第二次上机作业报告

姓名：汪义波

2.创建一个 NewRectangle 类

（1)double 类型的成员变量 width，height；

```
private double width;
private double height;
```

（2）默认构造方法；

```
public NewRectangle(){
        width=0.0;
        height=0.0;
      }
```

（3）带两个参数的构造方法；

```
public NewRectangle(double w,double h){
          width=w;
          height=h;
      }
```

（4）成员方法 getArea()返回面积；

```
public double getArea(){
          return (width*height);
      }
```

（5）成员方法 getPerimeter()返回周长；

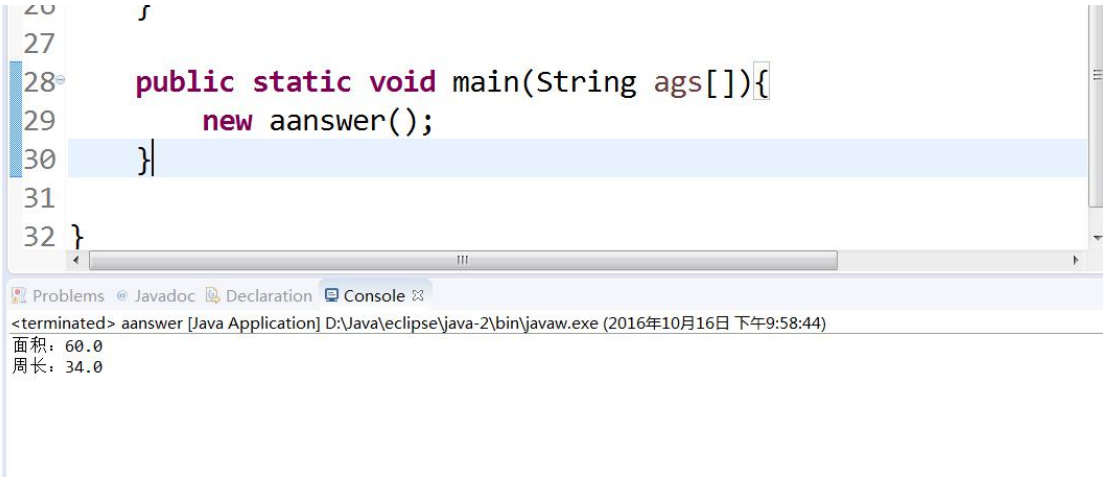
```
public double getPerimeter(){
          return (2*width+2*height);
      }
```

源代码：

```
public class aanswer {
    class NewRectangle{
        private double width;
        private double height;
        public NewRectangle(){
          width=0.0;
          height=0.0;
        }
        public NewRectangle(double w,double h){
            width=w;
            height=h;
        }
        public double getArea(){
```

```
            return (width*height);
        }
        public double getPerimeter(){
            return (2*width+2*height);
        }
    }
    public aanswer(){
        NewRectangle a= new NewRectangle(5.0,12.0);
        System.out.println("面积："+a.getArea()+"\n"+"周长："+a.getPerimeter());
    }
    public static void main(String ags[]){
        new aanswer();
    }
}
```

运行截图：



3.在第二题的基础上，完成如下程序：

（1）定义 point 类：

```
public class Point{
        private double x;
        private double y;
        public Point(){
          x=0.0;
          y=0.0;
        }
        public Point(double a,double b){
            x=a;
            y=b;
        }
        public double distance(Point p){
            double a=this.x-p.x;
            double b=this.y-p.y;
```

```
            return (Math.sqrt(a*a+b*b));
        }
    }
```

（2）修改第二题中的 NewRectangle 类，加入一个 point 类成员，代表矩形左下顶点的坐标；

```
Point p;
```

（3）修改 NewRFectangle 类，添加新的构造方法：

```
 public NewRectangle(){
            p=new Point();
            width=0.0;
            height=0.0;
            p.x=0.0;
            p.y=0.0;
          }
 public NewRectangle(double w,double h,double a,double b){
            p= new Point();
            width=w;
            height=h;
            p.x=a;
            p.y=b;
          }
```

（4）添加 boolean bPoint(Point p)，判断 p 是否在矩形内；

```
 boolean bPointIn(Point q){
if(this.p.x<=q.x&&q.x<=(this.p.x+this.width)&&this.p.y<=q.y&&q.y<=(this.p.y+th
is.height)){
        return true;
        }
else return false;
}
```

（5）添加新的方法，判断一个矩形是否包含在另一个矩形内，判断两个矩形是否有重叠部分；

```
boolean ifIn(NewRectangle q){
    Point w=new Point(q.p.x,q.p.y);
    Point r=new Point((q.p.x+q.width),(q.p.y+q.height));
    if(bPointIn(w)==true&&bPointIn(r)==true){
      return true;
       }
    else return false;
    }
boolean ifcover(NewRectangle q){
//根据矩形中心点判断
```

```
if(Math.abs(this.p.x+this.width/2-q.p.x-q.width/2)<(this.width/2+q.width/2)&&M
ath.abs(this.p.y+this.height/2-q.p.y-q.height/2)<(this.height/2+q.height/2))
     return true;
     else return false;
}
```

源代码：

```
public class modanswer {
     class Point{
            private double x;
            private double y;
            public Point(){
              x=0.0;
              y=0.0;
            }
            public Point(double a,double b){
                x=a;
                y=b;
            }
            public double distance(Point p){
                double a=this.x-p.x;
                double b=this.y-p.y;
                return (Math.sqrt(a*a+b*b));
            }
        }
     class NewRectangle{
            private double width;
            private double height;
            public Point p;
            public NewRectangle(){
              p=new Point();
              width=0.0;
              height=0.0;
              p.x=0.0;
              p.y=0.0;
            }
            public NewRectangle(double w,double h,double a,double b){
                p=new Point();
                width=w;
                height=h;
                p.x=a;
                p.y=b;
            }
```

```
            boolean bPointIn(Point q){

    if(this.p.x<=q.x&&q.x<=(this.p.x+this.width)&&this.p.y<=q.y&&q.y<=(this.p.
y+this.height)){
                    return true;
                }
                else return false;
            }
            boolean ifIn(NewRectangle q){
                Point w=new Point(q.p.x,q.p.y);
                Point r=new Point((q.p.x+q.width),(q.p.y+q.height));
                if(bPointIn(w)==true&&bPointIn(r)==true){
                return true;
                }
                else return false;
            }
            boolean ifcover(NewRectangle q){
                //根据矩形中心点判断

    if(Math.abs(this.p.x+this.width/2-q.p.x-q.width/2)<(this.width/2+q.width/2)
&&Math.abs(this.p.y+this.height/2-q.p.y-q.height/2)<(this.height/2+q.height/2))
                  return true;
                else return false;
            }
            public double getArea(){
                return (width*height);
            }
            public double getPerimeter(){
                return (2*width+2*height);
            }
        }
     public modanswer(){
         NewRectangle R=new NewRectangle(6,4,1,1);
         NewRectangle S=new NewRectangle(5,3,1,1);
         NewRectangle T=new NewRectangle(7,3,-2,-2);
         NewRectangle U=new NewRectangle(7,3,-1,-1);
         System.out.println(R.ifIn(S)+"\n");
         System.out.println(S.ifIn(R)+"\n");
         System.out.println(R.ifcover(S)+"\n");
         System.out.println(S.ifcover(R)+"\n");
         System.out.println(R.ifcover(T)+"\n");
         System.out.println(S.ifcover(T)+"\n");
         System.out.println(S.ifcover(U)+"\n");
     }
```

```
    public static void main(String ags[]){
        new modanswer();
    }
}
```

运行截图：

```
65      public modanswer(){
66          NewRectangle R=new NewRectangle(6,4,1,1);
67          NewRectangle S=new NewRectangle(5,3,1,1);
68          NewRectangle T=new NewRectangle(7,3,-2,-2);
69          NewRectangle U=new NewRectangle(7,3,-1,-1);
70          System.out.println(R.ifIn(S)+"\n");
71          System.out.println(S.ifIn(R)+"\n");
72          System.out.println(R.ifcover(S)+"\n");
73          System.out.println(S.ifcover(R)+"\n");
74          System.out.println(R.ifcover(T)+"\n");
75          System.out.println(S.ifcover(T)+"\n");
76          System.out.println(S.ifcover(U)+"\n");
77      }
```

Problems Javadoc Declaration Console

```
<terminated> modanswer [Java Application] D:\Java\eclipse\java-2\bin\javaw.exe (2016年10月16日 下午10:58:46)
true

false

true

true

false

false

true
```

Writable Smart Insert 11 : 7

9. 创建一个父类 Cycle，再创建三个自类 Unicycle,Bicycle，Tricycle；

（1）在 Cycle 类中定义 ride()方法，是的三个子类实例都能通过该方法向上转型为 Cycle 类；

```
 public class Cycle{
        public Cycle(){

        }
        public void ride(Cycle c){

        }
    }
    public class Unicycle extends Cycle{

    }
    public class Bicycle extends Cycle{
```

```
    }
    public class Tricycle extends Cycle{

    }
```

（2）加入 wheel（）类，返回车轮数，在子类中重写 wheel()类，返回车轮数，并判断多态性；
答：不同子类转换成 Cycle 类后 wheel()结果还是原来重写的轮子数，体现了多态性。

```
public class Cycle{
    public Cycle(){

    }
    public void ride(Cycle c){
        System.out.println(""+c.wheel());
    }
    public int wheel(){
        return 0;
    }
   }
   public class Unicycle extends Cycle{
     public int wheel(){
         return 1;
     }
   }
   public class Bicycle extends Cycle{
     public int wheel(){
         return 2;
     }
   }
   public class Tricycle extends Cycle{
     public int wheel(){
         return 3;
     }
   }
```

（3）在 Unicycle 和 Bicycle 中添加 balance，在 Tricycle 中不添加，在 ride 中调用 balance 方法，并用 instanceof 保证向下转型不会出错；

```
public class Cycle{
        public Cycle(){

        }
        public void ride(Cycle c){
            System.out.println(""+c.wheel());
```

```
            if(c instanceof Unicycle==true){
                ((Unicycle)c).balance();
            }
            if(c instanceof Bicycle==true){
                ((Bicycle)c).balance();
            }
        }
        public int wheel(){
            return 0;
        }
       }
       public class Unicycle extends Cycle{
         public int wheel(){
             return 1;
         }
         public void balance(){
             System.out.println("它不平衡");
         }
       }
       public class Bicycle extends Cycle{
         public int wheel(){
             return 2;
         }
         public void balance(){
             System.out.println("它不平衡");
         }
       }
       public class Tricycle extends Cycle{
         public int wheel(){
             return 3;
         }
       }
```

源代码：

```
public class ganswer {
    class Cycle{
        public Cycle(){

        }
        public void ride(Cycle c){
            System.out.println(""+c.wheel());
            if(c instanceof Unicycle==true){
                ((Unicycle)c).balance();
            }
            if(c instanceof Bicycle==true){
```

```
                ((Bicycle)c).balance();
            }
        }
        public int wheel(){
            return 0;
        }
       }
       public class Unicycle extends Cycle{
        public int wheel(){
            return 1;
        }
        public void balance(){
            System.out.println("它不平衡");
        }
       }
       public class Bicycle extends Cycle{
        public int wheel(){
            return 2;
        }
        public void balance(){
            System.out.println("它不平衡");
        }
       }
       public class Tricycle extends Cycle{
        public int wheel(){
            return 3;
        }
       }
      public ganswer(){
        Cycle a=new Cycle();
        a.ride(a);
        Unicycle b=new Unicycle();
        b.ride(b);
        Bicycle c=new Bicycle();
        c.ride(c);
        Tricycle d=new Tricycle();
        d.ride(d);
      }
      public static void main(String ags[]){
        new ganswer();
      }
}
```

运行截图：

```
37          public int wheel(){
38              return 3;
39          }
40      }
41  public ganswer(){
42      Cycle a=new Cycle();
43      a.ride(a);
44      Unicycle b=new Unicycle();
45      b.ride(b);
46      Bicycle c=new Bicycle();
47      c.ride(c);
48      Tricycle d=new Tricycle();
49      d.ride(d);
50  }
51  public static void main(String ags[]){
52      new ganswer();
53  }
```

Problems Javadoc Declaration Console

```
<terminated> ganswer [Java Application] D:\Java\eclipse\java-2\bin\javaw.exe (2016年10月16日 下午11:40:44)
0
1
它不平衡
2
它不平衡
3
```

Writable Smart Insert 48 :