

SEG 2105 Assignment 1

Design Analysis

Table 1. Pros and Cons of Various Designs

Design	Pros	Cons
Design 2	<ul style="list-style-type: none">• Polar coordinates will be accessible quickly• More readable code• Less storage taken up compared to storing both forms	<ul style="list-style-type: none">• Cartesian coordinates will take longer to retrieve• Harder to work in Cartesian coordinates
Design 3	<ul style="list-style-type: none">• Cartesian coordinates will be accessible quickly• More readable code• Less storage taken up compared to storing both forms	<ul style="list-style-type: none">• Polar coordinates will take longer to retrieve• Hard to work in Polar coordinates
Design 5	<ul style="list-style-type: none">• Very readable code, allows for unique implementation of subclasses	<ul style="list-style-type: none">• Requires casting in certain instances

Testing

The tests were conducted using VS Code, wherein the programs PointCP2Test.java, PointCP3Test.java, and PointCP5Test.java ran every function of their respective classes 1 billion times, except for the toString() function. This fixed number of iterations was repeated five times, and the median times will be presented below in Table 2. For a full tabulation of results, please consult Tables 3-7 in the Appendix. Note that a capitalized bold F corresponds to the fastest execution time for a given class in each method, and a capitalized bold S corresponds to the slowest execution time for a given class in each method. Note that all times are given in seconds.

Table 2. Median Results from Testing

Designs	Median Speed Calling getRho()	Median Speed Calling getTheta()	Median Speed Calling getX()	Median Speed Calling getY()	Median Speed
---------	-------------------------------------	---------------------------------------	--------------------------------	--------------------------------	-----------------

					Calling getDistance()
PointCP2	0.007	0.011	1.111	1.091	4.191
PointCP3	0.004	63.58	0.008	0.008	0.006
PointCP5 (CP2)	0.005	0.022	1.049	1.140	4.150
PointCP5(CP3)	0.018	6.381	0.015	0.019	0.017

The results from the testing validate the initial assumptions. In CP2, where the polar coordinates were stored, getRho() and getTheta() performed faster than getX() and getY(). In CP3 the reverse was true, and getTheta() was very slow, since the calculations for the angle involved multiple Java.Math functions which would increase runtime severely. Compared to their standalone classes, CP5 had mixed results in terms of performance compared to CP2 and CP3. Some functions, such as getTheta(), performed much better as subclasses.

In summary, while the get functions that retrieved stored values performed the billion iterations relatively quickly, the get functions that did not retrieve stored values and relied on calculating the values when requested performed slower. This is inline with initial assumptions

Appendix

Please note that all times are given in seconds.

Table 3. Results from Testing Round 1

Designs	Execution Time getRho()	Execution Time getTheta()	Execution Time getX()	Execution Time getY()	Execution Time getDistance()
PointCP2	0.004 (F)	0.011	1.295 (S)	1.262 (S)	4.200
PointCP3	0.003 (F)	64.13	0.007	0.008	0.006 (S)
PointCP5 (CP2)	0.008	0.031 (S)	1.139 (S)	1.148	4.122 (F)
PointCP5(CP3)	0.020	6.381	0.015	0.019	0.039 (S)

Table 4. Results from Testing Round 2

Designs	Execution Time getRho()	Execution Time getTheta()	Execution Time getX()	Execution Time getY()	Execution Time getDistance()
PointCP2	0.007	0.007	1.094	1.095	4.191
PointCP3	0.004	63.53 (F)	0.006 (F)	0.008	0.006 (S)
PointCP5 (CP2)	0.010 (S)	0.031 (S)	1.025 (F)	1.140	4.210 (S)
PointCP5(CP3)	0.011 (F)	6.383	0.012 (F)	0.014	0.014

Table 5. Results from Testing Round 3

Designs	Execution Time getRho()	Execution Time getTheta()	Execution Time getX()	Execution Time getY()	Execution Time getDistance()
PointCP2	0.008 (S)	0.006 (F)	1.084 (F)	1.063	4.222 (S)
PointCP3	0.004	63.68	0.011	0.009 (S)	0.005 (F)
PointCP5 (CP2)	0.005 (F)	0.016	1.049	1.100 (F)	4.150
PointCP5(CP3)	0.012	6.305 (F)	0.026	0.010 (F)	0.026

Table 6. Results from Testing Round 4

Designs	Execution Time getRho()	Execution Time getTheta()	Execution Time getX()	Execution Time getY()	Execution Time getDistance()
PointCP2	0.006	0.011	1.126	1.047 (F)	4.119 (F)
PointCP3	0.009 (S)	64.76 (S)	0.008	0.005 (F)	0.006 (S)
PointCP5 (CP2)	0.005 (F)	0.014 (F)	1.036	1.200 (S)	4.190
PointCP5(CP3)	0.018	6.350	0.028 (S)	0.027	0.017

Table 7. Results from Testing Round 5

Designs	Execution Time getRho()	Execution Time getTheta()	Execution Time getX()	Execution Time getY()	Execution Time getDistance()
PointCP2	0.008(S)	0.015 (S)	1.111	1.091	4.147
PointCP3	0.003 (F)	63.58	0.012 (S)	0.008	0.006 (S)
PointCP5 (CP2)	0.005 (F)	0.022	1.121	1.123	4.149
PointCP5(CP3)	0.028 (S)	6.628 (S)	0.015	0.033 (S)	0.011 (F)