

МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего образования*

***«МИРЭА – Российский технологический университет»***

**РТУ МИРЭА**

Отчет по выполнению практического задания №2

**Тема: “**Абстрактные суперклассы и его подклассы в Java”

Дисциплина Программирование на языке Джава

Выполнил студент Скребнев Алексей Николаевич

Группа ИНБО-02-20

**Москва 2021**

Оглавление

[Постановка задачи 3](#_Toc82418126)

[Реализация задания 4](#_Toc82418127)

[Вывод 9](#_Toc82418128)

# Постановка задачи

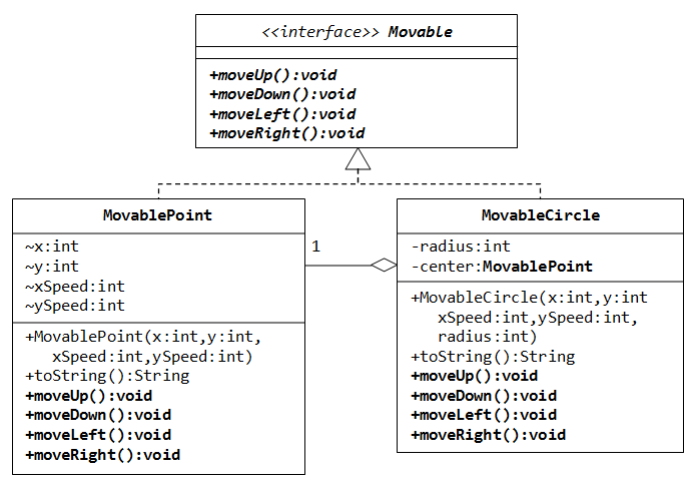
Цель работы: освоить на практике работу с абстрактными классами и наследованием на Java.

Задание 1. Абстрактный суперкласс Shape и его подклассы.

Задание перепишите суперкласс Shape и его подклассы так как это представлено на диаграмме Circle, Rectangle and Square.



Задание 2. Вам нужно написать два класса MovablePoint и MovableCircle – которые реализуют интерфейс Movable.



# Реализация задания

Код выполнен в среде IntelliJ IDEA 2021.2.1.

TestShape.java

package pract3;

import java.lang.\*;

import java.util.Scanner;

public class TestShape {

public static void main(String[] args){

Scanner in = new Scanner(System.in);

System.out.println("Введите цвет, правдивость заполненности и радиус");

Shape s1 = new Circle(in.nextLine(),in.nextBoolean(),in.nextDouble());

System.out.println(s1); // which version?

System.out.println(s1.getArea()); // which version?

System.out.println(s1.getPerimeter()); // which version?

Shape s2 = new Rectangle(1.0,1.5);

System.out.println(s2);

System.out.println(s2.getArea());

System.out.println(s2.getPerimeter());

Scanner in2 = new Scanner(System.in);

System.out.println("Введите цвет и правдивость заполненности");

Shape s3 = new Squere(3.0,in2.nextLine(),in2.nextBoolean());

System.out.println(s3);

System.out.println(s3.getArea());

System.out.println(s3.getPerimeter());

System.out.println("Введите позиции по х и у и скорости по данным координатам");

Movable move = new MovablePoint(in.nextInt(),in.nextInt(),in.nextInt(),in.nextInt());

System.out.println(move);

move.moveLeft();

move.moveUp();

System.out.println(move);

System.out.println("Введите позиции по х и у, скорости по данным координатам и радиус");

Movable moveCircle = new MovableCircle(in.nextInt(),in.nextInt(),in.nextInt(),in.nextInt(),in.nextInt());

System.out.println(moveCircle);

moveCircle.moveUp();

moveCircle.moveRight();

System.out.println(moveCircle);

}

}

Shape.java

package pract3;

public abstract class Shape {

public double pi = 3.14;

protected String color;

protected boolean filled;

public Shape(){

this.filled = false;

this.color = "default";

}

public Shape(String color, boolean filled){

this.color = color;

this.filled = filled;

}

public String getColor(){

return this.color;

}

public boolean getFilled(){

return this.filled;

}

public void setColor(String color){

this.color = color;

}

public void setFilled(boolean filled){

this.filled=filled;

}

public abstract double getArea();

public abstract double getPerimeter();

@Override

public abstract String toString();

}

Circle.java

package pract3;

public class Circle extends Shape{

protected double radius;

@Override

public double getArea(){

double res = pi\*radius\*radius;

return res;

}

@Override

public double getPerimeter(){

double res = 2\*pi\*radius;

return res;

}

@Override

public String toString(){

return ("Color of Circle is " + this.color + " , radius is " + this.radius + " and filled is " + this.filled );

}

public Circle(){

super();

}

public Circle(String color, boolean filled, double radius){

super(color,filled);

this.radius=radius;

}

public Circle(double radius){

super();

this.radius=radius;

}

public double getRadius(){

return this.radius;

}

public void setRadius(double radius){

this.radius=radius;

}

}

Rectangle.java

package pract3;

public class Rectangle extends Shape{

protected double width;

protected double length;

public Rectangle(){

super();

}

public Rectangle(double width,double length){

this.length = length;

this.width = width;

}

public Rectangle(double width,double length, String color,boolean filled){

super(color,filled);

this.width=width;

this.length=length;

}

public double getWidth(){

return this.width;

}

public double getLength(){

return this.length;

}

public void setWidth(double width){

this.width = width;

}

public void setLength(double length){

this.length = length;

}

@Override

public double getArea(){

double res = width\*length;

return res;

}

@Override

public double getPerimeter(){

double res = 2\*(width+length);

return res;

}

@Override

public String toString(){

return ("Color of Rectangular is " + this.color + " width is " + this.width + " length is " + this.length + " and filled is " + this.filled);

}

}

Squere.java

package pract3;  
  
public class Squere extends Rectangle{  
 public Squere(){  
 super();  
 }  
 public Squere(double side){  
 super(side,side);  
 }  
 public Squere(double side, String color, boolean filled){  
 super(side,side,color,filled);  
 }  
 public double getSide(){  
 return this.length;  
 }  
 public void setSide(double side){  
 this.length = side;  
 this.width = side;  
 }  
 @Override  
 public String toString(){  
 return ("Сторона квадрата side:" + this.width + " цвет квадрата - " + this.color + " filled is " + this.filled);  
 }  
}

Movable.java

package pract3;  
  
public interface Movable {  
 public void moveUp();  
 public void moveDown();  
 public void moveLeft();  
 public void moveRight();  
}

MovablePoint.java

package pract3;  
  
public class MovablePoint implements Movable {  
 protected int x;  
 protected int y;  
 protected int xSpeed;  
 protected int ySpeed;  
  
 MovablePoint(int x,int y,int xSpeed,int ySpeed){  
 this.x = x;  
 this.xSpeed = xSpeed;  
 this.y = y;  
 this.ySpeed = ySpeed;  
 }  
  
 public void moveUp(){  
 this.y +=ySpeed;  
 }  
  
 public void moveDown(){  
 this.y -=ySpeed;  
 }  
  
 public void moveLeft(){  
 this.x -=xSpeed;  
 }  
  
 public void moveRight(){  
 this.x +=xSpeed;  
 }  
  
 @Override  
 public String toString(){  
 return ("Позиция точки x:" + this.x + ", y:" + this.y + "Скорость по x:" + this.xSpeed + ". Скорость по y:" + this.ySpeed);  
 }  
}

MovableCircle.java

package pract3;  
  
public class MovableCircle implements Movable{  
 private int radius;  
 private MovablePoint center;  
  
 public MovableCircle(int x,int y,int xSpeed,int ySpeed,int radius){  
 this.radius = radius;  
 center = new MovablePoint(x,y,xSpeed,ySpeed);  
 }  
  
 @Override  
 public void moveDown() {  
 this.center.y -= this.center.ySpeed;  
 }  
  
 @Override  
 public void moveUp() {  
 this.center.y += this.center.ySpeed;  
 }  
  
 @Override  
 public void moveLeft() {  
 this.center.x -= this.center.xSpeed;  
 }  
  
 @Override  
 public void moveRight() {  
 this.center.x += this.center.xSpeed;  
 }  
 @Override  
 public String toString(){  
 return (this.center.toString()+ " радиус круга = " + this.radius);  
 }  
}

# Вывод

В данной работе я освоил на практике работу с абстрактными классами и наследованием на Java.