## 实验2：类与对象的使用（1）

### 1.实验目的

（1）掌握定义类的方法。

（2）掌握对象的声明和创建方法。

### 2.实验内容

（1）定义一个Point类，有成员变量x和y坐标，有重载的构造方法和计算到原点的距离的方法。再定义一个Line类，该类包含两个Point类型的实例变量，用以表示一条线段的两个端点，还有以下方法：重载的构造方法；计算线段长度方法；判断线段是否水平方法；判断是否为垂直线段的方法；计算线段的斜率方法；计算线段的中点方法。最后定义测试类测试Line类。

1.　源代码：

import java.util.\*;

import java.lang.\*;

class Point {

public double x;

public double y;

public Point(double x\_arg, double y\_arg) {

x = x\_arg;

y = y\_arg;

}

public Point() {

this(0, 0);

}

public double cal\_distence() {

return Math.sqrt(x \* x + y \* y);

}

}

class Line {

private Point PointX;

private Point PointY;

public Line(double x\_x, double x\_y, double y\_x, double y\_y) {

PointX = new Point(x\_x, x\_y);

PointY = new Point(y\_x, y\_y);

}

public Line() {

this(0, 0, 0, 0);

}

/\* 计算线段长度方法 \*/

public double length() {

return Math.sqrt(Math.pow((PointX.x - PointY.x), 2) + Math.pow((PointX.y - PointY.y), 2));

}

/\* 判断线段是否水平方法 \*/

public boolean shuiping() {

if (PointX.y == PointY.y)

return true;

else

return false;

}

/\* 判断是否为垂直线段的方法 \*/

public boolean chuizhi() {

if (PointX.x == PointY.x)

return true;

else

return false;

}

/\* 计算线段的斜率方法(默认排除斜率不存在的情况) \*/

public double xielv() {

return (PointX.y - PointY.y) / (PointX.x - PointY.x);

}

/\* 计算线段的中点方法 \*/

public String zhongdian() {

return "[" + (PointX.x + PointY.x) / 2 + "," + (PointX.y + PointY.y) / 2 + "]";

}

}

public class Main {

public static void main(String[] args) {

Line t1 = new Line(2, 3, 1, 1);

System.out.println("坐标点为（2,3）,（1,1）的线段的距离长是：" + t1.length());

System.out.println("坐标点为（2,3）,（1,1）的线段水平吗? ：" + t1.shuiping());

System.out.println("坐标点为（2,3）,（1,1）的线段垂直吗? ：" + t1.chuizhi());

System.out.println("坐标点为（2,3）,（1,1）的线段斜率是 ：" + t1.xielv());

System.out.println("坐标点为（2,3）,（1,1）的线段中点是 ：" + t1.zhongdian());

Line t2 = new Line(6, 3, 9, 3);

System.out.println("坐标点为（6,3）,（9,3）的线段水平吗? ：" + t2.shuiping());

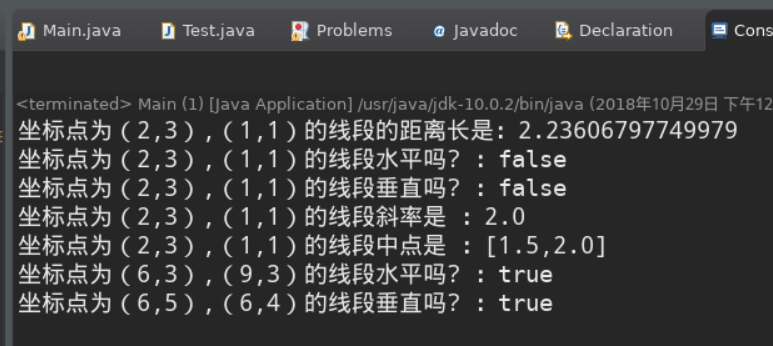
Line t3 = new Line(6, 5, 6, 4);

System.out.println("坐标点为（6,5）,（6,4）的线段垂直吗? ：" + t3.chuizhi());

}

}

1. 运行截图：



1. 定义一个分数类，它包括分子、分母、构造方法、计算分数相乘方法和显示分数的方法。再定义测试类，创建2个分数对象，计算并输出它们的乘积。
2. 源代码：

class fraction {

private int x;

private int y;

public fraction(int x\_arg, int y\_arg) {

x = x\_arg;

y = y\_arg;

}

public fraction() {

this(0, 0);

}

public fraction fra\_mul(fraction rfs) {

this.x \*= rfs.x;

this.y \*= rfs.y;

return this;

}

public void prin() {

System.out.println(x + "/" + y);

}

}

public class Test {

public static void main(String[] args) {

fraction t1 = new fraction(2, 3);

fraction t2 = new fraction(1, 3);

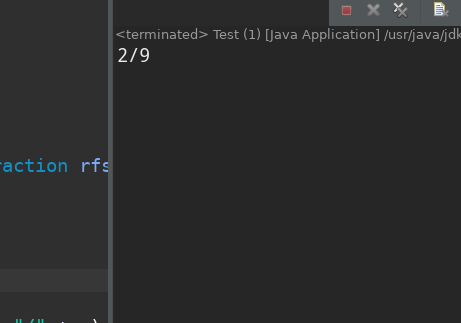
t1.fra\_mul(t2);

t1.prin();

}

}

1. 运行截图：



### 实验总结

这次实验主要练习了类的定义和方法的定义及使用，基本都比较矮简单．重点难点在数学公式的使用，囧～．希望以后能多使用点java的特性或者是工具类．