



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 № 17

Название: Внутренние классы. Интерфейсы

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

М.О. Усманов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Цель работы

Получение первичных навыков работы с внутренними классами и интерфейсами языка программирования Java.

Ход работы

Задание 1.

– Создать класс Справочная Служба Общественного Транспорта с внутренним классом, с помощью объектов которого можно хранить информацию о времени, линиях маршрутов и стоимости проезда.

– Создать класс Computer (компьютер) с внутренним классом, с помощью объектов которого можно хранить информацию об операционной системе, процессоре и оперативной памяти.

Листинг 1 – Код класса «Helpdesk»

```
import java.util.ArrayList;

public class Helpdesk {
    ArrayList<Route> routes = new ArrayList<>();

    public void addRoute(String time, String destination, int cost) {
        routes.add(
            new Route(time, destination, cost)
        );
    }

    class Route {
        String time;
        String destination;
        int cost;

        public Route(String time, String destination, int cost) {
            this.time = time;
            this.destination = destination;
            this.cost = cost;
        }

        public void setTime(String time) {
            this.time = time;
        }

        public void setDestination(String destination) {
            this.destination = destination;
        }
    }
}
```

```

    public void setCost(int cost) {
        this.cost = cost;
    }

    public String getTime() {
        return time;
    }

    public String getDestination() {
        return destination;
    }

    public int getCost() {
        return cost;
    }

    @Override
    public String toString() {
        return "Route{" +
            "time='" + time + '\'' +
            ", destination='" + destination + '\'' +
            ", cost=" + cost +
            '}';
    }
}

```

Листинг 2 – Класс «Computer»

```

public class Computer {
    Parameters parameters;

    public Computer(String os, String processor, int ram) {
        parameters = new Parameters(os, processor, ram);
    }

    class Parameters {
        String os;
        String processor;
        int ram;

        public Parameters(String os, String processor, int ram) {
            this.os = os;
            this.processor = processor;
            this.ram = ram;
        }

        public void setOs(String os) {
            this.os = os;
        }

        public void setProcessor(String processor) {
            this.processor = processor;
        }

        public void setRam(int ram) {

```

```

        this.ram = ram;
    }

    public String getOs() {
        return os;
    }

    public String getProcessor() {
        return processor;
    }

    public int getRam() {
        return ram;
    }

    @Override
    public String toString() {
        return "Computer{" +
                "os='" + os + '\'' +
                ", processor='" + processor + '\'' +
                ", RAM=" + ram +
                '}';
    }
}

```

Задание 2.

Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов

- interface Врач <- class Хирург <- class Нейрохирург.
- interface Корабль <- class Грузовой Корабль <- class Танкер.

Листинг 3 – Код интерфейса «Ship»

```

public interface Ship {

    public void travelTo(String destination);

}

```

Листинг 4 – Код класса «Cargo»

```

public class Cargo implements Ship {

    String destination;
    public int payload = 10;

    @Override
    public void travelTo(String destination) {
        this.destination = destination;
    }

}

```

```
}
```

Листинг 5 – Код класса «Tanker»

```
public class Tanker extends Cargo {  
  
    int fuel;  
  
    public Tanker() {  
        payload = 150;  
        fuel = 0;  
    }  
  
    public void refuel(int amount) {  
        fuel += amount;  
    }  
  
}
```

Листинг 6 – Код интерфейса «Physician»

```
public interface Physician {  
    public void doTreatment();  
}
```

Листинг 7 – Код класса «Surgeon»

```
public class Surgeon implements Physician {  
  
    String area = "Surgery";  
  
    @Override  
    public void doTreatment() {  
        System.out.println("Вырезан аппендицит");  
    }  
  
}
```

Листинг 8 – Код класса «Neurosurgeon»

```
public class Neurosurgeon extends Surgeon {  
  
    @Override  
    public void doTreatment() {  
        System.out.println("Проведена операция на мозге");  
    }  
  
    public void doScan() {  
        System.out.println("Сделана энцефалограмма");  
    }  
  
}
```

Результатом выполнения задания является структура интерфейсов и классов, соответствующая условию поставленной задачи.

Местоположение репозитория с файлами проекта

Файлы проекта расположены в репозитории веб-платформы для совместной разработки Github. Местоположение в репозитории:

https://github.com/s314/big-data-studies/tree/main/lab_4_var_1_7_8

https://github.com/s314/big-data-studies/tree/main/lab_4_var_2_7_8

Вывод

По итогам выполнения лабораторной работы были получены навыки программирования с использованием внутренних классов и интерфейсов на языке Java. Была разработаны классовые структуры, позволяющие максимально охватить изучаемую тему.