Assignment 1

Riley de Gans 300170104

Nidhi Pareshkumar Thakkar 300202450

E26

Variables considered

1. Simplicity of Code
2. Efficiency when creating instances
3. Efficiency when doing polar computations
4. Efficiency when doing cartesian computations
5. Amount of memory used

Design #	Advantages	Disadvantages
Design 1	- Fast computations if storing the desired coordinate type, checks coordinate type, but no computation - High efficiency when creating instances, only 2 arguments	- Slow computations if not storing the desired coordinate type, checks coordinate type and computes - Higher memory usage, stores value and flag - Low ease of coding, extra methods for conversion, checking for flag etc
Design 2	- Fast polar computations, no checking coordinate type and no computation - High efficiency when creating instances, only 2 arguments - RotatePoint method is more efficient than design 3 - Lowest memory usage, only stores value - High ease of coding, simplest implementation and readability	- Slow cartesian computations, no checking coordinates, but computations - getDistance method requires more calculations and more time
Design 3	- Fast cartesian computations, no checking coordinate type and no	- Slow polar computations, no checking coordinates, but computations

	computation - High efficiency when creating instances, only 2 arguments - getDistance method is more efficient than Design 2. - Lowest memory usage, only stores value - High ease of coding, simplest implementation and readability	- RotatePoint requires more time as there are more calculations than design 2.
Design 5	- Fast computations if storing the desired coordinate type, no checking coordinate types and no computation, but dynamic binding	- Slow computations if not storing the desired coordinate type, no checking coordinate types, but computations and dynamic binding - Higher memory usage, stores superclass and subclass - Low efficiency when creating instances and low simplicity of coding, confusion caused by subclass superclass relationship

E30

Comparing Average Computation Speed of Different Designs for Different Operations
Time(ms)

Operations	Design 1	Design 2	Design 3	Design 5
getX()	348	378	7	360
getY()	347	376	8	359
getRho()	669	8	897	683
getTheta()	456	8	754	460

getDistance(PointCP other	1324	1657	832	1329
rotatePoint(double rotation)	1988	22	987	654

By measuring the computation speed we can say that getX() method is most efficient in design 3 while it is least efficient in design 2. Similarly, the getY() method is most efficient in design 3 while it is least efficient in design 2. However, the getRho() method and getTheta() method is most efficient in design 2 and least efficient in design 3. The getDistance(PointCP other) method works well in design 3. And the rotatePoint(double rotation) method works well in design 2.