Betsy De La Cruz

March 15, 2017

Purpose: To calculate pie using the Monte Carlo method.

Sample:

```
Number of coordinates: 10.0
Coordinates inside the circle: 8
Coordinates outside the circle: 2
Approximation to PI is 3.2
```

```
Number of coordinates: 10.0
Coordinates inside the circle: 7
Coordinates outside the circle: 3
Approximation to PI is 2.8
```

```
Number of coordinates: 100.0
Coordinates inside the circle: 79
Coordinates outside the circle:
21
```

```
Number of coordinates: 100.0
Coordinates inside the circle: 81
Coordinates outside the circle: 19
Approximation to PI is 3.24
```

```
Number of coordinates: 1000.0
Coordinates inside the circle: 782
Coordinates outside the circle: 218
Approximation to PI is 3.128
```

```
Number of coordinates: 1000.0
Coordinates inside the
circle: 804
Coordinates outside the
circle: 196
Approximation to PI is 3.216
```

```
Number of coordinates: 10000.0
Coordinates inside the circle: 7895
Coordinates outside the circle: 2105
Approximation to PI is 3.158
```

```
Number of coordinates: 10000.0
Coordinates inside the circle: 7900
Coordinates outside the circle: 2100
Approximation to PI is 3.16
```

```
Number of coordinates:
1000000.0
Coordinates inside the
circle: 785014
Coordinates outside the
circle: 214986
Approximation to PI is
3.140056
```

```
Number of coordinates: 100000.0
Coordinates inside the circle:
78424
Coordinates outside the circle:
21576
Approximation to PI is 3.13696
```

```
Number of coordinates: 100000.0
Coordinates inside the circle:
78707
Coordinates outside the circle:
21293
Approximation to PI is 3.14828
```

Analysis:

As the number of coordinates increases, the approximation approaches more to the expected value of PI as the law of large number states. This simulations shows the probability of PI as the number of times a coordinates lands inside the circle over all the number of conditions generated. This makes sense because, mentally picturing this, having small amount of coordinates will have less points inside the circle, whereas having large amount of coordinates will have a higher chance of more points inside the circle. More points inside the circle fills up most of the area so it will approach to pi, since the area inside the circle is all pi. It will only equal to pi if the whole quadrant is filled with points, but that is less likely to happen with random numbers, so it will be roughly PI with a high number of coordinates.

Before going to spring break, I was reviewing the assignment and I couldn't mentally figure out a blueprint of how to start this assignment, so I went to office hours to seek for help. Going to office hours was helpful because Dr.Joyce did go step by step of how to do this assignment. I tried thinking of another way to approach this assignment based on the notes from office hours, but I could not figure out another way. So I somewhat did this assignment from what was discussed during office hours.

This assignment took me about 2 hours and only because I could not figure out a small mistake during most of the time, and the mistake was that I did not cast a float near the end of my program.

```java
import java.util.Random;
public class PI {

	public static void main(String[] args) {
		Random GenNum = new Random();
		float x,y,PI;
		int count=0, countOut=0;
		final float coordinates=1000000;

		for(int i=1; i<=coordinates;i++)
		{
		// setting x and y to random numbers
		x=GenNum.nextFloat();
		y=GenNum.nextFloat();
		// for my purpose to hand calculate for accuracy
		//System.out.println("x is " + x);
		//System.out.println("y is " + y);
		// starting to count the coordinates inside and outside the circle
			if ( x*x+y*y<1)
				count++;
			else
				countOut++;
		}
		PI=(float)count/coordinates*4;
		System.out.println("Number of coordinates: " + coordinates);
		System.out.println("Coordinates inside the circle: " + count);
		System.out.println("Coordinates outside the circle: " + countOut);
		System.out.println("Approximation to PI is " + PI);
	}
}
```