

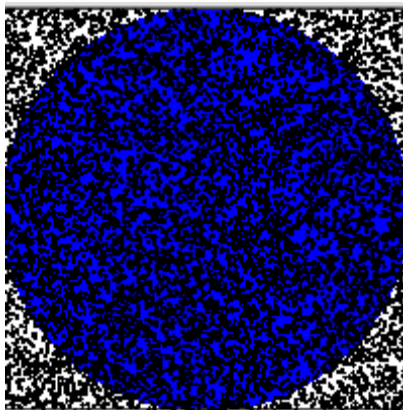


Introduction to Programming with Java

Task: GraphicalPiApproximation

Description:

Your task is to implement a program which calculates an approximation of PI visually by using **Monte Carlo Integration** algorithm. Recall our PiApproximation program. You will adapt the



same idea here but considering the difference in central points. Addition to this, you need to show this process (see the figure) by using what you learn about Graphics in class.

Information about Monte Carlo Integration algorithm:

The ratio of the number of points within the circle and total number of points is an estimate of the ratio of the two areas, blue circle and square, which is $\pi/4$. Multiply the result by 4 to estimate π .

In order to code this program, you have to write code in

- createPoint method
- run method

createPoint method takes two arguments and returns a GOval object. Its prototype is as follows:

```
private GOval createPoint(double x, double y)
```

These two arguments need to be randomly created and given to this method. It creates a point by using these two arguments and POINT_RADIUS variable which is given as a constant.

Requirements:

1. Center of a point should be at (x, y)

2. Point should be filled with black.

After you have completed the code in `createPoint` method, you need to write code inside the `run` method.

In the `run` method, you need to create random points inside the square, draw them on the screen using `createPoint` and count how many points fall in the circle. We provided you a variable above named ``inside``, for the counting process.

At the beginning, `NPOINTS` equals 1000. This does not provide you a good approximation but in the implementation phase, it gives you some intuition quickly. When you successfully implement this, please try a larger value of `NPOINTS` to see if the approximation would be better.

Requirements:

- A good approximation to Pi. Pi is approximately 3,14159265. In this example it is sufficient to achieve 3.14 ± 0.03 approximation.

Note that in an x-y Cartesian coordinate system, the circle with centre coordinates (a, b) and radius r, if the point is in the circle, following inequality should be correct:

$$(x - a)^2 + (y - b)^2 < r^2$$

Location Information:

Square: should be 200x200 and starts at (0,0)

Circle: Central point of the circle is at (100, 100) and its radius is 100.

Points: should fall within square randomly.

You might like to use private constants created at the bottom of this file.

Throughout the exercise, you will write code only between these two comment lines:

// Your code starts here

// Your code ends here

For this problem, in order to get full credit, please make sure that you present relatively a good Pi approximation (we expect to see 3.14 ± 0.03) and visualize the calculation step.

You can also find instructions and some tips for implementing this program in GraphicalPiApproximation.java.