



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 13**

**Название:**                    Внутренние классы и интерфейсы в Java

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

Студент                    ИУ6-13М  
(Группа)

\_\_\_\_\_  
(Подпись, дата)                    Е.К. Матвиенко  
(И.О. Фамилия)

Преподаватель

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

Москва, 2021

**Цель лабораторной работы:** получение первичных навыков работы с внутренними классами и интерфейсами языка программирования Java.

## Ход работы:

### Задание №1:

1. Создать класс Mobile с внутренним классом, с помощью объектов которого можно хранить информацию о моделях телефонов и их свойствах.
2. Создать класс Художественная Выставка с внутренним классом, с помощью объектов которого можно хранить информацию о картинах, авторах и времени проведения выставок.

### Листинг выполнения подзадачи 1:

#### Класс Mobile:

```
package laba_4.task1.mobile;
import java.util.ArrayList;

public class Mobile {
    String firm;
    ArrayList<Model> models;

    public Mobile() {
        models = new ArrayList<>();
    }

    public Mobile(String firm) {
        this.firm = firm;
        models = new ArrayList<>();
    }

    public void addModel(String name){
        models.add(new Model(name));
    }

    @Override
    public String toString() {
        return "Mobile \n{" +
            "firm: '" + firm + '\'' +
            "\nmodels: \n" + models +
            '}';
    }

    class Model{
        String name;
        ArrayList<Attribute> attributes;

        public Model() {
            attributes = new ArrayList<>();
        }

        public Model(String name) {
            this.name = name;
            attributes = new ArrayList<>();
        }
    }
}
```

```

    }

    public void addAttribute(String name, int amount){
        Attribute attribute = new Attribute(name, amount);
        attributes.add(attribute);
    }

    public void addAttribute(String name){
        Attribute attribute = new Attribute(name);
        attributes.add(attribute);
    }

    @Override
    public String toString() {
        return "\n    Model{" +
            "name='" + name + '\'' +
            "\n        attributes=" + attributes + '\n' +
            "    }";
    }

    class Attribute{
        String name;
        int amount;

        public Attribute() {
        }

        public Attribute(String name) {
            this.name = name;
            this.amount = -1;
        }

        public Attribute(String name, int amount) {
            this.name = name;
            this.amount = amount;
        }

        @Override
        public String toString() {
            if(amount != -1) {
                return "Attribute{" +
                    "name='" + name + '\'' +
                    ", amount=" + amount +
                    '}';
            }
            else {
                return "Attribute{" +
                    "name='" + name +
                    '}';
            }
        }
    }
}

```

### Основная программа MainMobile:

```

package laba_4.task1.mobile;

//3. Создать класс Mobile с внутренним классом, с помощью объектов которого
// можно хранить информацию о моделях телефонов и их свойствах.
public class MainMobile {
    public static void main(String[] args) {

```

```

        Mobile mobile = new Mobile("Xiaomi");

        mobile.addModel("Mi 11");
        mobile.models.get(0).addAttribute("RAM", 8);
        mobile.models.get(0).addAttribute("ROM", 256);
        mobile.models.get(0).addAttribute("Accumulator", 4600);
        mobile.models.get(0).addAttribute("NFC");

        mobile.addModel("Mi 8");
        mobile.models.get(1).addAttribute("RAM", 6);
        mobile.models.get(1).addAttribute("ROM", 128);
        mobile.models.get(0).addAttribute("Accumulator", 3400);

        System.out.println(mobile);
    }
}

```

### Результат выполнения подзадачи 1:

```

C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Educational Edition 2021.3.2\lib\idea_rt.jar=55027:C:\Program Files\JetBrains\IntelliJ IDEA Educational Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\metel\IdeaProjects\my_lab_4\out\production\my_lab_4 laba_4.task1.mobile.MainMobile
Mobile
{firm: 'Xiaomi'
models:
[
  Model{name='Mi 11'
    attributes=[Attribute{name='RAM', amount=8}, Attribute{name='ROM', amount=256}, Attribute{name='Accumulator', amount=4600}, Attribute{name='NFC', amount=3400}]
  },
  Model{name='Mi 8'
    attributes=[Attribute{name='RAM', amount=6}, Attribute{name='ROM', amount=128}]
  }]
}
Process finished with exit code 0

```

### Листинг выполнения подзадачи 2:

#### Класс Exhibition:

```

package laba_4.task1.exhibition;

import java.sql.Time;
import java.time.Year;
import java.util.ArrayList;
import java.util.Date;

public class Exhibition {
    String name_of_exhibition;
    ArrayList<Picture> pictures;

    public Exhibition(String name, Date date){

```

```

        this.name_of_exhibition = name;
        pictures = new ArrayList<Picture>();
    }
    public void addPicture(String name, String author, Year year){pictures.add(new
Picture(name, author, year));
    }

    @Override
    public String toString() {
        return "Exhibition \n{" +
            "name: '" + name_of_exhibition + '\'' +
            "\npictures: \n" + pictures +
            '}';
    }

    class Picture{
        String p_name;
        String p_author;
        Year p_year;

        public Picture(){
        }
        public Picture(String name, String author, Year year){
            p_name = name;
            p_author = author;
            p_year = year;
        }
    }
}

```

## Основная программа MainExhibition:

```

package laba_4.task1.exhibition;

import java.time.Year;
import java.util.Date;

//4. Создать класс Художественная Выставка с внутренним классом,
// с помощью объектов которого можно хранить информацию о картинах,
// авторах и времени проведения выставок.
public class MainExhibition {
    public static void main(String[] args) {

        Date date = new Date(1212121212121L);

        Exhibition exhibition1 = new Exhibition("Aivazovsky", date);

        exhibition1.addPicture("Sea1","Aivazovsky_Exib", Year.of(1856));
        exhibition1.addPicture("Sea2","Aivazovsky", Year.of(1856));
        exhibition1.addPicture("Sea3","Aivazovsky", Year.of(1856));
        exhibition1.addPicture("Sea4","Aivazovsky", Year.of(1856));

        Exhibition exhibition2 = new Exhibition("VanGog_Exib", date);
        exhibition2.addPicture("Pict_1","VanGog", Year.of(1856));
        exhibition2.addPicture("Pict_2","Aivazovsky", Year.of(1856));
        exhibition2.addPicture("Pict_3","Aivazovsky", Year.of(1856));
        exhibition2.addPicture("Pict_4","Aivazovsky", Year.of(1856));

        System.out.println(exhibition1);
        System.out.println(exhibition2);
    }
}

```

## Результат выполнения подзадачи 2:

```
MainExhibition x
C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Educational Edition 2021.3.2\lib\idea_rt.jar=55065:C:\Program Files\JetBrains\IntelliJ IDEA Educational Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\metel\IdeaProjects\my_lab_4\out\production\my_lab_4 laba_4.task1.exhibition.MainExhibition
Exhibition
{name: 'Aivazovsky'
pictures:
[laba_4.task1.exhibition.Exhibition$Picture@7530d0a, laba_4.task1.exhibition.Exhibition$Picture@27bc2616, laba_4.task1.exhibition.Exhibition$Picture@3941a79c, laba_4.task1.exhibition.Exhibition$Picture@506e1b77]}
Exhibition
{name: 'VanGog_Exib'
pictures:
[laba_4.task1.exhibition.Exhibition$Picture@4fca772d, laba_4.task1.exhibition.Exhibition$Picture@9807454, laba_4.task1.exhibition.Exhibition$Picture@3d494fbf, laba_4.task1.exhibition.Exhibition$Picture@1ddc4ec2]}

Process finished with exit code 0
```

## Задание №2:

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

1. interface Сотрудник <- class Инженер <- class Руководитель.
2. interface Здание <- abstract class Общественное Здание <- class Театр.

## Листинг выполнения подзадачи 1:

### Интерфейс Employee:

```
package laba_4.task2.employee;

public interface Employee {
    String introduce();
}
```

### Класс Engineer:

```
package laba_4.task2.employee;

public class Engineer implements Employee{
    private String name;
    private String position;
    private String organization;
    private String department;

    public Engineer() {
    }

    public Engineer(String name, String position, String organization, String department) {
    }
}
```

```

        this.name = name;
        this.position = position;
        this.organization = organization;
        this.department = department;
    }

    @Override
    public String introduce() {
        return " \nThis is " + name + ",\nI am a " + position + " in " +
organization + " at " + department + " department";
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPosition() {
        return position;
    }

    public void setPosition(String position) {
        this.position = position;
    }

    public String getOrganization() {
        return organization;
    }

    public void setOrganization(String organization) {
        this.organization = organization;
    }

    public String getDepartment() {
        return department;
    }

    public void setDepartment(String department) {
        this.department = department;
    }

    @Override
    public String toString() {
        return "Engineer{" +
            "name='" + name + '\'' +
            ", position='" + position + '\'' +
            ", organization='" + organization + '\'' +
            ", department='" + department + '\'' +
            '}';
    }
}

```

### Класс Ruler:

```

package laba_4.task2.employee;

public class Ruler extends Engineer{

    public Ruler() {
    }
}

```

```

    public Ruler(String name, String position, String organization, String
department) {
        super(name, position, organization, department);
    }

    @Override
    public String introduce() {
        return super.introduce() + ", I am head of this department";
    }
}

```

### Основная программа MainEngineer:

```

package laba_4.task2.employee;

//3.    interface Сотрудник <- class Инженер <- class Руководитель.
public class MainEngineer {
    public static void main(String[] args) {

        Engineer engineer = new Engineer("Ivanov Ivan Ivanovich", "process engineer",
"NII CHAVO", "Magic");
        Ruler head = new Ruler("Petrov Petr Petrovich", "main engineer", "NII CHAVO",
"Magic");

        System.out.println(engineer.introduce());
        System.out.println();
        System.out.println(head.introduce());
    }
}

```

### Результат выполнения подзадачи 1:

```

Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath
C:\Users\metel\IdeaProjects\my_lab_4\out\production\my_lab_4 laba_4.task2.employee.MainEn

This is Ivanov Ivan Ivanovich,
I am a specialist in NII Skolkovo at AI department

This is Petrov Petr Petrovich,
I am a main engineer in NII Skolkovo at AI department, I am head of this department

Process finished with exit code 0
|

```



## Листинг выполнения подзадачи 2:

### Интерфейс Building:

```
package laba_4.task2.building;

public interface Building {
    void open();
    void close();
    String toString();
}
```

### Класс PublicBuilding:

```
package laba_4.task2.building;

abstract class PublicBuilding implements Building {
    protected int capacity;
    protected String name;

    PublicBuilding(String name) {
        this.name = name;
    }

    int getCapacity() {
        return this.capacity;
    }
    String getName(){
        return this.name;
    }
}
```

### Класс Theater:

```
package laba_4.task2.building;

public class Theater extends PublicBuilding {

    Theater(String name) {
        super(name);
    }

    @Override
    public void open() {
    }

    @Override
    public void close() {
    }
    @Override
    public String toString(){
        return "Teatr:" + name;
    }
}
```

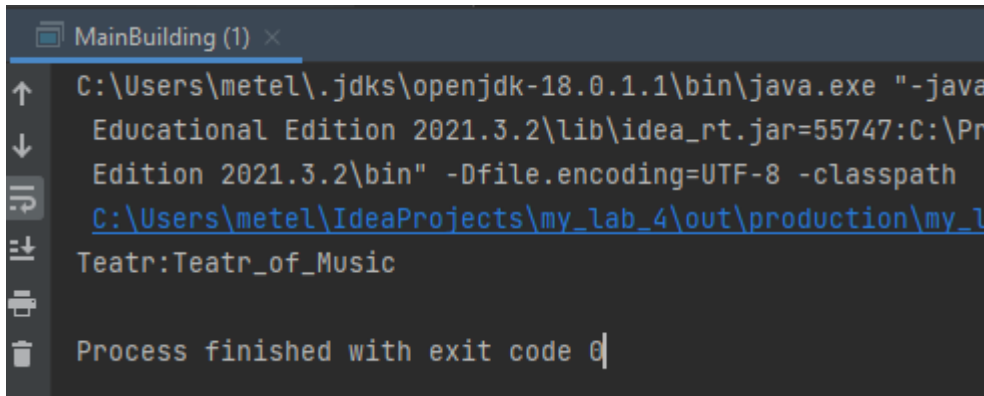
## Основная программа MainBuilding:

```
package laba_4.task2.building;
//4.    interface Здание <- abstract class Общественное Здание <- class Театр.

public class MainBuilding {
    public static void main(String[] args) {

        Theater my_teatr  = new Theater("Teatr_of_Music");
        System.out.println(my_teatr);
    }
}
```

## Результат выполнения подзадачи 2:



```
MainBuilding (1) x
C:\Users\metel\.jdk\openjdk-18.0.1.1\bin\java.exe "-javaa
Educational Edition 2021.3.2\lib\idea_rt.jar=55747:C:\Pro
Edition 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath
C:\Users\metel\IdeaProjects\my_lab_4\out\production\my_la
Teatr:Teatr_of_Music
Process finished with exit code 0
```

Программное решение представлено в репозитории распределённой системы управления версиями Git:

[https://github.com/matvilen/BigDataLanguages/tree/main/my\\_lab\\_4/src/laba\\_4](https://github.com/matvilen/BigDataLanguages/tree/main/my_lab_4/src/laba_4)

**Вывод:** при выполнении лабораторной работы были получены навыки работы с внутренними классами и интерфейсами языка программирования Java.