



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения

**ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3
по дисциплине
«Программирование на языке Java»**

Тема: Наследование в Java

Выполнил студент группы ИКБО-16-20

Пак С.А.

Принял ассистент кафедры ИиППО

Русляков А.А.

Практические работы выполнены «_____» 2021г.

«Зачтено» «_____» 2021г

Москва 2021

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ.....	3
1. Механизм наследования.....	3
2. Абстрактные классы.....	3
РЕШЕНИЕ ЗАДАЧИ.....	4
1. Постановка задачи.....	4
2. Программный код.....	4
3. Вывод программы.....	9
ВЫВОД.....	11

ТЕОРЕТИЧЕСКОЕ ВВЕДЕНИЕ

1. Механизм наследования

С помощью наследования можно расширить функционал уже имеющихся классов за счет добавления нового функционала или изменения старого.

Чтобы объявить один класс наследником от другого, надо использовать после имени класса-наследника ключевое слово `extends`, после которого идет имя базового класса.

Способность к изменению функциональности, унаследованной от базового класса, называется полиморфизмом и является одним из ключевых аспектов объектно-ориентированного программирования наряду с наследованием и инкапсуляцией.

С помощью ключевого слова `super` мы можем обратиться к любому члену базового класса - методу или полю, если они не определены с модификатором `private`.

Хотя наследование очень интересный и эффективный механизм, но в некоторых ситуациях его применение может быть нежелательным. И в этом случае можно запретить наследование с помощью ключевого слова `final`.

Кроме запрета наследования можно также запретить переопределение отдельных методов.

2. Абстрактные классы

Кроме обычных классов в Java есть абстрактные классы. Абстрактный класс похож на обычный класс. В абстрактном классе также можно определить поля и методы, в то же время нельзя создать объект или экземпляр абстрактного класса. Абстрактные классы призваны предоставлять базовый функционал для классов-наследников. А производные классы уже реализуют этот функционал.

При определении абстрактных классов используется ключевое слово `abstract`.

Кроме обычных методов абстрактный класс может содержать абстрактные методы. Такие методы определяются с помощью ключевого слова `abstract` и не имеют никакого функционала.

Производный класс обязан переопределить и реализовать все абстрактные методы, которые имеются в базовом абстрактном классе. Также следует учитывать, что если класс имеет хотя бы один абстрактный метод, то данный класс должен быть определен как абстрактный.

РЕШЕНИЕ ЗАДАЧИ

1. Постановка задачи

Вариант: 3

Задание: создать класс, описывающий мебель. С помощью наследования реализовать различные виды мебели. Также создать класс FurnitureShop, моделирующий магазин мебели. Протестировать работу классов.

2. Программный код

Файл Furniture.java, содержащий абстрактный класс:

```
package ru.mirea.classes;

public abstract class Furniture {
    protected String name;           // название мебели
    protected String brand;          // фирма мебели
    protected double price;          // цена мебели

    /**
     * Конструктор по умолчанию
     */
    public Furniture() {
        this.name = "";
        this.brand = "";
        this.price = 0.0;
    }

    /**
     * Конструктор не по умолчанию
     * @param name      название мебели
     * @param brand     фирма, производившая мебель
     * @param price     цена мебели
     */
    public Furniture(String name, String brand, double price) {
        this.name = name;
        this.brand = brand;
        this.price = price;
    }

    /**
     * Геттер для поля name
     * @return          название мебели
     */
    public String getName() {
        return this.name;
    }
}
```

```

}

/**
 * Геттер для поля brand
 * @return      фирма, производившая мебель
 */
public String getBrand() {
    return this.brand;
}

/**
 * Геттер для поля price
 * @return      цена мебели
 */
public double getPrice() {
    return this.price;
}

/**
 * Сеттер для поля name
 * @param name      новое название мебели
 */
public void setName(String name) {
    this.name = name;
}

/**
 * Сеттер для поля brand
 * @param brand      новая фирма, производившая мебель
 */
public void setBrand(String brand) {
    this.brand = brand;
}

/**
 * Сеттер для поля price
 * @param price      новая цена мебели
 */
public void setPrice(double price) {
    this.price = price;
}

/**
 * Объединяет всю информацию об объекте в одну строку
 * @return      строка с информацией об объекте
 */
@Override

```

```

    public abstract String toString();
}

```

Файл Table.java:

```

package ru.mirea.classes;

public class Table extends Furniture {
    protected double width;           // ширина стола
    protected double length;          // длина стола
    protected double height;          // высота стола

    /**
     * Конструктор по умолчанию
     */
    public Table() {
        super();
        this.width = 0.0;
        this.length = 0.0;
        this.height = 0.0;
    }

    /**
     * Конструктор не по умолчанию
     * @param name        название мебели
     * @param brand       фирма, производившая мебель
     * @param price       цена мебели
     * @param width       ширина стола
     * @param length      длина стола
     * @param height      высота стола
     */
    public Table(String name,
                  String brand,
                  double price,
                  double width,
                  double length,
                  double height
    ) {
        super(name, brand, price);
        this.width = width;
        this.length = length;
        this.height = height;
    }

    /**
     * Геттер для поля width
     * @return            ширина стола
     */
}

```

```

public double getWidth() {
    return this.width;
}

/**
 * Геттер для поля length
 * @return        длина стола
 */
public double getLength() {
    return this.length;
}

/**
 * Геттер для поля height
 * @return        высота стола
 */
public double getHeight() {
    return this.height;
}

/**
 * Сеттер для поля width
 * @param width    новая ширина стола
 */
public void setWidth(double width) {
    this.width = width;
}

/**
 * Сеттер для поля length
 * @param length    новая длина стола
 */
public void setLength(double length) {
    this.length = length;
}

/**
 * Сеттер для поля height
 * @param height    новая высота стола
 */
public void setHeight(double height) {
    this.height = height;
}

/**
 * Объединяет всю информацию об объекте в одну строку
 * @return        строка с информацией об объекте

```

```

    */
    @Override
    public String toString() {
        return ("Table {\n"
            + "\tname: " + this.name + "\n"
            + "\tbrand: " + this.brand + "\n"
            + "\tprice: " + this.price + "\n"
            + "\twidth: " + this.width + "\n"
            + "\tlength: " + this.length + "\n"
            + "\theight: " + this.height + "\n"
            + "}");
    }
}

```

Файл Chair.java:

```

package ru.mirea.classes;

public class Chair extends Table {
    /**
     * Конструктор по умолчанию
     */
    public Chair() {
        super();
    }

    /**
     * Конструктор не по умолчанию
     * @param name      название мебели
     * @param brand      фирма, производившая мебель
     * @param price      цена мебели
     * @param width      ширина стола
     * @param length     длина стола
     * @param height     высота стола
     */
    public Chair(String name,
                  String brand,
                  double price,
                  double width,
                  double length,
                  double height
    ) {
        super(name, brand, price, width, length, height);
    }
    /**
     * Объединяет всю информацию об объекте в одну строку

```



```

    * @return          строка с информацией об объекте
    */
    @Override
    public String toString() {
        return ("Chair {\n"
            + "\tname: " + this.name + "\n"
            + "\tbrand: " + this.brand + "\n"
            + "\tprice: " + this.price + "\n"
            + "\twidth: " + this.width + "\n"
            + "\tlength: " + this.length + "\n"
            + "\theight: " + this.height + "\n"
            + "}")
    };
}
}

```

Файл FurnitureShop.java:

```

package ru.mirea;

import ru.mirea.classes.*;

public class FurnitureShop {
    public static void main(String[] args) {
        Furniture[] goods = new Furniture[5];

        goods[0] = new Table("Table #1", "Famous Furniture", 12.02,
5.0, 2.0, 3.0);
        goods[1] = new Table("Table #2", "Infamous Furniture", 5.123,
5.0, 2.0, 3.0);
        goods[2] = new Table("Table #3", "Notorious Furniture", 2.3,
4.0, 2.5, 1.0);
        goods[3] = new Chair("Chair #1", "Good Furniture", 2.3, 4.0,
2.5, 1.0);
        goods[4] = new Chair("Chair #2", "Bad Furniture", 0.2235, 4.0,
2.5, 1.0);

        System.out.println("Список доступной мебели:");

        for (Furniture good : goods) {
            System.out.println(good);
            System.out.println();
        }
    }
}

```

3. Вывод программы

Программа выводит информацию о всей доступной мебели (рис.1).

```
31 Список доступной мебели:
30 Table {
29     name: Table #1
28     brand: Famous Furniture
27     price: 12.02
26     width: 5.0
25     length: 2.0
24     height: 3.0
23 }
22
21 Table {
20     name: Table #2
19     brand: Infamous Furniture
18     price: 5.123
17     width: 5.0
16     length: 2.0
15     height: 3.0
14 }
13
12 Table {
11     name: Table #3
10     brand: Notorious Furniture
9     price: 2.3
8     width: 4.0
7     length: 2.5
6     height: 1.0
5 }
4
3 Chair {
2     name: Chair #1
1     brand: Good Furniture
33     price: 2.3
1     width: 4.0
2     length: 2.5
3     height: 1.0
4 }
5
6 Chair {
7     name: Chair #2
8     brand: Bad Furniture
9     price: 0.2235
10     width: 4.0
11     length: 2.5
12     height: 1.0
13 }
```

2.5, 1.0);

System.out.println("C

for (Furniture good :

System.out.println

System.out.println

}

}

}

3. Вывод программы

Программа выводит ин

Рис.1 Вы

Рис.1 Выходные данные программы ч.1

ВЫВОД

В ходе выполнения работы изучил понятие наследования, и научился реализовывать наследование в Java.