Performance Analysis:

Median run time of the test(in ms):

Function	Design 1 (Initial with Polar)	Design 1 (Initial with Cartesian)	Design 2	Design 3	Design 5 (Initial with Polar)	Design 5 (Initial with Cartesian)
getX()	273 Min: 260 Max: 295	12 Min: 12 Max: 13	258 Min: 239 Max: 283	3 Min: 2 Max: 3	238 Min: 235 Max: 239	3 Min: 2 Max: 3
getY()	233 Min: 225 Max: 243	33 Min: 32 Max: 37	230 Min: 209 Max: 244	2 Min: 1 Max: 2	210 Min: 208 Max: 211	2 Min: 2 Max: 3
getRho()	3 Min: 3 Max: 3	16 Min: 14 Max: 18	3 Min: 2 Max: 3	2 Min: 2 Max: 3	2 Min: 2 Max :3	2 Min: 2 Max: 3
getTheta()	3 Min: 3 Max: 3	1227 Min: 1213 Max: 1241	3 Min: 2 Max: 3	1138 Min: 1127 Max: 1157	2 Min: 2 Max: 3	1253 Min: 1230 Max: 1292
getDistance()	890 Min: 881 Max: 904	8 Min: 7 Max: 8	843 Min: 801 Max: 869	3 Min: 3 Max: 4	825 Min: 802 Max: 852	4 Min: 3 Max: 4
rotatePoint()	1318 Min: 1277 Max: 1398	802 Min: 771 Max: 831	1208 Min: 1189 Max: 1233	801 Min: 800 Max: 802	1253 Min: 1207 Max: 1291	846 Min: 808 Max: 874
Total Run Time	2721 Min: 2684 Max: 2770	2098 Min: 2099 Max: 2120	2545 Min: 2387 Max: 2593	1949 Min: 1940 Max: 1966	2531 Min: 2462 Max: 2567	2110 Min: 2056 Max: 2177

Description of test:

The above result was tested using the program PerformanceTest, which is also available on GitHub. To obtain accurate data, five tests were performed, and the median, minimum and maximum were recorded.

For the tests, a fixed number of iterations (100,000,000) were used to calculate the elapsed time for each method, namely getX, getY, getRho, getTheta, getDistance, and rotatePoint.

The program is designed to test the elapsed time by creating different instances of point classes with random variables and then invoking the respective methods (testPointCP, testPointCP2, testPointCP3, testPointCP5Polar, testPointCP5Cartesian). It will then print out the elapsed time for each method, including the total time of the test.

Discussion:

In the PointCP class, the "get" method for each coordinate will only perform quickly when the point is created with that type of coordinate. For example, when the point is initialized with polar coordinates, the average time for retrieving rho and theta is 3ms, while retrieving the x and y coordinates takes around 250ms.

In design 2 and design 3, there is the same issue as in PointCP; however, on average, the time taken to retrieve each coordinate has decreased by around 20ms when computing different types of coordinates.

In design 5, since it is an abstract class with subclasses of design 2 and 3, it has similar results to design 2 and 3, but it has a clearer and simpler method calling.

In conclusion, the efficiency of computation will be slower when calculating different types of coordinates compared to their initial type. However, PointCP is relatively slower than other designs.