|  |  |
| --- | --- |
| 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 09.04.01/07 **Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

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

**Название: Внутренние Внутренн**

Внутренние классы, интерфейсы

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

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

Москва, 2022

**Вариант 1 и 2, номера 9,10.**

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

10. Создать класс Cinema (кино) с внутренним классом, с помощью объектов которого можно хранить информацию об адресах кинотеатров, фильмах и времени сеансов.

Код программы:

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

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

10. interface Фильм <- class Отечественный Фильм <- class Комедия.

**Код программы:**

*package daniil;*

*import java.util.ArrayList;*

*public class lab {*

*public static class Park{*

*String name;*

*public ArrayList<Attraction> list\_attr;*

*public Park() {*

*list\_attr = new ArrayList<>();*

*}*

*public Park(String name) {*

*this.name = name;*

*list\_attr = new ArrayList<>();*

*}*

*public void add\_attr(String name, int drive\_time, int open\_time, int close\_time, int cost){*

*list\_attr.add(new Attraction(name, drive\_time, open\_time, close\_time, cost));*

*}*

*public static class Attraction{*

*// хранить информацию об аттракционах, времени их работы и стоимости.*

*String name;*

*int drive\_time;*

*int open\_time;*

*int close\_time;*

*int cost;*

*public Attraction(String name, int drive\_time, int open\_time, int close\_time, int cost) {*

*this.name = name;*

*this.drive\_time = drive\_time;*

*this.open\_time = open\_time;*

*this.close\_time = close\_time;*

*this.cost = cost;*

*}*

*@Override*

*public String toString() {*

*return "Attraction{" +*

*"name='" + name + '\'' +*

*", drive\_time=" + drive\_time +*

*", open\_time=" + open\_time +*

*", close\_time=" + close\_time +*

*", cost=" + cost +*

*'}';*

*}*

*}*

*public static class Cinema{*

*String name;*

*ArrayList<Addreses> cin\_list;*

*public Cinema() {*

*cin\_list = new ArrayList<>();*

*}*

*public Cinema(String name) {*

*this.name = name;*

*cin\_list = new ArrayList<>();*

*}*

*public void add\_new\_address(String addres, int time\_open, int time\_close){*

*cin\_list.add(new Addreses(addres, time\_open, time\_close));*

*}*

*@Override*

*public String toString() {*

*return "Cinema{" +*

*"name='" + name + '\'' +*

*", cin\_list=" + cin\_list +*

*'}';*

*}*

*public static class Addreses{*

*String addres;*

*int time\_open;*

*int time\_close;*

*ArrayList<String> film\_list;*

*ArrayList<String> time\_spend;*

*ArrayList<Integer> cost;*

*public Addreses() {*

*film\_list = new ArrayList<>();*

*time\_spend = new ArrayList<>();*

*cost = new ArrayList<>();*

*}*

*public Addreses(String addres, int time\_open, int time\_close) {*

*this.addres = addres;*

*this.time\_open = time\_open;*

*this.time\_close = time\_close;*

*film\_list = new ArrayList<>();*

*time\_spend = new ArrayList<>();*

*cost = new ArrayList<>();*

*}*

*public void add\_new\_film(String film, String time, Integer prise){*

*film\_list.add(new String(film));*

*time\_spend.add(new String(time));*

*cost.add(new Integer(prise));*

*}*

*@Override*

*public String toString() {*

*return "Addreses{" +*

*"addres='" + addres + '\'' +*

*", time\_open=" + time\_open +*

*", time\_close=" + time\_close +*

*", film\_list=" + film\_list +*

*", time\_spend=" + time\_spend +*

*", cost=" + cost +*

*'}';*

*}*

*}*

*}*

*}*

*interface Furniture{*

*public String getname();*

*}*

*public static abstract class Wardrobe implements Furniture{*

*String name;*

*int cost;*

*public String getname(){*

*return this.name;*

*}*

*public Wardrobe() {*

*}*

*public Wardrobe(String name, int cost) {*

*this.name = name;*

*this.cost = cost;*

*}*

*public void test(){*

*System.out.println("Test");*

*}*

*}*

*public static class Book\_war extends Wardrobe{*

*@Override*

*public void test(){*

*System.out.println("Book");*

*}*

*}*

*interface Film{*

*String getname();*

*}*

*public static class Rus\_film implements Film{*

*String name;*

*int duration;*

*public Rus\_film() {*

*}*

*public Rus\_film(String name, int duration) {*

*this.name = name;*

*this.duration = duration;*

*}*

*public String getname(){*

*return this.name;*

*}*

*}*

*public static class Comedy extends Rus\_film{*

*int time\_release;*

*public Comedy(int time\_release) {*

*this.time\_release = time\_release;*

*}*

*public Comedy() {*

*}*

*public String getname(){*

*return Integer.toString(this.time\_release);*

*}*

*}*

*public static void main(String[] args) {*

*}*

*}*

**Вывод:** научились работать с внутренними классами и интерфейсами.