|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Отчет по выполнению лабораторной работы 3

**Тема. НАСЛЕДОВАНИЕ В JAVA**

Дисциплина Программирование на языке Джава

Выполнил

|  |  |
| --- | --- |
| студент | Болотов Михаил |
|  | Фамилия И.О. |
| группа | ИКБО-06-19 |
|  | Номер группы |

Москва 2020

**Содержание**

[​ Теория 2](#__RefHeading___Toc4128_811265792)

[​ Абстрактные классы 2](#__RefHeading___Toc27521_811265792)

[​ Задание 2](#__RefHeading___Toc4136_811265792)

[​ Код 2](#__RefHeading___Toc4138_811265792)

[​ Plate.java 2](#__RefHeading___Toc27523_811265792)

[​ Bowl.java 4](#__RefHeading___Toc27525_811265792)

[​ Dish.java 5](#__RefHeading___Toc27527_811265792)

[​ DishesTest.java 7](#__RefHeading___Toc27529_811265792)

[​ Скриншот 8](#__RefHeading___Toc3748_811265792)

[​ Заключение 8](#__RefHeading___Toc3750_811265792)

[​ Библиографический список 8](#__RefHeading___Toc3752_811265792)

# Теория

Одним из ключевых аспектов объектно-ориентированного программирования является наследование. С помощью наследования можно расширить функционал уже имеющихся классов за счет добавления нового функционала или изменения старого.

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

### 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

# Задание

Создать абстрактный класс, описывающий посуду(Dish). С помощью наследования реализовать различные виды посуды, имеющие свои свойства и методы. Протестировать работу классов.

# Код

### Plate.java

package dev.ky3he4ik.lab.lab3;  
  
public class Plate extends Dish {  
 protected double capacity;  
 protected double currCapacity = 0;  
  
 public Plate(double capacity, double cost, double durability, boolean isClean) {  
 super(cost, durability, isClean);  
 this.capacity = capacity;  
 }  
  
 public Plate(double cost, double durability, boolean isClean) {  
 super(cost, durability, isClean);  
 }  
  
 public Plate(double capacity, double cost, double durability) {  
 super(cost, durability);  
 this.capacity = capacity;  
 }  
  
 public Plate(double capacity, double cost) {  
 super(cost);  
 this.capacity = capacity;  
 }  
  
 public Plate() {  
 super();  
 capacity = 0;  
 }  
  
 @Override  
 void fill() {  
 isClean = false;  
 currCapacity = capacity;  
 }  
  
 @Override  
 public void setClean(boolean clean) {  
 if (clean)  
 currCapacity = 0;  
 super.setClean(clean);  
 }  
  
 public double getCurrCapacity() {  
 return currCapacity;  
 }  
  
 @Override  
 public String toString() {  
 return "Plate " + durability + "/" + durability + "; cost: " + cost + "$; capacity: " + capacity + "ml " + (isClean ? "clean" : "dirty");  
 }  
  
 public double getCapacity() {  
 return capacity;  
 }  
  
 public void setCapacity(double capacity) {  
 this.capacity = capacity;  
 }  
}

### Bowl.java

package dev.ky3he4ik.lab.lab3;  
  
public class Bowl extends Dish {  
 protected String pattern;  
  
 public Bowl(String pattern, double cost, double durability, boolean isClean) {  
 super(cost, durability, isClean);  
 this.pattern = pattern;  
 }  
  
 public Bowl(String pattern, double cost, double durability) {  
 super(cost, durability);  
 this.pattern = pattern;  
 }  
  
 public Bowl(double cost, double durability, boolean isClean) {  
 super(cost, durability, isClean);  
 pattern = "None";  
 }  
  
 public Bowl(String pattern, double cost) {  
 super(cost);  
 this.pattern = pattern;  
 }  
  
 public Bowl(double durability, double cost) {  
 super(cost, durability);  
 pattern = "None";  
 }  
  
 public Bowl(String pattern) {  
 this.pattern = pattern;  
 }  
  
 public Bowl(double cost) {  
 super(cost);  
 pattern = "None";  
 }  
  
 public Bowl() {  
 pattern = "None";  
 }  
  
 @Override  
 void fill() {  
 isClean = false;  
 }  
  
 @Override  
 public String toString() {  
 return "Bowl " + durability + "/" + durability + "; cost: " + cost + "$; pattern: " + pattern + " " + (isClean ? "clean" : "dirty");  
 }  
  
 public String getPattern() {  
 return pattern;  
 }  
  
 public void setPattern(String pattern) {  
 this.pattern = pattern;  
 }  
}

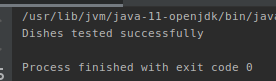
### Dish.java

package dev.ky3he4ik.lab.lab3;  
  
abstract public class Dish {  
 protected double cost;  
 protected double durability;  
 protected boolean isClean;  
  
 public Dish(double cost, double durability, boolean isClean) {  
 this.cost = cost;  
 this.durability = durability;  
 this.isClean = isClean;  
 }  
  
 public Dish(double cost, double durability) {  
 this.cost = cost;  
 this.durability = durability;  
 isClean = true;  
 }  
  
 public Dish(double cost) {  
 this.cost = cost;  
 durability = 0;  
 isClean = true;  
 }  
  
 public Dish() {  
 cost = 0;  
 durability = 0;  
 isClean = true;  
 }  
  
 public double getCost() {  
 return cost;  
 }  
  
 public void setCost(double cost) {  
 this.cost = cost;  
 }  
  
 public double getDurability() {  
 return durability;  
 }  
  
 public void setDurability(double durability) {  
 this.durability = durability;  
 }  
  
 public boolean isClean() {  
 return isClean;  
 }  
  
 public void setClean(boolean clean) {  
 isClean = clean;  
 }  
  
 abstract void fill();  
  
 @Override  
 public abstract String toString();  
}

### DishesTest.java

package dev.ky3he4ik.lab.lab3;  
  
public class DishesTest {  
 public static void main(String[] args) {  
 Plate plate = new Plate(123, 321, 12, false);  
 assert plate.getCapacity() == 123;  
 assert plate.getCost() == 321;  
 assert plate.getDurability() == 12;  
 assert !plate.isClean();  
 plate = new Plate();  
 assert plate.getCapacity() == 0;  
 assert plate.getCost() == 0;  
 assert plate.getDurability() == 0;  
 assert plate.isClean();  
 plate.setCapacity(1);  
 plate.setCost(2);  
 plate.setDurability(3);  
 plate.setClean(false);  
 assert plate.getCapacity() == 1;  
 assert plate.getCost() == 2;  
 assert plate.getDurability() == 3;  
 assert !plate.isClean();  
 plate.setClean(true);  
 plate.fill();  
 assert plate.getCurrCapacity() == plate.getCapacity();  
 assert !plate.isClean();  
  
 Bowl bowl = new Bowl("Simple", 321, 12, false);  
 assert bowl.getPattern().equals("Simple");  
 assert bowl.getCost() == 321;  
 assert bowl.getDurability() == 12;  
 assert !bowl.isClean();  
 bowl = new Bowl();  
 assert bowl.getPattern().equals("None");  
 assert bowl.getCost() == 0;  
 assert bowl.getDurability() == 0;  
 assert bowl.isClean();  
 bowl.setPattern("Golden");  
 bowl.setCost(654);  
 bowl.setDurability(8);  
 bowl.setClean(false);  
 assert bowl.getPattern().equals("Golden");  
 assert bowl.getCost() == 654;  
 assert bowl.getDurability() == 8;  
 assert !bowl.isClean();  
 bowl.setClean(true);  
 bowl.fill();  
 assert !bowl.isClean();  
 System.out.println("Dishes tested successfully");  
 }  
}

# Скриншот



# Заключение

В данной лабораторной работе я изучил понятие наследования, и научился реализовывать наследование в Java.

# Библиографический список

1. Зорина Н.В. Курс лекций по Объектно-ориентированному программированию на Java, МИРЭА, Москва, 2016
2. Программирование на языке Java: работа со строками и массивами. Методические указания. [Электронный ресурс] : Учебно-методические пособия — Электрон. дан. — СПб. : ПГУПС, 2015. — 24 с.
3. Кожомбердиева, Г.И. Программирование на языке Java: создание графического интерфейса пользователя: учеб. пособие. [Электронный ресурс] : Учебные пособия / Г.И. Кожомбердиева, М.И. Гарина. — Электрон. дан. — СПб.: ПГУПС, 2012. — 67 с.
4. Вишневская, Т.И. Технология программирования. Часть 1. [Электронный ресурс] / Т.И. Вишневская, Т.Н. Романова. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана, 2007. — 59 с.