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

**Институт информационных технологий (ИИТ)  
Кафедра практической и прикладной информатики (ППИ)**

Практическая работа №3

по дисциплине «[Объектно-ориентированное программирование](http://yandex.ru/clck/jsredir?bu=gs6i37&from=yandex.ru%3Bsearch%2F%3Bweb%3B%3B&text=&etext=6946.LbXym-ahv1pxKVAz2hf9KlPqloebKlVw9wBB3G8HAdh5PoOG0StODMkJQO482epD_olHQL90zn-w5Jrk3WGB1D6o_zb0MlHSftMRmGe0ftvwZFFvu4h0utYc2bksZLTirdN7ivKg2UvFfBdzr_MiLg.225438a9e0fc3da19a2297477b9d44806d5db05b&uuid=&state=Em5uB10Ym2yYXpZKRFvY8hpXT7l4NK6-neJyELJlZHT1RbEWUe0bjcCJCVwA0EtoD_aV8NmsLDPQb0sjTYpqyA,,&&cst=AiuY0DBWFJ5fN_r-AEszk_pjsh0oFwbchANSowSMJTd94w6KOKKASi2FDJAwHZv__5hnlmYgPEw-RXyaZDHVbvNlTqt_DPZKj1nzLl3L-nyK7oOQ-mw04R8LXLvDh8z5n4Vf7BOWWVEcZxJXo-8b-NdvgAKmDbis3i1pI1HyZsZqN2BydLhjT9BezS_6u_FoJQMPBMiT-XPOXzedNseYFRmdiJoOjkxcw5rJAtFMRob9PjGqqf0OJfP83yoJU0XUw485esCub9EW1el_SnyoaResArcqoA32qb_IqJIipTVKhFONyWbM_oUfsfiAl65MhyBk6-I5ortKcVgTC1DuvlbpUFtt4wYA0xhL430X08FdXdnQFqAPCOlzt7D3ZLRqD-71p3ZTImlFbXw5hkjRuA,,&data=,,&sign=4a202b0b42a2108bcf97d9d5458997cf&keyno=0&b64e=2&ref=orjY4mGPRjk5boDnW0uvlrrd71vZw9kpVBUyA8nmgRFSi5A7LMz1rfhmsIkfEwsgTY6Qc268FgLkUV0iLuYantaan9uaa1IJPLaVaduct_w3CiR6gnEGgMejWaHvVuFlhyNOxYhAc5-6gygT0o2jPisdzaanw9nbl4eAgPxrQK9VWUF1oIWCpYudFSS3pLuvt4DP_MVhi3d-nElSH8NXENYbBGWApp-AzIFQQDc_YUnmjY-6w03-XYFXg4Eo6tTcp4nMQFFqYj5kWjWD5SBqtIIjlLa_IPKEE98Le0FVym8M4dVMLGJT77pjLvKL1CXEcL4cbnwK4DPDQJgS55qE7bfk_EXINy9Fo-bZJAzKqi)»

«НАСЛЕДОВАНИЕ. АБСТРАКТНЫЕ СУПЕРКЛАССЫ И ИХ ПОДКЛАССЫ В JAVA.»

Выполнила студентка группы ИНБО-04-18 Ефремова С.Г.

Принял доцент Алпатов А.Н.

Москва 2019

# Введение

Основной задачей является получение практических навыков разработки программ и освоить на практике работу с абстрактными классами и наследованием на Java.

Задание: Перепишите суперкласс Shape и его подклассы, так как это представлено на рисунке 1 Circle, Rectangle and Square.



Рисунок 1 Диаграмма суперкласса Shape

# Основная часть

Класс, содержащий абстрактные методы, называется абстрактным классом. Такие классы при определении помечаются ключевым словом abstract.

Абстрактный метод внутри абстрактного класса не имеет тела, только прототип. Он состоит только из объявления и не имеет тела. По сути, мы создаём шаблон метода.

Если вы объявляете класс, производный от абстрактного класса, но хотите иметь возможность создания объектов нового типа, вам придётся предоставить определения для всех абстрактных методов базового класса.

Если этого не сделать, производный класс тоже останется абстрактным, и компилятор заставит пометить новый класс ключевым словом abstract.

Абстрактный класс не может содержать какие-либо объекты, а также абстрактные конструкторы и абстрактные статические методы. Любой подкласс абстрактного класса должен либо реализовать все абстрактные методы суперкласса, либо сам быть объявлен абстрактным.

Ход работы

Для работы программы создаются 5 файлов: Main.java, Circle.java, Rectangle.java, Square.java и Shape.java. В Main.java создаются объекты класса для тестирования, а в остальных файлах – класса с диаграммы.

Пример работы программы представлен на рис.2

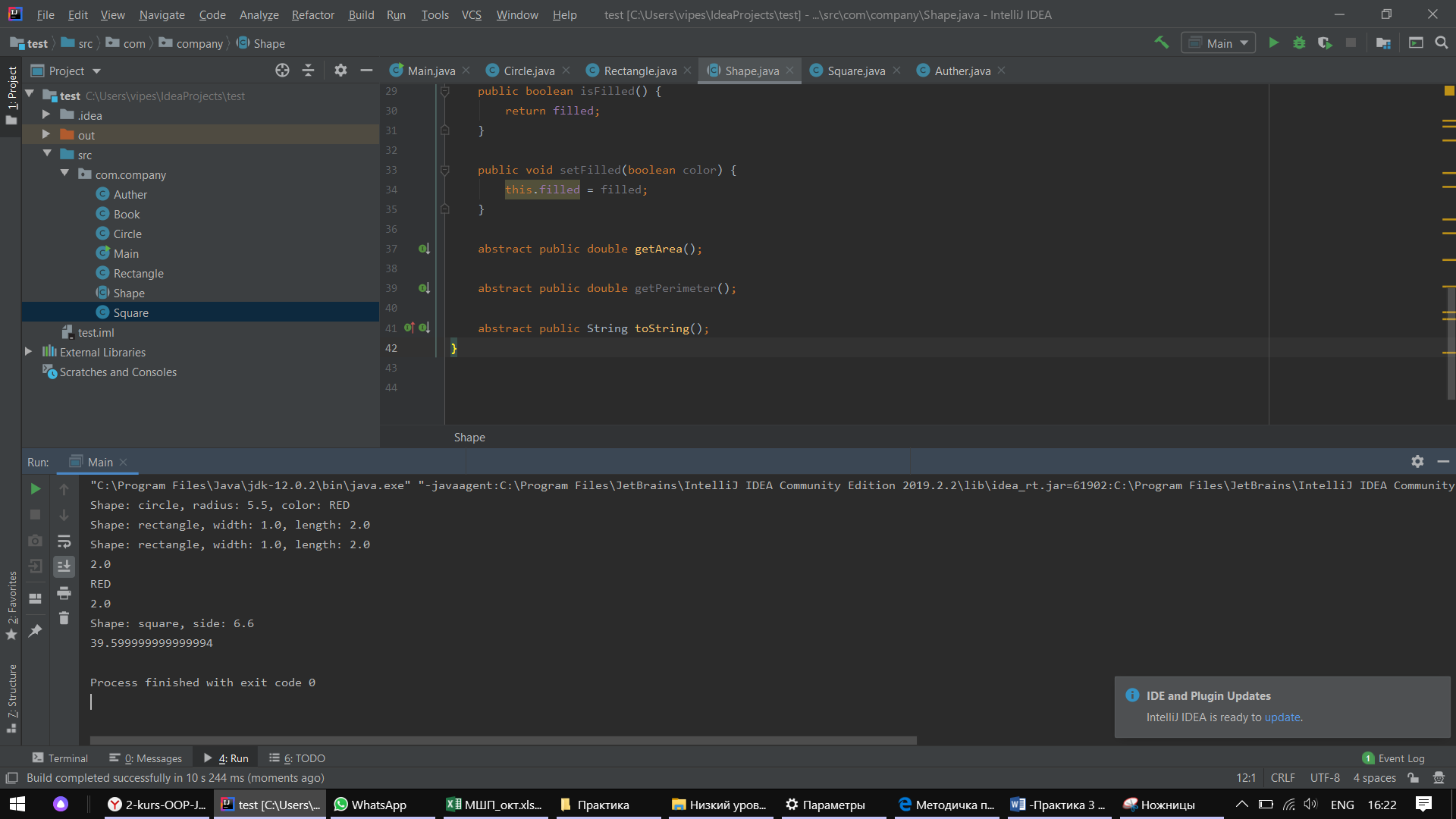


Рисунок 2 Пример работы программы

Код программы

Square.java:

|  |
| --- |
| package com.company; |
|  |  |
|  | public class Square extends Rectangle{ |
|  | protected double side; |
|  | public Square() { |
|  | this.filled = false; |
|  | this.color = "blue"; |
|  | } |
|  | public Square(double side) { |
|  | this.filled = false; |
|  | this.color = "blue"; |
|  | this.side = side; |
|  | } |
|  | public Square(double side, String color, boolean filled) { |
|  | this.filled = filled; |
|  | this.color = color; |
|  | this.side = side; |
|  | } |
|  | public double getSide() { |
|  | return side; |
|  | } |
|  | public void setSide(double side) { |
|  | this.side = side; |
|  | } |
|  | @Override |
|  | public double getArea() { |
|  | return 6\*side; |
|  | } |
|  |  |
|  | @Override |
|  | public double getPerimeter() { |
|  | return 12 \* side; |
|  | } |
|  |  |
|  | @Override |
|  | public String toString() { |
|  | return "Shape: square, side: " + this.side + ""; |
|  | } |
|  | } |

Shape.java:

|  |
| --- |
|  |
| package com.company; |
|  | |  | |
|  | abstract class Shape { | |
|  | protected String color; | |
|  | protected boolean filled; | |
|  | public Shape(){ | |
|  | color="blue"; | |
|  | filled=true; | |
|  | }; | |
|  | public Shape(String color, boolean filled){ | |
|  | this.color=color; | |
|  | this.filled=filled; | |
|  | }; | |
|  | public String getColor() { | |
|  | return color; | |
|  | } | |
|  | public void setColor(String color) { | |
|  | this.color = color; | |
|  | } | |
|  | public boolean isFilled() { | |
|  | return filled; | |
|  | } | |
|  | public void setFilled(boolean color) { | |
|  | this.filled = filled; | |
|  | } | |
|  | abstract public double getArea(); | |
|  | abstract public double getPerimeter(); | |
|  | abstract public String toString();} | |
|  |  | |

Rectangle.java:

|  |
| --- |
| package com.company; |
|  |  |
|  | public class Rectangle extends Shape { |
|  | protected double width; |
|  | protected double length; |
|  | public Rectangle() { |
|  | this.filled = false; |
|  | this.color = "blue"; |
|  | width = 2; |
|  | length=4; |
|  | } |
|  |  |
|  | public Rectangle(double width, double length) { |
|  | this.filled = false; |
|  | this.color = "blue"; |
|  | this.width = width; |
|  | this.length=length; |
|  | } |
|  | public Rectangle(double width, double length, String color, boolean filled) { |
|  | this.filled = filled; |
|  | this.color = color; |
|  | this.width = width; |
|  | this.length=length; |
|  | } |
|  | public double getWidth() { |
|  | return width; |
|  | } |
|  | public void setWidth(double width) { |
|  | this.width = width; |
|  | } |
|  | public double getLength() { |
|  | return length; |
|  | } |
|  | public void setLength(double length) { |
|  | this.length = length; |
|  | } |
|  | @Override |
|  | public double getArea() { |
|  | return width\*length; |
|  | } |
|  |  |
|  | @Override |
|  | public double getPerimeter() { |
|  | return 2 \* width \* length; |
|  | } |
|  |  |
|  | @Override |
|  | public String toString() { |
|  | return "Shape: rectangle, width: " + this.width + ", length: "+this.length; |
|  | } |
|  | } |

Circle.java:

|  |
| --- |
| package com.company; |
|  |  |
|  | public class Circle extends Shape { |
|  | protected double radius; |
|  |  |
|  | public Circle() { |
|  | this.filled = false; |
|  | this.color = "blue"; |
|  | radius = 1; |
|  | } |
|  |  |
|  | public Circle(double radius) { |
|  | this.filled = false; |
|  | this.color = "blue"; |
|  | this.radius = radius; |
|  | } |
|  |  |
|  | public Circle(double radius, String color, boolean filled) { |
|  | this.radius = radius; |
|  | this.color = color; |
|  | this.filled = filled; |
|  | } |
|  |
|  | public double getRadius() { |
|  | return radius; |
|  | } |
|  |  |
|  | public void setRadius(double radius) { |
|  | this.radius = radius; |
|  | } |
|  |  |
|  | @Override |
|  | public double getArea() { |
|  | return Math.PI \* radius \* radius; |
|  | } |
|  |  |
|  | @Override |
|  | public double getPerimeter() { |
|  | return 2 \* Math.PI \* radius; |
|  | } |
|  |  |
|  | @Override |
|  | public String toString() { |
|  | return "Shape: circle, radius: " + this.radius + ", color: "+this.color; |
|  | } |
|  | } |

Main.java:

|  |
| --- |
| package com.company; |
|  |  |
|  | public class Main { |
|  |  |
|  | public static void main(String[] args) { |
|  | Shape s1 = new Circle(5.5, "RED", false); |
|  | System.out.println(s1.toString()); |
|  | Shape s2 = new Rectangle(1.0, 2.0, "RED", false); |
|  | System.out.println(s2.toString()); |
|  | Rectangle r1 = (Rectangle)s2; |
|  | System.out.println(r1); |
|  | System.out.println(r1.getArea()); |
|  | System.out.println(r1.getColor()); |
|  | System.out.println(r1.getLength()); |
|  | Shape s4 = new Square(6.6); |
|  | System.out.println(s4); |
|  | System.out.println(s4.getArea()); |
|  | }} |

# Вывод

В ходе выполнения лабораторных работ были получены практические навыки разработки программ и освоена на практике работа с абстрактными классами и наследованием на Java в программе IDE IntelliJ IDEA.

# Список используемых источников

1. Начинающим Java программистам (эл. ресурс) URL: https://habr.com/ru/post/43293/, (02.09.2019)
2. Программное обеспечение Java (эл. ресурс) URL: https://www.oracle.com/ru/java/, (02.09.2019)
3. Где используется Java и зачем нужна? (эл. ресурс) URL: https://javarush.ru/groups/posts/1079-gde-ispoljhzuetsja-java, (02.09.2019)
4. Руководство по языку программирования Java (эл. ресурс) URL: https://metanit.com/java/tutorial/, (02.09.2019)