|  |
| --- |
| SEG2105 |
| Object Oriented Concepts |
| Assignment 1 |

|  |
| --- |
| Natalia Maximo (8756866), Liam Wilford (8807534)  9-30-2018 |

Point CP Class Designs

Three separate designs of the PointCP class were implemented based on the specifications articulated in the document “Object Oriented Concepts”. The first, Design 2, stored coordinates provided only as their polar value, whereas the second – Design 3 – stored coordinates provided only in their Cartesian form. The last item, Design 6, was an interface that specified the criteria that must be met by any class that should implement it. Two further classes (clones of the PointCP classes of Design 2 and Design 3) were placed in the same package, named appropriately and made to implement the Design 6 class ‘PointCP’.

Testing was done on all three designs by measuring the time, in nanosecond, that each method took to produce its results. For the data collected in Tables 1-4, the data was measured over tests of 10,000 randomized samples for a total of 100 tests. For the testing of Design 6, half of the randomized samples were initialized as PointCartesian objects, and half of the samples were initialized as PointPolar objects.

It is evident based on the experimental data that the far superior implementation for most of the methods utilized by the PointCP class is Design 3. When considering average sample time[[1]](#footnote-1), best overall test time[[2]](#footnote-2), worst overall test time[[3]](#footnote-3), and median test time[[4]](#footnote-4), the Design 3 implementation far outpaced its peers in all tests other than the Generate Polar Coordinate test. This result is largely expected, as the calculations used in all calculating methods utilized Cartesian values in their equations.

It should be noted, as well, that the values of Design 6 tests were found to be roughly an average of the Design 2 and Design 3 tests, given that the implemented classes themselves were an amalgamation of the two other designs. Given this, it would be the most advantageous if an implementation were made with Design 6 that primarily favoured a PointCartesian object, except for when quick access to the polar equivalents were primarily and routinely needed.

Array, ArrayList, Vector Testing

The Array in this use case is superior in terms of its time to construct, and its time to sum. Obviously, the advantages that Vectors and Array Lists have over arrays is their capacity to grow, as the array must be created with a set size.

When comparing the ArrayList and Vector classes’ performance[[5]](#footnote-5), it is interesting to note that the vector was slightly faster when growing, compared to the growing Arraylist, but was slower when both were created at a preset length. However, the Vector class was more than 10x slower than the ArrayList when summing the elements, which is a substantial difference.

In conclusion, if the number of elements is known, an Array will be most efficient. If the number of elements is not known, either Vector or ArrayList should be used, but if one is going to iterate over the list in any capacity, ArrayList is far superior.

Tables

Table 1. Average Sample time in ns for each test

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design 2 | Design 3 | Design 6 |
| Calculate Distance | 44.53 | 21.00 | 32.52 |
| Rotate Point | 135.64 | 55.69 | 98.33 |
| Generate Cartesian | 38.60 | 20.18 | 31.54 |
| Generate Polar | 19.83 | 73.38 | 47.92 |

Table 2. Best test time in ns

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design 2 | Design 3 | Design 6 |
| Calculate Distance | 400500 | 174646 | 269369 |
| Rotate Point | 1297934 | 508591 | 905388 |
| Generate Cartesian | 359245 | 173506 | 277898 |
| Generate Polar | 174357 | 674966 | 406190 |

Table 3. Worst test time in ns

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design 2 | Design 3 | Design 6 |
| Calculate Distance | 957146 | 622377 | 1093691 |
| Rotate Point | 3170112 | 1340615 | 3163586 |
| Generate Cartesian | 714844 | 517339 | 761465 |
| Generate Polar | 559506 | 2652159 | 959157 |

Table 4. Median test time in ns

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design 2 | Design 3 | Design 6 |
| Calculate Distance | 416426.0 | 182323.5 | 279193.0 |
| Rotate Point | 1310011.0 | 521666.5 | 915050.0 |
| Generate Cartesian | 372044.0 | 180486.0 | 289427.5 |
| Generate Polar | 185023.5 | 706559.5 | 451980.5 |

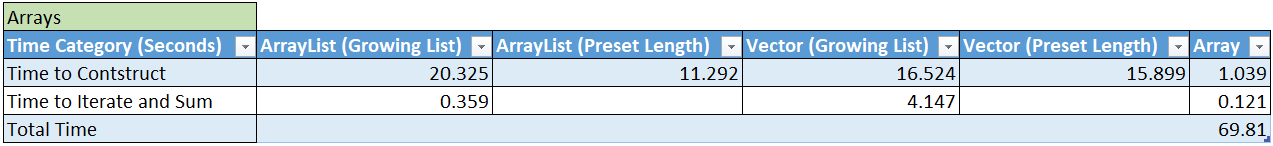
Figures

Figure 1. Testing of Array, Vector, and ArrayList datatypes

1. See Table 1 [↑](#footnote-ref-1)
2. See Table 2 [↑](#footnote-ref-2)
3. See Table 3 [↑](#footnote-ref-3)
4. See Table 4 [↑](#footnote-ref-4)
5. See Figure 1 [↑](#footnote-ref-5)