|  |  |  |
| --- | --- | --- |
|  | Pros | Cons |
| Design 1 | * Simple code | * Least efficient instance creation * Uses more memory * Slowest |
| Design 2 | * Simple code * Efficient instance creation * Doesn’t use much memory |  |
| Design 3 | * Simple code * Efficient instance creation * Doesn’t use much memory |  |
| Design 5 | * Efficient instance creation * Doesn’t use much memory * Fastest | * Most complex code |

Test outputs:

**Design1**

**Type C Test output**

Rotate test

Max time = 4

Min time = 0

Total time = 10

Distance test

Max time = 1

Min time = 0

Total time = 2

**Type P Test output**

Rotate test

Max time = 3

Min time = 0

Total time = 5

Distance test

Max time = 2

Min time = 0

Total time = 2

**Design2 Test**

Rotate test

Max time = 5

Min time = 0

Total time = 8

Distance test

Max time = 0

Min time = 0

Total time = 0

**Design3 Test**

Rotate test

Max time = 2

Min time = 0

Total time = 6

Distance test

Max time = 0

Min time = 0

Total time = 0

**Design5 Test**

**design3**

Rotate test

Max time = 5

Min time = 0

Total time = 9

Distance test

Max time = 1

Min time = 0

Total time = 1

**design2**

Rotate test

Max time = 3

Min time = 0

Total time = 6

Distance test

Max time = 1

Min time = 0

Total time = 1

I tested by creating instances of PointCP with random numbers running either rotatePoint() or getDistance() 1000 times. I store the max time it took, the minimum time and the total time.

|  |  |  |  |
| --- | --- | --- | --- |
| Design | maxTime (ms) | minTime (ms) | totalTime (ms) |
| Design 1 Type C | Rotate (4)  Distance (1) | Rotate (0)  Distance (0) | Rotate (10)  Distance (3) |
| Design 1 Type P | Rotate (3)  Distance (2) | Rotate (0)  Distance (0) | Rotate (5)  Distance (2) |
| Design 2 | Rotate (5)  Distance (0) | Rotate (0)  Distance (0) | Rotate (8)  Distance (0) |
| Design 3 | Rotate (2)  Distance (0) | Rotate (0)  Distance (0) | Rotate (6)  Distance (0) |
| Design5 (design3) | Rotate (5)  Distance (1) | Rotate (0)  Distance (0) | Rotate (9)  Distance (1) |
| Design5 (design2) | Rotate (3)  Distance (1) | Rotate (0)  Distance (0) | Rotate (6)  Distance (1) |

Rotation time is faster when storing polar coordinates, distance time is faster when storing cartesian coordinates. Rotation time is a lot slower than distance time especially when storing cartesian coordinates. The fastest design is design5. The slowest design is design1.