivitiesView(700,
300, "Все активности");
        allActivitiesView.getAllActivities(state);
    }
}
```

```

    }
    private void getAllCompanies() {
        AllActivitiesView allActivitiesView = new AllActivitiesView(570,
200, "Все активности");
        allActivitiesView.getAllCompanies(state);
    }
}

```

## 5.4 Класс AddCompanyView

```

package malinatrash.Views;

import malinatrash.Instances.AircraftFactory;
import malinatrash.Instances.InsuranceCompany;
import malinatrash.Instances.ShipbuildingCompany;

import malinatrash.Model.State;

import javax.swing.*.*;

public class AddCompanyView extends View {
    State state;
    private boolean isAircraft = false;
    private boolean isInsurance = false;
    private boolean isShipbuilding = false;
    AddCompanyView(int width, int height, String title) {
        super(width, height, title);
        setDefaultCloseOperation(HIDE_ON_CLOSE);
    }

    @Override
    public void viewDidLoad() {
        super.viewDidLoad();
        JPanel panel = new JPanel();
        add(panel);
        addRadioButtons(panel);
        JTextField companyName = new JTextField(35);
        panel.add(companyName);
        addButton(panel, companyName);
    }
    private void addRadioButtons(JPanel panel) {
        JRadioButton isAircraft = new JRadioButton("Авиастроительная");
        JRadioButton isInsurance = new JRadioButton("Страховая");
        JRadioButton isShipbuilding = new
JRadioButton("Судостроительная");

        isAircraft.addActionListener(e -> {
            chooseAircraft();
            if (isOnlyOne()) {
                this.isShipbuilding = false;
                isShipbuilding.setSelected(false);
                this.isInsurance = false;
                isInsurance.setSelected(false);
            }
        });
        isShipbuilding.addActionListener(e -> {
            chooseShipbuilding();

```

```

        if (isOnlyOne()) {
            this.isAircraft = false;
            isAircraft.setSelected(false);
            this.isInsurance = false;
            isInsurance.setSelected(false);
        }
    });
    isInsurance.addActionListener(e -> {
        chooseInsurance();
        if (isOnlyOne()) {
            this.isShipbuilding = false;
            isShipbuilding.setSelected(false);
            this.isAircraft = false;
            isAircraft.setSelected(false);
        }
    });

    panel.add(isInsurance);
    panel.add(isAircraft);
    panel.add(isShipbuilding);
}

private void addButton(JPanel panel, JTextField button) {
    JButton addButton = new JButton("Добавить");
    addButton.addActionListener(e -> {
        if (isInsurance) {
            state.addOrganization(new
InsuranceCompany(button.getText()));
        } else if (isAircraft) {
            state.addOrganization(new
AircraftFactory(button.getText()));
        } else if (isShipbuilding) {
            state.addOrganization(new
ShipbuildingCompany(button.getText()));
        }
        setVisible(false);
    });
    panel.add(addButton);
}

private boolean isOnlyOne() {
    int count = 0;
    if (isAircraft) {
        count++;
    } if (isInsurance) {
        count++;
    } if (isShipbuilding) {
        count++;
    }
    return count != 1;
}

private void chooseAircraft() {
    isAircraft = true;
}

private void chooseInsurance() {
    isInsurance = true;
}

private void chooseShipbuilding() {

```

```

        isShipbuilding = true;
    }
}

```

## 5.5 Класс AllActivitiesView

```

package malinatrash.Views;

import malinatrash.Model.State;
import malinatrash.Model.TableModelData;

import javax.swing.*.*;
import java.awt.*.*;

public class AllActivitiesView extends View {
    AllActivitiesView(int width, int height, String title) {
        super(width, height, title);
        setDefaultCloseOperation(HIDE_ON_CLOSE);
    }
    public void getAllCompanies(State state) {
        createTable(state, "Companies");
    }
    public void getAllActivities(State state) {
        createTable(state, "Activities");
    }
    private void createTable(State state, String type) {
        JTable table = new JTable();
        JScrollPane scrollPane = new JScrollPane(table);
        scrollPane.setPreferredSize(new Dimension(480, 300));
        if (type.equals("Companies")) {
            table.setModel(new TableModelData(state, type));
        }
        table.setModel(new TableModelData(state, type));
        add(table);
    }
}

```

## 5.6 Класс FindCompanyView

```

package malinatrash.Views;

import malinatrash.Model.State;

import javax.swing.*.*;

import static javax.swing.JOptionPane.showMessageDialog;

public class FindCompanyView extends View {
    State state;
    FindCompanyView(int width, int height, String title) {
        super(width, height, title);
        setDefaultCloseOperation(HIDE_ON_CLOSE);
    }

    @Override
    public void viewDidLoad() {
        super.viewDidLoad();
        JPanel panel = new JPanel();
    }
}

```

```

        add(panel);
        JTextField companyName = new JTextField(35);
        panel.add(companyName);
        addButton(panel, companyName);
    }
    private void addButton(JPanel panel, JTextField companyName) {
        JButton findButton = new JButton("Выполнить действие");
        findButton.addActionListener(e -> {
            setVisible(false);
            showMessageDialog(null,
state.doSelectedActivity(companyName.getText()));
        });
        panel.add(findButton);

        JButton deleteButton = new JButton("Удалить");
        deleteButton.addActionListener(e -> {
            state.deleteSelectedCompany(companyName.getText());
            setVisible(false);
        });
        panel.add(deleteButton);
    }
}

```

## 5.7 Класс State

```

package malinatrash.Model;
import malinatrash.Instances.Organization;
import java.util.ArrayList;
public class State {
    private final ArrayList<Organization> organizations
= new ArrayList<>();
    public int getSize() {
        return organizations.size();
    }
    public Organization getOrganization(int index) {
        return organizations.get(index);
    }
    public void deleteSelectedCompany(String name) {
        for (Organization company: organizations) {
            if (company.getName() == null ? name == null :
company.getName().equals(name)) {
                organizations.remove(company);
                System.out.println("Успешно удалено!");
                return;
            }
        }
        System.out.println("Компании с выбранным именем
не существует!");
    }
    public String doSelectedActivity(String name) {
        for (Organization company: organizations) {

```

```

        if (company.getName() == null ? name == null :
company.getName().equals(name)) {
            return company.processOperation();
        }
    }
    return ("Компании с выбранным именем не
существует!");
}
    public void addOrganization(Organization
organization) {
        organizations.add(organization);
    }
}

```

## 5.8 Класс TableModelData

```
package malinatrash.Model;
```

```
import javax.swing.table.AbstractTableModel;
```

```

public class TableModelData extends AbstractTableModel {
    private final State data;
    private final String type;
    public TableModelData(State state, String type) {
        this.data = state;
        this.type = type;
    }
    @Override
    public int getRowCount() {
        return data.getSize();
    }
    @Override
    public int getColumnCount() {
        return 3;
    }
    @Override
    public Object getValueAt(int rowIndex, int columnIndex) {
        switch (columnIndex) {
            case 0: return rowIndex + 1;
            case 1: return data.getOrganization(rowIndex).getName();
            case 2:
                if (type.equals("Companies")) {
                    return data.getOrganization(rowIndex).getType();
                }
                return
data.getOrganization(rowIndex).processOperation();
        }
        return null;
    }
}

```



## 5.8 Класс Organization

```
package malinatrash.Instances;

import java.util.Random;

public abstract class Organization {

    private final String name;
    private String type;
    public Organization(String name) {
        this.name = name;
        this.type = "";
    }
    public String getName() {
        return name;
    }
    public String getType() {
        return type;
    }

    protected String getObjetName() {
        int leftLimit = 97; // letter 'a'
        int rightLimit = 122; // letter 'z'
        Random random = new Random();
        StringBuilder buffer = new StringBuilder(5);
        for (int i = 0; i < 5; i++) {
            int randomLimitedInt = leftLimit + (int)
                (random.nextFloat() * (rightLimit - leftLimit + 1));
            buffer.append((char) randomLimitedInt);
        }

        return buffer.toString();
    }
    public void setType(String type) {
        this.type = type;
    }
    public abstract String processOperation();
}
```

## 5.9 Класс AircraftFactory

```
package malinatrash.Instances;

public class AircraftFactory extends Organization {

    public AircraftFactory(String name) {
        super(name);
        setType("Авиазавод");
    }
    @Override
    public String processOperation() {
        return (getType() + " под названием " + getName() + " построил
новый самолет под названием " + craftAirPlane());
    }
}
```

```

        private String craftAirPlane() {
            return getObjetName();
        }
    }
}

```

## 5.10 Класс InsuranceCompany

```

package malinatrash.Instances;

public class InsuranceCompany extends Organization {

    public InsuranceCompany(String name) {
        super(name);
        setType("Страховая компания");
    }

    private String insuranceHuman() {
        return getObjetName();
    }

    @Override
    public String processOperation() {
        return (getType() + " под названием " + getName() + "
застраховала человека с ID: " + insuranceHuman());
    }
}

```

## 5.11 Класс ShipbuildingCompany

```

package malinatrash.Instances;

public class ShipbuildingCompany extends Organization {

    public ShipbuildingCompany(String name) {
        super(name);
        setType("Судостроительная компания");
    }

    public String getShip() {
        return getObjetName();
    }

    @Override
    public String processOperation() {
        return (getType() + " под названием " + getName() + "
построила новое судно с названием: " + getShip());
    }
}

```

## **Список литературы**

1.     Статья на Хабре: "Охота на мифический MVC (часть первая)"  
(Дата обращения 20.02.2022)
2.     Статья на Хабре: "Охота на мифический MVC (часть вторая)"  
(Дата обращения 20.02.2022)