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 |  | * 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 |  |
| Design 3: Store cartesian coordinates only | * Returning Cartesian coordinates is simply returning instance vars |  |
| Design 6: Interface with designs 2 and 3 as classes implementing it. |  |  |

## Hypothesis of Running Times

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | PointCP (P) | PointCP (C) | PointCP2 | PointCP3 | Point6 (PointP) | Point6 (PointC) |
| getRho();getTheta();: | fast | slow | fast | 68955 | fast | 78623 |
| getX();getY();: | slow | fast | 99128 | fast | 97821 | fast |
| convertStorageToPolar(): | fast | 2036 | fast | 72704 | fast | 75602 |
| convertStorageToCartesian(): | 779 | 726 | 519 | 472 | 458 | 484 |
| rotatePoint(rotation): | 218683 | 220253 | 408202 | 249449 | 432449 | 228028 |
| getDistance(rotation): | 725 | 523 | 219361 | 512 | 180132 | 464 |
| toString(): | slow | slow | fast | fast | fast | fast |

## Test Procedure

## Results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Timings for 10 000 000 runs (milliseconds) |  |  |  |  |  |  |
|  | PointCP (P) | PointCP (C) | PointCP2 | PointCP3 | Point6 (PointP) | Point6 (PointC) |
| getRho();getTheta();: | 89 | 1039 | 39 | 712 | 34 | 703 |
| getX();getY();: | 2041 | 42 | 880 | 38 | 930 | 34 |
| convertStorageToPolar(): | 38 | 48 | 44 | 704 | 38 | 708 |
| convertStorageToCartesian(): | 36 | 42 | 47 | 50 | 48 | 47 |
| rotatePoint(rotation): | 3729 | 2111 | 4363 | 2295 | 5150 | 4207 |
| getDistance(rotation): | 76 | 66 | 3545 | 47 | 2095 | 35 |
| toString(): | 12186 | 7253 | 6816 | 6432 | 7272 | 6364 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Timings for 100 000 000 runs (milliseconds) |  |  |  |  |  |  |
|  | PointCP (P) | PointCP (C) | PointCP2 | PointCP3 | Point6 (PointP) | Point6 (PointC) |
| getRho();getTheta();: | 128 | 10866 | 82 | 7366 | 93 | 8532 |
| getX();getY();: | 11613 | 89 | 9796 | 78 | 13590 | 95 |
| convertStorageToPolar(): | 90 | 306 | 148 | 8660 | 205 | 9999 |
| convertStorageToCartesian(): | 91 | 118 | 130 | 112 | 139 | 110 |
| rotatePoint(rotation): | 19057 | 19216 | 38483 | 20017 | 44351 | 19761 |
| getDistance(rotation): | 9132 | 79 | 18971 | 77 | 20611 | 92 |
| toString(): | 81215 | 79047 | 75418 | 71251 | 67975 | 11119 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Timings for 1 000 000 000 runs (milliseconds) |  |  |  |  |  |  |
|  | PointCP (P) | PointCP (C) | PointCP2 | PointCP3 | Point6 (PointP) | Point6 (PointC) |
| getRho();getTheta();: | 512 | 73383 | 484 | 68955 | 570 | 78623 |
| getX();getY();: | 121269 | 567 | 99128 | 464 | 97821 | 514 |
| convertStorageToPolar(): | 481 | 2036 | 495 | 72704 | 563 | 75602 |
| convertStorageToCartesian(): | 779 | 726 | 519 | 472 | 458 | 484 |
| rotatePoint(rotation): | 218683 | 220253 | 408202 | 249449 | 432449 | 228028 |
| getDistance(rotation): | 725 | 523 | 219361 | 512 | 180132 | 464 |
| toString(): | 742463 | 760226 | 655574 | 1105293 | 705080 | 1078330 |

## Discussion

# Arrays