



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 10**

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

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

Студент

ИУ6-23М

(Группа)

\_\_\_\_\_  
(Подпись, дата)

И.С. Коломенский

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

## Цель работы:

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

## Выполнение:

### Задание 1:

1. Создать класс City (город) с внутренним классом, с помощью объектов которого можно хранить информацию о проспектах, улицах, площадях.
2. Создать класс Cinema (кино) с внутренним классом, с помощью объектов которого можно хранить информацию об адресах кинотеатров, фильмах и времени сеансов.

### Задание 2:

1. Абстрактный класс Книга (Шифр, Автор, Название, Год, Издательство).  
Подклассы Справочник и Энциклопедия.
2. interface Фильм <- class Отечественный Фильм <- class Комедия.

Листинг файла Book.java

```
package com.company;

public abstract class Book {
    private String shifr;
    private String author;
    private String name;
    private int yr;
    private String izd;

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

    public void setAuthor(String author) {
        this.author = author;
    }

    public void setIzd(String izd) {
        this.izd = izd;
    }

    public void setYr(int yr) {
        this.yr = yr;
    }

    public void setShifr(String shifr) {
        this.shifr = shifr;
    }
}
```

```

//Получение значений

public String getName() {
    return name;
}

public int getYr() {
    return yr;
}

public String getAuthor() {
    return author;
}

public String getIzd() {
    return izd;
}

public String getShifr() {
    return shifr;
}
//Метод вывода абстрактно
public abstract void OutDate();
}

```

#### Листинг файла City.java

```

package com.company;
//Реализуем класс Город
public class City {
    private double area;
    private int population;
    private String name;//Название города
    String []streets;//Названия улиц
    int cntStr;//Количество улиц
    int cntAvn;//Количество проспектов
    int cntArs;//Количество площадей

    private class Avenues
    {
        private int width;
        private int length;
        private String name;
        public Avenues()
        {

        }

        public Avenues(int w,int l)
        {
            this.length=l;
            this.width=w;
        }

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

        public void setWidth(int width) {
            this.width = width;
        }

        public void setLength(int length) {
            this.length = length;
        }

        public int getLength() {
            return length;
        }
    }
}

```

```

    }

    public int getWidth() {
        return width;
    }

    @Override
    public String toString() {
        return String.format("Проспект: %20s | %5d | %5d",name,width,length);
    }
}

private class Areas
{
    private String type;//Тип площади: вокзальные, предзаводские и др.
    private String name;//Наименование
    public Areas()
    {
        this.type="торговые";//По умолчанию
        this.name="Y";
    }
    public Areas(String nam,String tp)
    {
        this.name=nam;
        this.type=tp;
    }
    public void setType(String type) {
        this.type = type;
    }

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

    @Override
    public String toString() {
        return String.format("Площадь : %15s | %15s",name,type);
    }
}

private Avenues[] avs;//проспекты
private Areas[]ars;//Площади
public City(int cntStreet,int cntAvns,int cntAres)
{
    this.cntArs=0;
    this.cntAvn=0;
    this.cntStr=0;
    //@cntStreet-количество улиц
    //@cntAvns-количество проспектов в городе
    this.avs=new Avenues[cntAvns+5];
    this.population=10000;//мин для статуса города
    this.area=1230;//
    this.streets=new String[cntStreet+1];
    this.ars=new Areas[cntAres+1];
}

public void AddStreet(String strname)
{
    this.streets[cntStr++]=strname;
}
//добавление площадей
public void AddAreas(String nm,String type)
{
    if(cntArs<ars.length)
    {
        this.ars[cntArs]=new Areas();
        this.ars[cntArs].setName(nm);
    }
}

```

```

        this.ars[cntArs++].setType(type);
    }
}
//добавление проспектов
public void AddAvenues(String nm,int w,int l)
{
    //Добавим все
    //
    if(cntAvn<avs.length)
    {
        this.avs[cntAvn]=new Avenues();
        this.avs[cntAvn].setName(nm);
        this.avs[cntAvn].setLength(l);
        this.avs[cntAvn++].setWidth(w);
    }
}

public void DisplayStreets()
{
    System.out.println("*****УЛИЦЫ*****");
    for(int i=0;i<cntStr;i++)
    {
        System.out.println(streets[i]);
    }
}

public void DisplayAreas()
{
    System.out.println("*****Площади*****");
    for(int i=0;i<cntArs;i++)
    {
        System.out.println(ars[i]);
    }
}

public void DisplayPros()
{
    System.out.println("*****УЛИЦЫ*****");
    for(int i=0;i<cntAvn;i++)
    {
        System.out.println(avs[i]);
    }
}

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

public String getName() {
    return name;
}

public void setArea(double area) {
    this.area = area;
}

public void setPopulation(int population) {
    this.population = population;
}

@Override
public String toString() {
    return String.format("Название: %20s  Население(К-во): %9d  Площадь:
%f",name,population,area);
}
}

```

### Листинг файла Ensiklopdia.java

```
package com.company;

public class Ensiklopdia extends Book{
    private int cntPage;//Количество страниц
    @Override
    public void OutDate() {
        System.out.println(String.format("Энциклопедия: %15s  Шифр:%7s  Год:%5d
Назначено: %s  Издательство: " +
        "%15s  Количество страниц:
%5d",getShifr(),getShifr(),getYr(),getName(),getIzd(),getCntPage()));
    }
    public Ensiklopdia()
    {
        //Пустой конструктор
    }
    public Ensiklopdia(String name)
    {
        //@name-Наименование
        this.setName(name);
    }
    public int getCntPage() {
        return cntPage;
    }

    public void setCntPage(int cntPage) {
        this.cntPage = cntPage;
    }
}
```

### Листинг файла Guide.java

```
package com.company;

public class Guide extends Book{
    private String type;//Онлайн или бумажный

    @Override
    public void OutDate() {
        System.out.println(String.format("Справочник: %15s | Тип: %12s| Шифр:%7s
Год:%5d  Назначено: %s  Издательство: " +
        "%15s",type,getShifr(),getShifr(),getYr(),getName(),getIzd()));
    }

    Guide(String nm)
    {
        //@nm-Наименование справочника
        this.setName(nm);
    }

    public void setType(String type) {
        this.type = type;
    }

    public String getType() {
        return type;
    }
}
```

### Листинг файла Main1.java

```
package com.company;

public class Main1 {
```

```

public static void main(String[] args) {
    City ct=new City(5,5,2);
    ct.setArea(125444);
    ct.setName("Moscow");
    ct.setPopulation(12000000);
    ct.AddAreas("Kreml","Общий");
    ct.AddAvenues("Centre",12,2000);
    ct.AddStreet("Арбат");
    System.out.println(ct);
    ct.DisplayAreas();
    ct.DisplayPros();
    ct.DisplayStreets();
}
}

```

Листинг файла Main2.java

```

package com.company;

public class Main2 {

    public static void main(String[] args) {
        Ensiklopdia en=new Ensiklopdia("Термины");
        en.setCntPage(12543);
        en.setAuthor("Симон.К.Р");
        en.setIzd("Санкт Петербург");
        en.setShifr("DC-125");
        en.setYr(1997);
        en.OutDate();
        Guide gd=new Guide("Турист");
        gd.setName("Разговорный краткий ");
        gd.setShifr("AB-124");
        gd.setYr(2005);
        gd.setAuthor("Фаронов.");
        gd.OutDate();
    }
}

```

```

Название:          Moscow  Население(К-во):  12000000  Площадь: 125444,000000
*****Площади*****
Площадь :          Kreml |          Общий
*****Улицы*****
Проспект:          Centre |    12 |    2000
*****Улицы*****
Арбат

```

Рисунок 1 - Результат выполнения кода решения задачи 1

Энциклопедия:	DC-125	Шифр: DC-125	Год: 1997	Назначено: Термины	Издательство: Санкт Петербург	Количество страниц: 12543
Справочник:	null	Тип: AB-124	Шифр: AB-124	Год: 2005	Назначено: Разговорный краткий	Издательство: null

Рисунок 2 - Результат выполнения кода решения задачи 2

### **Ссылка на программное решение:**

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Wingo11/BigDataLanguages/tree/Lab4/src>

### **Вывод:**

При выполнении лабораторной работы были получены навыки работы с внутренними классами и интерфейсами языка программирования Java.