

00 第五次作业-2 要求

2020.4

1. 作业目标

使用蒙特卡洛仿真方法求圆周率。

2. 作业要求

2.1 业务背景

(1) 蒙特卡罗仿真

蒙特卡罗法也称统计模拟法、统计试验法。是把概率现象作为研究对象的数值模拟方法。是按抽样调查法求取统计值来推定未知特性量的计算方法。蒙特卡罗是摩纳哥的著名赌城，该法为表明其随机抽样的本质而命名。故适用于对离散系统进行计算仿真试验。在计算仿真中，通过构造一个和系统性能相近似的概率模型，并在数字计算机上进行随机试验，可以模拟系统的随机特性。

蒙特卡罗法的基本思想是：为了求解问题，首先建立一个概率模型或随机过程，使它的参数或数字特征等于问题的解；然后通过对模型或过程的观察或抽样试验来计算这些参数或数字特征，最后给出所求解的近似值。解的精确度用估计值的标准误差来表示。蒙特卡罗法的主要理论基础是概率统计理论，主要手段是随机抽样、统计试验。

(2) 圆周率的近似计算方法

使用蒙特卡洛方法估算圆周率，具体思路如下：

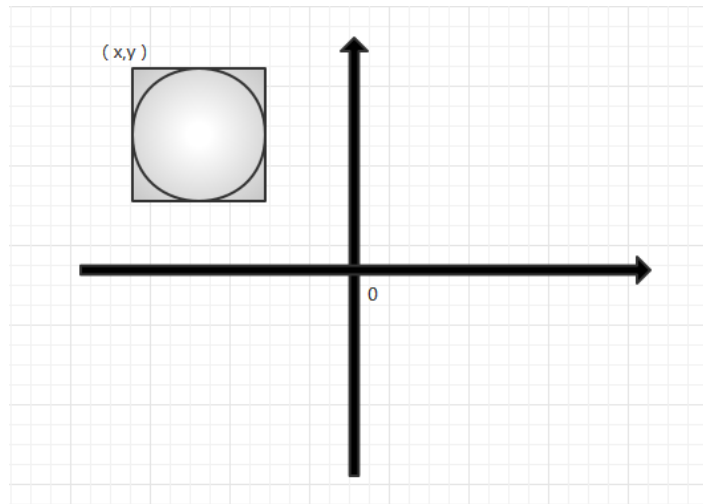
思考一个边长为 s 的正方形，它的内接圆和正方形的面积之比为 $\pi r^2 / 4 r^2 = \pi / 4$ 。由此可以在正方形的面积范围内产生大量 (n 个，需要足够大) 随机的点，再统计这些点在圆的范围内的数量 m 。则 $\pi / 4 = m / n$ ，由此可估算 π 的值。

2.2 程序功能需求

(1) 工程设计思路

在平面直角坐标系内随机选取一个坐标点，假定该坐标点为矩形左上角的位置，同时输入该矩形的长和宽，由于本例中要求矩形为正方形，所以必须要求输入的该矩形场合宽务必相等，以上均为系统的输入。

据此，可以求出该正方形内接圆的圆心坐标点及半径。然后就可以通过随机数方法获取落于正方形之内的坐标点，同时统计落在圆内的点的个数，由此就可以估算出 π 的值。如下图所示。



(2) 类设计思路

根据上述分析，可以设计如下类：坐标点、矩形、圆、仿真类。

(3) 输入输出规则

①输入规则

从键盘输入四个实型数和一个整型数，分别为矩形左上角的横坐标、纵坐标、矩形长度、矩形宽度和投点次数，数与数之间可以用一个或多个空格或回车分隔。

②输出规则

- 如果矩形长度与宽度不相等（非正方形）或长宽数据非法，则输出“Wrong Format”。
- 如果估算出的 π 与 Math.PI 差值小于 $1\text{E-}3$ ，则输出“Success”，否则输出“failed”。

3. 作业内容和成果物

3.1 作业内容

程序源码。

3.2 提交内容

在 PTA 系统中提交程序源码进行测试。

4. 作业要求和限制

4.1 输入输出示例

输入示例 1:

```
2.5 9.243 2.5 2.5 35
```

输出示例 1:

```
failed
```

输入示例 2:

```
-6.5 4.2 8 8 10000000
```

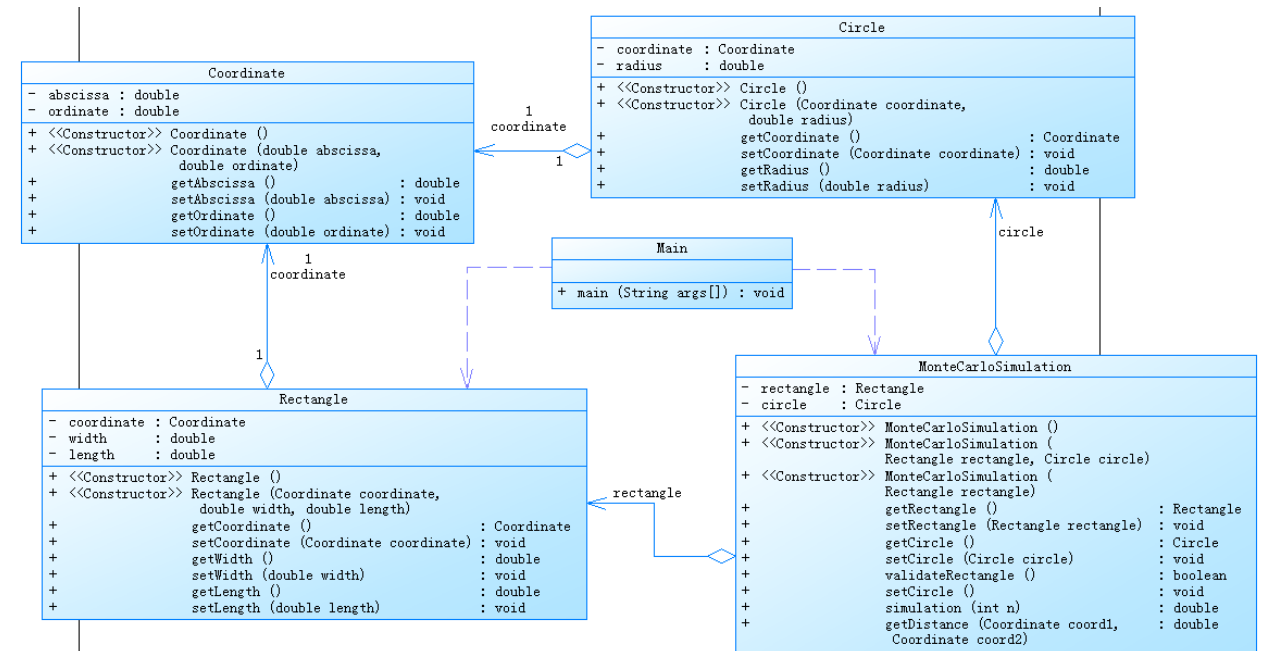
输出示例 2:

Success

输入、输出字符编码采用 utf-8 格式，例如，逗号为英文半角字符“，”，而非中文全角字符“，”。

4.2 设计要求

(1) 类结构设计如下：



(2) Main 类源码如下：

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        double abscissa,ordinate;
        double length,width;
        int count = 0;
        Scanner input = new Scanner(System.in);

        abscissa = input.nextDouble();
        ordinate = input.nextDouble();
        length = input.nextDouble();
        width = input.nextDouble();
        count = input.nextInt();

        Rectangle rectangle = new Rectangle(new Coordinate(abscissa,ordinate),length,width);
        MonteCarloSimulation monteCarlo = new MonteCarloSimulation(rectangle);
        if(monteCarlo.validateRectangle()){
            monteCarlo.setCircle();
            if((Math.abs(monteCarlo.simulation(count) - Math.PI)) <= 1e-3){
                System.out.println("Success");
            }
            else{
                System.out.println("failed");
            }
        }else{
            System.out.println("Wrong Format");
        }
    }
}
```

4.4 测试准则

PTA

5. 其它说明事项

5.1 设计建议

- (1) 坐标点类、矩形类、圆类设计尽量做到独立、可复用；
- (2) 查询蒙特卡罗仿真方法的应用；

5.2 Tips

类结构如何完善才能使得系统具有较好的可复用性。

通过此例，进一步深入理解面向对象设计原则中的“单一职责原则”。

6. 其他规定

- (1) 文档中**粗体**字体部分为强制要求。
- (2) 无效作业，以下三种情况视为无效作业。
 - 1) 程序不能编译和运行；
 - 2) 无法通过任何一个可以输出正常结果的公共测试案例；
 - 3) 测试程序无法识别。