Group member 1: Jiachen Yu 300302386

Group member 2: Hengjing zhang 300288003

E26, E28-30

- E26

Design	Pros	Cons
Design1	• Able to use either polar or cartesian coordinates. • getX(), getY() methods run fast for Cartesian coordinate since they are simple return • getTheta(), getRho methods run fast for Polar coordinate since they are simple return	• Almost every methods need to consider the types of the point • getX(), getY() methods run slow for polar coordinate since they need to calculate first • getTheta(), getRho methods run slow for cartesian coordinate since they need to calculate first
Design2	• getRho() method runs fast • getTheta() method runs fast	• Mathematical calculations are performed every time the getX() and getY() methods are called, which may impact performance, especially when a large number of coordinate calculations are needed.
Design3	• getX() method runs fast • getY() method runs fast	• Mathematical calculations are performed when getTheta() called, making it become slow • Will create a new instance when rotating point. More memory needed.
Design5 (abstract)	• Can be implemented with specific behaviors in different subclasses. • Allows different subclasses to inherit. • The abstract class increases the code's flexibility and extensibility.	• Must be created to implement these methods, which could add complexity to the code.

- E28-E30

With Random Coordinates	Design1			Design5		
	Max	Median	Min	Max	Median	Min
getX()	70.65738	20.35642	0.00000	47.93554	20.35817	0.00000
getY()	124.20313	40.88833	0.00000	48.02971	20.37038	0.00000
getRho()	3.16721	3.06250	0.00000	3.08938	0.00000	0.00000
getTheta()	2.83308	0.00000	0.00000	2.67846	0.00000	0.00000

With Cartesian Coordinates only	Design1			Design5		
	Max	Median	Min	Max	Median	Min
getX()	11.67413	0.00000	0.00000	3.12004	0.00000	0.00000
getY()	17.42413	0.00029	0.00000	2.64867	0.00000	0.00000
getRho()	4.94713	0.00000	0.00000	2.56967	0.00000	0.00000
getTheta()	122.74471	51.85331	0.00000	111.76775	53.23719	0.00000

With Polar Coordinates only	Design1			Design5		
	Max	Median	Min	Max	Median	Min
getX()	74.48804	20.26273	0.00000	51.39425	20.26519	0.00000
getY()	124.57025	40.58204	0.00000	48.34646	20.33171	0.00000
getRho()	3.05438	0.00000	0.00000	2.87933	0.00000	0.00000
getTheta()	2.63721	0.00000	0.00000	2.45996	0.00000	0.00000

For the performance test, it is necessary to run each Function (e.g. getX()) a very large number of times to ensure the runtime for each function can be measure in normal scale. By this reason, the test is run under 1000000 times repeating of each function. For performance testing accuracy the of Design 1 and Design 5, we have collected the 100 runs of the procedure above to get median and also different coordinates measurement applied during the test. (Cartesian, Polar and Random)

Base on the performance test result we can conclude:

1. The efficiency of Design 1 and 5 getX() and getY() under Cartesian only mostly the same and running time are also very low, the reason is these function is basically simple return the parameters so function is O(1). Relatively, the getRho() and getTheta() take more time because the math calculation involved to return the value.

2. Oppositely, in Polar only test, the situation became reversed, the reason is that getX() and getY() have the math function involved and getRho() and getTheta() were simple return.
   The testing result is basically match our hypotheses.

3. With the random coordinates simple test, the functions running time between Design 1 and Design 5 is relatively close. The main reason is that the active function in both Designs are

the same, the only difference between Design 1 and 5 is design 1 contain all function in one class and Design 5 is an abstract class which need to trigger design5.PointCP3 and design5.PointCP2 for active function. But basically they will hold same efficiency during this test.