



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

Название: Внутренние классы и интерфейсы. Вариант 9

Дисциплина: Языки программирования для работы с большими
данными

Студент ИУ6-22М
(Группа)

К.Ю. Каташинский
(Подпись, дата) (И.О. Фамилия)

Преподаватель

П.В. Степанов
(Подпись, дата) (И.О. Фамилия)

Москва, 2024

Цель работы

Изучить работу внутренних классов и интерфейсов в языке программирования Java.

Ход работы

Задание 1. Создать класс Park (парк) с внутренним классом, с помощью объектов которого можно хранить информацию об аттракционах, времени их работы и стоимости.

Код программы представлен в листинге 1, результат – на рисунке 1.

Листинг 1 – Код программы задания 1

```
package Lab4;

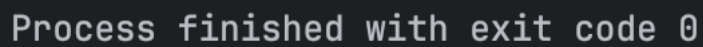
class Park {
    private class Attraction {
        private String name, time;
        private int price;
        public Attraction(String name, String time, int
price) {
            this.name = name;
            this.time = time;
            this.price = price;
        }
    }
    private Attraction[] attractions = {};
    public void addAttraction(String name, String time, int
price) {
        var t = new Attraction[this.attractions.length +
1];
        System.arraycopy(this.attractions, 0, t, 0,
attractions.length);
        t[t.length - 1] = new Attraction(name, time,
price);
    }
}
```

```

        this.attractions = t;
    }
}

public class Variant19 {
    public static void main(String[] args) {
        var park = new Park();
        park.addAttraction("abc", "10:00-11:00", 500);
    }
}

```



```

Process finished with exit code 0

```

Рисунок 1 – Результат задания 1

Задание 2. Создать класс Cinema (кино) с внутренним классом, с помощью объектов которого можно хранить информацию об адресах кинотеатров, фильмах и времени сеансов.

Код программы представлен в листинге 2, результат – на рисунке 2.

Листинг 2 – Код программы задания 2

```

package Lab4;

class Cinema {
    private class Information {
        private String address, time;
        private String[] movies;
        public Information(String address, String time,
String[] movies) {
            this.address = address;
            this.time = time;
            this.movies = movies;
        }
    }

    private Information[] informations = {};
}

```

```

        public void addInformation(String address, String time,
String[] movies) {
            var t = new Information[this.informations.length +
1];

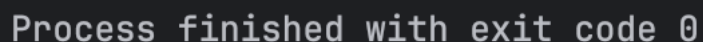
            System.arraycopy(this.informations, 0, t, 0,
informations.length);

            t[t.length - 1] = new Information(address, time,
movies);

            this.informations = t;
        }
    }

    public class Variant110 {
        public static void main(String[] args) {
            var cinema = new Cinema();
            cinema.addInformation("Moscow", "10:00-11:00", new
String[]{});
        }
    }
}

```



```

Process finished with exit code 0

```

Рисунок 2 – Результат задания 2

Задание 3. Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов interface Мебель <- abstract class Шкаф <- class Книжный Шкаф.

Код программы представлен в листинге 3, результат – на рисунке 3.

Листинг 3 – Код программы задания 3

```

package Lab4;

interface Furniture {
    public default void call() {}
}

```

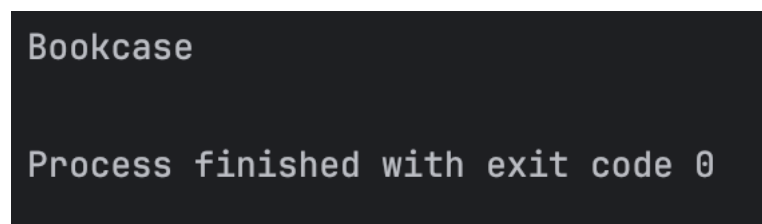
```

abstract class Wardrobe implements Furniture {
    @Override
    public void call() {
        System.out.println("Wardrobe");
    }
}

class Bookcase extends Wardrobe {
    @Override
    public void call() {
        System.out.println("Bookcase");
    }
}

public class Variant29 {
    public static void main(String[] args) {
        new Bookcase().call();
    }
}

```



```

Bookcase

Process finished with exit code 0

```

Рисунок 3 – Результат задания 3

Задание 4. Реализовать абстрактные классы или интерфейсы, а также наследование и полиморфизм для следующих классов interface Фильм <- class Отечественный Фильм <- class Комедия.

Код программы представлен в листинге 4, результат – на рисунке 4.

Листинг 4 – Код программы задания 4

```

package Lab4;

interface Movie {
    public default void call() {}
}

```

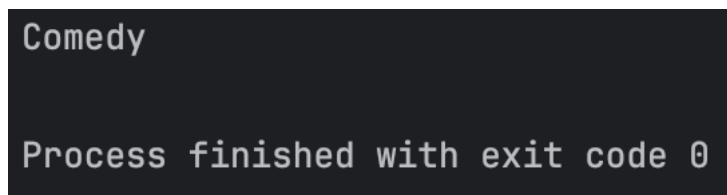
```

abstract class DomesticFilm implements Movie {
    @Override
    public void call() {
        System.out.println("DomesticFilm");
    }
}

class Comedy extends DomesticFilm {
    @Override
    public void call() {
        System.out.println("Comedy");
    }
}

public class Variant210 {
    public static void main(String[] args) {
        new Comedy().call();
    }
}

```



```

Comedy

Process finished with exit code 0

```

Рисунок 4 – Результат задания 4

Вывод

В ходе выполнения лабораторной работы была изучена работа внутренних классов и интерфейсов в языке программирования Java.