|  |  |
| --- | --- |
| 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 №** 8 |  |

**Название:**

Внутренние классы. Интерфейсы

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

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

Москва, 2022

## Цель работы

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

## Ход работы

*Задание 1.*

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

9. Создать класс Park (парк) с внутренним классом, с помощью объектов которого можно хранить информацию об аттракционах, времени их работы и стоимости.

Листинг 1 – Код класса «Park»

import java.util.ArrayList;

public class Park {

ArrayList<Attraction> attractions = new ArrayList<>();

public void AddAtraction(String title, int openTime, int closeTime, int cost) {

attractions.add(new Attraction(title, openTime, closeTime, cost));

}

public class Attraction {

String title;

int openTime;

int closeTime;

int cost;

public Attraction(String title, int openTime, int closeTime, int cost) {

this.title = title;

this.openTime = openTime;

this.closeTime = closeTime;

this.cost = cost;

}

public String getTitle() {

return title;

}

public int getOpenTime() {

return openTime;

}

public int getCloseTime() {

return closeTime;

}

public int getCost() {

return 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

8. interface Корабль <- class Грузовой Корабль <- class Танкер.

9. interface Мебель <- abstract class Шкаф <- class Книжный Шкаф.

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 «Furniture»

public interface Furniture {

public void Stash(String something);

public int getAmount();

}

Листинг 7 – Код абстрактного класса «Furniture»

public abstract class Cupboard implements Furniture {

int shelves;

@Override

public void Stash(String something) {

System.out.println("Stashed " + something);

}

}

Листинг 8 – Код класса «Bookshelf»

public class Bookshelf extends Cupboard {

public Bookshelf(int shelves) {

this.shelves = shelves;

}

@Override

public int getAmount() {

return shelves;

}

}

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

## 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.