E26:

|  | Advantages | Disadvantages |
| --- | --- | --- |
| Design1 | 1.It’s efficient for users to choose which coordinate they want and get results. 2.Flexibility when choosing the coordinate systems. | 1.Complex code because it needs more methods and variables.2.Slow running speed. |
| Design2 | 1.Get the polar coordinates more efficiently.2.Simple storage for polar coordinates. | Only suit for polar coordinate case, or it needs conversion. |
| Design3 | 1.Get the cartesian coordinates more efficiently.2.Simple storage for cartesian coordinates. | Only suit for cartesian coordinate case, or it needs conversion. |
| Design4 | 1.No need to compute and convert data between two systems. | More memory is used because it needs to save more data. |
| Design5 | 1.Flexibility when choosing the coordinate system depending on which concrete class is used. 2.Polymorphism is used in the design which fits the rules of engineers.. | 1.Complex code which creates three classes to achieve the goal. 2. Slow running speed compare with others. |

**Time magnitude Test**

Design2:

1. Elapsed time for 10000 iterations: 9 milliseconds
2. Elapsed time for 10000 iterations: 10 milliseconds
3. Elapsed time for 10000 iterations: 10 milliseconds
4. Elapsed time for 10000 iterations: 11 milliseconds
5. Elapsed time for 10000 iterations: 9 milliseconds
6. Elapsed time for 10000 iterations: 10 milliseconds
7. Elapsed time for 10000 iterations: 9 milliseconds
8. Elapsed time for 10000 iterations: 9 milliseconds
9. Elapsed time for 10000 iterations: 9 milliseconds
10. Elapsed time for 10000 iterations: 9 milliseconds

Design3:

1. Elapsed time for 10000 iterations: 7 milliseconds
2. Elapsed time for 10000 iterations: 8 milliseconds
3. Elapsed time for 10000 iterations: 7 milliseconds
4. Elapsed time for 10000 iterations: 8 milliseconds
5. Elapsed time for 10000 iterations: 8 milliseconds
6. Elapsed time for 10000 iterations: 8 milliseconds
7. Elapsed time for 10000 iterations: 8 milliseconds
8. Elapsed time for 10000 iterations: 8 milliseconds
9. Elapsed time for 10000 iterations: 8 milliseconds
10. Elapsed time for 10000 iterations: 7 milliseconds

Design5:

1. Elapsed time for 10000 iterations: 24 milliseconds
2. Elapsed time for 10000 iterations: 27 milliseconds
3. Elapsed time for 10000 iterations: 26 milliseconds
4. Elapsed time for 10000 iterations: 24 milliseconds
5. Elapsed time for 10000 iterations: 25 milliseconds
6. Elapsed time for 10000 iterations: 28 milliseconds
7. Elapsed time for 10000 iterations: 26 milliseconds
8. Elapsed time for 10000 iterations: 28 milliseconds
9. Elapsed time for 10000 iterations: 26 milliseconds
10. Elapsed time for 10000 iterations: 29 milliseconds

|  | Design2 | Design3 | Design5 |
| --- | --- | --- | --- |
| Average Time Consuming | 9.5ms | 7.7ms | 26.3ms |
| Maxium Time | 11ms | 8ms | 29ms |
| Minium Time | 9ms | 7ms | 24ms |

From the table, it it easy to tell that compared with design 2 and design 3, design 5 needs more time to create variables and call the method as for design 2 and 3, they just operate simple case with only one data sort, but for design 5, there are parent and child classes need to be called each time, which consumes a lot of times. The result verifies the hypothesis.

For my test, I set an iteration number which is close to 10 seconds, and set a time spot at the beginning of the program, and another at the end of the program. Making the program run for about 10 seconds, I call every method in the class every time, and finally get the answer.

int numIterations = 10000; // Number of iterations

PointCP[] points = new PointCP5[numIterations];

Random random = new Random();

long startTime = System.currentTimeMillis();

for(int i=0; i < numIterations;i++){  
 create instances

call every method in the class

}

long endTime = System.currentTimeMillis();

long elapsedTime = endTime - startTime;

Test for design 5 is a little bit different

It creates random instances first and then go through every instance by calling methods.