



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

МАГИСТЕРСКАЯ ПРОГРАММА 09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных.

О Т Ч Е Т

по лабораторной работе № 4

Вариант 6

Название: Внутренние классы и интерфейсы в Java

Дисциплина: Языки программирования для работы с большими
данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

В.А. Елисеев

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Цель работы: получение навыков работы с внутренними классами и интерфейсами языка программирования Java.

Задание 1:

6. Создать класс Shop (магазин) с внутренним классом, с помощью объектов которого можно хранить информацию об отделах, товарах и услуг.
7. Создать класс Справочная Служба Общественного Транспорта с внутренним классом, с помощью объектов которого можно хранить информацию о времени, линиях маршрутов и стоимости проезда.

Подзадача 1.

Код программы:

```

public class Shop {
    public class Department {
        public class Product {
            String name;
            int price;

            Product(String name, int price) {
                this.name = name;
                this.price = price;
            }
        }
    }

    public class Service {
        String name;
        int price;

        Service(String name, int price) {
            this.name = name;
            this.price = price;
        }
    }

    String name;
    Product[] products;
    Service[] services;

    Department(String name, Product[] products, Service[] services) {
        this.name = name;
        this.products = products;
        this.services = services;
    }
}

String name;
Department[] departments;

Shop(String name, Department[] departments) {
    this.name = name;
    this.departments = departments;
}
}

```

Подзадача 2.

Код программы:

```

public class PublicTransportService {
    public class Route {
        public class RouteTime {
            String dispatchTime;
            String arriveTime;

            RouteTime(String dispatchTime, String arriveTime) {
                this.dispatchTime = dispatchTime;
                this.arriveTime = arriveTime;
            }
        }

        RouteTime[] schedule;
        int fare;

        Route(RouteTime[] schedule, int fare) {
            this.schedule = schedule;
            this.fare = fare;
        }
    }

    Route[] routes;

    PublicTransportService(Route[] routes) {
        this.routes = routes;
    }
}

```

Задание 2:

6. interface Корабль <- abstract class Военный Корабль <- class Авианосец.
7. interface Врач <- class Хирург <- class Нейрохирург.

Подзадача 1.

Код программы:

```

public interface IShip {
    public void swim();
}

```

```

public abstract class WarShip implements IShip {
    public String[] weapons;

    public void fire(String position) {
        System.out.println("Я выстрелил в " + position);
    };

    WarShip(String[] weapons) {
        this.weapons = weapons;
    }

    WarShip(int weaponCount) {
        this.weapons = new String[weaponCount];
    }
}

```

```

public class AircraftCarrier extends WarShip {
    AircraftCarrier(String[] weapons) {
        super(weapons);
        //TODO Auto-generated constructor stub
    }

    AircraftCarrier(int weaponCount) {
        super(weaponCount);
        //TODO Auto-generated constructor stub
    }

    @Override
    public void swim() {
        // TODO Auto-generated method stub
    }
}

```

Подзадача 2.

Код программы:

```

public interface IDoctor {
    public void treat(String person);
}

```

```
public abstract class Surgeon implements IDoctor {
    private void treatWithCare(String person) {
        System.out.println("Я вылечил " + person + " с хирургической точностью");
    }

    @Override
    public void treat(String person) {
        treatWithCare(person);
    }
}
```

```
public class NeuroSurgeon extends Surgeon {
    String name;

    NeuroSurgeon(String name) {
    }

    @Override
    public void treat(String person) {
        System.out.println("Я вылечил " + person + " с нейрохирургической точностью");
    }
}
```

Ссылка на программное решение:

<https://github.com/ArMaxik/BigDataLanguages/tree/main/lr4>

Вывод: в ходе лабораторной работы были получены навыки работы с внутренними классами и интерфейсами языка программирования Java.