

SEG 2105A/SEG2505A  
 Assignment 1  
 Michelle Dussault: 8174245  
 Matthew Demczyk: 8648966

E 26)

Table 2 - Hypothesized Properties of Implemented Designs

Design Type	Simplicity of code	Efficiency when creating instances	Efficiency when doing computations that require both storage systems	Amount of memory used	Ease of use for implementation
<b>Design 1</b>	Complex	Efficient	Efficient	Moderate	Easy – easily switch between 2 types
<b>Design 2</b>	Simple	Somewhat efficient	Not efficient	Moderate	Moderate – only stores as one type
<b>Design 3</b>	Simple	Somewhat efficient	Not efficient	Moderate	Moderate – only stores as one type
<b>Design 5</b>	Medium	Efficient	Not efficient	Least	Easy – can store both types in one variable of the superclass's type (because of inheritance)

E 30 )

Table 2- Average computation speeds in milliseconds

Operations	<u>Design 1</u>		<u>Design 2</u>	<u>Design 3</u>	<u>Design 5</u>	
	<i>Cartesian</i>	<i>Polar</i>	<i>Polar</i>	<i>Cartesian</i>	<i>Cartesian</i>	<i>Polar</i>
Retrieving Rho and Theta	Median: 4333  Min: 4277 Max: 4362	Median: 3390  Min: 3343 Max: 3457	Median: 4016  Min:4001 Max:4072	Median: 6577  Min:6460 Max:6770	Median: 4300  Min:4274 Max:4392	Median: 3440  Min:3324 Max:3468
Retrieving X and Y	Median: 3414  Min:3321 Max:3467	Median: 4691  Min: 4625 Max: 4756	Median: 8211  Min:8105 Max:8374	Median: 4075  Min:4035 Max:4681	Median: 3432  Min:3361 Max:3469	Median: 4690  Min:4610 Max:4772
Creating a large number of Cartesian instances	Median: 3155  Min:3104 Max:3221	N/A	Median: 6591  Min:6536 Max:6629	Median: 3934  Min:3908 Max:4641	Median: 3161  Min:3078 Max:3229	N/A
Creating a large number of polar instances	N/A	Median: 3151  Min:3124 Max:3185	Median: 3905  Min:3861 Max:3953	Median: 8194  Min:8099 Max:8276	N/A	Median: 3164  Min:3074 Max:3199
Running the method getDistance()  (finding distance between 2 polar points or 2 Cartesian points)	Median: 4902  Min:4771 Max: 5477	Median: 9488  Min:8880 Max:9918	Median: 15721  Min:15419 Max:16004	Median: 6577  Min:6460 Max:6770	Median: 4888  Min:4818 Max:5405	Median: 9435  Min:9338 Max:9550

Running the method getDistance() (finding distance between a polar and Cartesian point)	Median: 6759  Min:6619 Max: 7321		N/A	N/A	Median: 6573  Min: 6456 Max:7767	
Running the method rotatePoint()	Median: 7951  Min:7844 Max: 8865	Median: 11276  Min:9128 Max:11359	Median: 9787  Min:9675 Max:9993	Median: 7603  Min:7350 Max:8217	Median: 7964  Min:7849 Max:9046	Median: 11250  Min:11183 Max:11365

### Summary Of Results and Analysis of Hypothesis:

Our initial hypothesis was that it would be simple to create instances of design 5, while taking longer for designs 2 and 3. This was shown to be true. When creating an instance of design2 or 3 and passing it coordinates for the opposite system than it was storing in (i.e. passing Polar coordinates to design3, which only stores cartesian) it would take much longer due to converting them to the 'correct' storage system in the constructor. However when creating polar design2 instances and cartesian design3 instances the run time was very similar to design 5.

Our initial hypothesis for retrieving and doing computations for both systems was shown to be wrong by the data. We expected that it would take the same amount of time to retrieve the opposite coordinates in each design, but designs 2 and 3 took a larger amount of time to do this. However, this data is likely incorrect, as the methods are all implemented the same. A source of error may be that the tests for designs 2 and 3 were done on a different computer than design 5, and the computer used for design 2 and 3 is slower than design 5. We believe these values should be closer to the same.

Overall, we did not see a big difference between the runtimes for design 1 and design 5. This was expected, because both classes are flexible and can easily switch from polar to Cartesian, and they were tested on the same computer.

For the method rotatePoint(), it took longer to run it with polar coordinates because the method requires Cartesian values, therefore the polar coordinates must always be converted to Cartesian, and doing this an arbitrarily large amount of times will surely add to the runtime.

### How we tested our methods:

To test the efficiency of our different classes and methods, we created many test methods, but they all had the same layout, which is described below.

Each iteration of our for loop would: create an instance of a certain class, store the instance in an array, perform an operation on the instance that was just created (ex: getX()) and store the result of the operation in another array. We set our “for” loops to loop 10,000,000 times. We used System.currentTimeMillis() (end time minus start time) in order to measure the amount of time it took to for a certain class to perform a certain operation.

To obtain the computation speed of our methods, we ran each test method 11 times, and recorded the results in an excel spreadsheet. We could then easily sort the values obtained and find the median, maximum and minimum value.

#### Screenshots of output in terminal:

```
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Creating 10,000,000 instances of PointCPPolarDesign5: 3132 ms
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Creating 10,000,000 instances of PointCPPolarDesign5: 3181 ms
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Creating 10,000,000 instances of PointCPPolarDesign5: 3164 ms
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Creating 10,000,000 instances of PointCPPolarDesign5: 3138 ms
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Creating 10,000,000 instances of PointCPPolarDesign5: 3187 ms
```

```
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Test GetDistanceCartesianDesign1: 4771 ms
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Test GetDistanceCartesianDesign1: 4876 ms
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Test GetDistanceCartesianDesign1: 4919 ms
Michelles-MacBook-Pro:design5 michelledussault$ java LargeTest
Test GetDistanceCartesianDesign1: 4798 ms
```

```
Terminal File Edit View Search Terminal Help
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ javac LargeTest.java
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ echo testing retrieving rho
o/theta from design3
testing retrieving rho/theta from design3
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing retrieving Rho/Theta from 10,000,000 Design3:6770 ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing retrieving Rho/Theta from 10,000,000 Design3:6695 ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing retrieving Rho/Theta from 10,000,000 Design3:6577 ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing retrieving Rho/Theta from 10,000,000 Design3:6460 ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing retrieving Rho/Theta from 10,000,000 Design3:6636 ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing retrieving Rho/Theta from 10,000,000 Design3:6567 ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing retrieving Rho/Theta from 10,000,000 Design3:6572 ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$
```

```
Terminal File Edit View Search Terminal Help
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ javac LargeTest.java
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ echo testing creation of c
artesian design 2, the constructor will conver to polar when creating the instan
ces
testing creation of cartesian design 2, the constructor will conver to polar whe
n creating the instances
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing creating 10,000,000 Cartesian Design2:6629ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing creating 10,000,000 Cartesian Design2:6620ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing creating 10,000,000 Cartesian Design2:6545ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing creating 10,000,000 Cartesian Design2:6536ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing creating 10,000,000 Cartesian Design2:6568ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing creating 10,000,000 Cartesian Design2:6591ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$ java LargeTest
Testing creating 10,000,000 Cartesian Design2:6628ms
matthew@Matthew-Linux:~/repos/sega1-final/main/part1$
```