Iona Buchanan 6671041

Christopher Wong

Assignment 1

# Polar/Cartesian Points

## Advantages and Disadvantages of each design

|  |  |  |
| --- | --- | --- |
| Design | Pros | Cons |
| Design 1: Store one type of coordinates using a single pair of instance variables, with a flag indicating which type is stored | * Versatile * Everything can be done with this single class | * Extra memory to store flag * Most operations require checking the value of the flag |
| Design 2: Store polar coordinates only | * Returning polar coordinates is simply returning instance vars | * Retrieving cartesian coordinates requires calculation/conversion each time |
| Design 3: Store cartesian coordinates only | * Returning Cartesian coordinates is simply returning instance vars | * Retrieving polar coordinates requires calculation/conversion each time |
| Design 6: Interface with designs 2 and 3 as classes implementing it. | * Ties together designs 2 and 3, making the full design versatile like Design 1 | * Requires interface class file |

## Hypothesis of Running Times

The greatest factor in running time is whether the method requires calculating the conversion between and . Running time is also affected by the checking of coordinate type in the case of Design 1.

Estimated relative running times for each method:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | PointCP (P) | PointCP (C) | PointCP2 | PointCP3 | Point6 (PointP) | Point6 (PointC) |
| getRho();getTheta();: | fast | slow | fastest | slow | fastest | slow |
| getX();getY();: | slow | fast | slow | fastest | slow | fastest |
| convertStorageToPolar(): | fast | slower | fast | slow | fast | slow |
| convertStorageToCartesian(): | slower | fast | slow | fast | slow | fast |
| rotatePoint(rotation): | slow | faster | slow | faster | slow | faster |
| getDistance(pointB): | slow | fast | slow | fast | slow | fast |
| toString(): | slow | slow | slow | slow | slow | slow |

## Test Procedure

1. Create random instances of each design class (polar and cartesian).
2. For each instance, call each method (getRho(), getTheta(), convertStorageToPolar() etc), times and record time in milliseconds. Choose sufficiently large to get accurate timings.

## Results

## Discussion

* getDistance() is slow for polar coordinates because it requires converting both sets of coordinates to Cartesian.
* getString() is slow due to the string concatenation/formatting operation.
* Designs 2, 3 and 6 are more specialized (they are fast depending on the type of coordinates being used. However, Design 1 is more versatile and better for dealing with both types of coordinates together.

# Arrays