**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 2**

по дисциплине:

«Визуальные средства разработки программных приложений»

на тему:

«Наследование в *Java*»

Выполнил: студент гр. ИТП-31

Расшивалов Н.И.

Принял: ассистент

Михалевич В.Г.

Гомель 2021

**Цель работы:** изучить механизм наследования *Java*.

**Задание:** Вариант 8. Условия задания на рисунках 1 – 2.

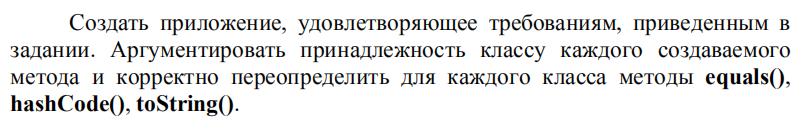


Рисунок 1 – Условие задания



Рисунок 2 – Вариант задания

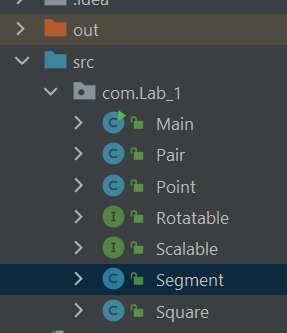


Рисунок 3 – Структура созданных классов

Результат выполнения задания представлен на рисунке 4.

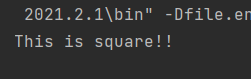


Рисунок 4 – Результат выполнения задания

**Вывод:** Были написаны абстрактные классы и их реализации соответственно поставленному заданию. Были написаны аксессоры и переопределения наследуемых методов.

**ПРИЛОЖЕНИЕ А**

Листинг созданных классов

package com.Lab\_1;  
  
public class Pair<T1, T2> {  
 private T1 first;  
 private T2 second;  
  
 public Pair(T1 first, T2 second) {  
 this.first = first;  
 this.second = second;  
 }  
  
 public T1 getFirst() {  
 return first;  
 }  
  
 public T2 getSecond() {  
 return second;  
 }  
}

package com.Lab\_1;  
  
public class Point implements Rotatable {  
 private Pair<Double, Double> coordinates;  
  
 public Point(double x, double y) {  
 this.coordinates = new Pair<>(x, y);  
 }  
  
 public double getX() {  
 return coordinates.getFirst();  
 }  
  
 public double getY() {  
 return coordinates.getSecond();  
 }  
  
 /\*  
 \* NOTE: point has no size, therefore it cannot be scaled. Only it's coordinates can.  
 \* \*/  
 public void scaleCoordinates(double factor) {  
 double x = coordinates.getFirst();  
 double y = coordinates.getSecond();  
  
 double newX = x \* factor;  
 double newY = y \* factor;  
  
 coordinates = new Pair<>(newX, newY);  
 }  
  
 @Override  
 public void rotate(double angle) {  
 double x = coordinates.getFirst();  
 double y = coordinates.getSecond();  
 double newX = x \* Math.cos(angle) - y \* Math.sin(angle);  
 double newY = x \* Math.sin(angle) + y \* Math.cos(angle);  
 coordinates = new Pair<>(newX, newY);  
 }  
}

package com.Lab\_1;  
  
public interface Rotatable {  
 void rotate(double angle);  
}

public interface Scalable {  
 void scale(double factor);  
}

package com.Lab\_1;  
  
public class Segment implements Rotatable, Scalable {  
 private Pair<Point, Point> ends;  
  
 public Segment(double x1, double y1, double x2, double y2) {  
 ends = new Pair<>(new Point(x1, y1), new Point(x2, y2));  
 }  
  
 public double length() {  
 double x1 = ends.getFirst().getX();  
 double y1 = ends.getFirst().getY();  
 double x2 = ends.getSecond().getX();  
 double y2 = ends.getSecond().getY();  
  
 return Math.sqrt((x2 - x1) \* (x2 - x1) + (y2 - y1) \* (y2 - y1));  
 }  
  
 @Override  
 public void rotate(double angle) {  
 ends.getFirst().rotate(angle);  
 ends.getSecond().rotate(angle);  
 }  
  
 @Override  
 public void scale(double factor) {  
 ends.getFirst().scaleCoordinates(factor);  
 ends.getSecond().scaleCoordinates(factor);  
 }  
}

package com.Lab\_1;  
  
public class Square implements Rotatable, Scalable {  
 private Pair<Segment, Segment> sides;  
  
 public Square(double x1, double y1, double x2, double y2, double x3, double y3) {  
 Segment side1 = new Segment(x1, y1, x2, y2);  
 Segment side2 = new Segment(x2, y2, x3, y3);  
 sides = new Pair<>(side1, side2);  
 }  
  
 @Override  
 public void rotate(double angle) {  
 sides.getFirst().rotate(angle);  
 sides.getSecond().rotate(angle);  
 }  
  
 @Override  
 public void scale(double factor) {  
 sides.getFirst().scale(factor);  
 sides.getSecond().scale(factor);  
 }  
}

package com.Lab\_1;  
public class Main {  
  
 public static void main(String[] args) {  
 Segment segment1 = new Segment(1,1,1,5);  
 Segment segment2 = new Segment(1,5,5,5);  
 Segment segment3 = new Segment(5,5,5,1);  
 Segment segment4 = new Segment(5,1,1,1);  
  
 if (segment1.length() == segment2.length() & segment2.length() == segment3.length() &  
 segment3.length() == segment4.length() & segment4.length() == segment1.length()){  
 System.out.println("This is square!!");  
 }else {  
 System.out.println("This is no square!!");  
 }  
  
 Square square = new Square(1,1,1,5,5,5);  
  
 }  
}