|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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.**

**Отчет**

|  |
| --- |
| **по лабораторной работе № 4** |
|  |
| **Вариант 16** |

**Название:**

Внутренние классы и интерфейсы

**Дисциплина:** Языки программирования для работы с большими данными

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-23М | |  |  | В.И. Тарасов |
|  | | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  | |  |  |  |  |
| Преподаватель | |  |  |  | П.В. Степанов |
|  | |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Цель работы** –получение навыков работы с внутренними классами и интерфейсами в языке Java.

**Ход работы**

**Задание 1:**

1. Создать класс Shop (магазин) с внутренним классом, с помощью объектов которого можно хранить информацию об отделах, товарах и услуг.
2. Создать класс Cправочная Cлужба Oбщественного Tранспорта с внутренним классом, с помощью объектов которого можно хранить информацию о времени, линиях маршрутов и стоимости проезда.

Листинг 1 – Код класса HelpDesk

package src.lab4\_1;

import java.util.ArrayList;

public class HelpDesk {

ArrayList<Route> routes = new ArrayList<Route>();

public void addRoute(String time, String destination, int cost){

routes.add(new Route(time, destination, cost));

}

public 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 setCost(int cost) {

this.cost = cost;

}

public void setDestination(String destination) {

this.destination = destination;

}

public void setTime(String time) {

this.time = time;

}

@Override

public String toString() {

return "Route{" +

"time='" + time + '\'' +

", destination='" + destination + '\'' +

", cost=" + cost +

'}';

}

}

}

Листинг 2 – Код класса Shop

package src.lab4\_1;

import java.util.ArrayList;

public class Shop {

ArrayList<Aisle> aisles = new ArrayList<Aisle>();

public void addAisle(String department, String product, String service){

aisles.add(new Aisle(department, product, service));

}

public class Aisle {

String department;

String product;

String service;

public Aisle(String department, String product, String service){

this.department = department;

this.product = product;

this.service = service;

}

}

}

**Задание 2:**

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

1. interface Корабль <- abstract class Военный Корабль <- class Авианосец.
2. interface Врач <- class Хирург <- class Нейрохирург.

Листинг 3 – Код интерфейса Ship

package src.lab4\_2;

public interface Ship {

void performAction();

}

Листинг 4 – Код абстрактного класса BattleShip

package src.lab4\_2;

public abstract class BattleShip implements Ship{

public String name;

public void performAction(){

System.out.println("Выстрелить из пушки");

}

}

Листинг 5 – Код класса AircraftCarrier

package src.lab4\_2;

public class AircraftCarrier extends BattleShip{

public String name = "Carrier";

public void performAction(){

System.out.println("Запустить самолет");

}

}

Листинг 6 – Код интерфейса Physician

package src.lab4\_2;

public interface Physician {

public void doTreatment();

}

Листинг 7 – Код класса Surgeon

package src.lab4\_2;

public class Surgeon implements Physician {

String area = "Surgery";

@Override

public void doTreatment(){

System.out.println("Вправлено плечо");

}

}

Листинг 8 – Код класса Neurosurgeon

package src.lab4\_2;

public class Neurosurgeon extends Surgeon{

@Override

public void doTreatment(){

System.out.println("Удален мозг");

}

public void doScan(){

System.out.println("Сделана энцефалограмма");

}

}

Результатом выполнения является классовая структура, соответствующая заданной.

**Местоположение проектных файлов** –файлы проекта расположены в репозитории GitHub по следующим адресам:

<https://github.com/Fregatty/BigDataLanguages/tree/main/src/lab4_1>

<https://github.com/Fregatty/BigDataLanguages/tree/main/src/lab4_2>

**Вывод** – в результате выполнения лабораторной работы были получены навыки работы с внутренними классами и интерфейсами языка Java. Была разработана классовая структура, включающая все варианты (интерфейсы, абстрактные классы и обычные классы).