**Q.E26**

|  | Pros | Cons |
| --- | --- | --- |
| D1 | * Simple * Easier to code * Less memory Intake | * Less efficient * Slow |
| D2 | * Efficient instance creation * Less time to create | * Takes more memory |
| D3 | * Simple code * Less time to create an instance | * Takes more time to convert from one system to another * More memory usage |
| D5 | * Less time to create instances since there are subclasses * Fatest | * Complex code * Uses the most memory since it has subclasses |

**Q. E27, E28, E29**

This test assesses the efficiency of the system in converting between Design1 and Design5 while taking into account constant values and system considerations. It involves generating random integers for both Design1 and Design5 to iterate through, performing coordinate retrieval and system conversion methods in each iteration. The timer for both designs measures time in milliseconds.

|  |  | Min Time (ms) | Max Time (ms) | Median Time (ms) |
| --- | --- | --- | --- | --- |
| Design 1 | Distance  Rotation | 0.000  1.661 | 0.403  1.736 | 0.0000001  1.701 |
| Design 2 | Distance  Rotation | 0.387  1.656 | 0.427  1.853 | 0.401  1.716 |
| Design 3 | Distance  Rotation | 0.0  0.908 | 0.005  1.023 | 0.0000001  0.945 |
| Design 5 (D2) | Distance  Rotation | 0.387  0.882 | 0.523  1.143 | 0.402  0.908 |
| Design 5 (D3) | Distance  Rotation | 0.000  0.902 | 0.005  0.940 | 0.0000001  0.927 |

As shown above in the table, we can clearly see that the design 5 with both using design 2 and 3 are much faster than design 1, 2, and 3. We can conclude that although design 5 is more complex and requires more memory to compile and run, it is indeed faster at calculating the distance and rotation of vectors.