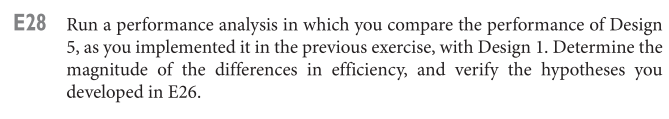


|  |  |  |
| --- | --- | --- |
| Design | Advantages (Pros) | Disadvantages (Cons) |
| Design 1 | Flexibility to store either coordinate type. | Increased memory usage due to an extra flag, complexity in code |
| Design 2 | Simple and memory-efficient for polar coordinates. | Inefficient for computations requiring Cartesian coordinates. |
| Design 3 | Simple and memory-efficient for Cartesian coordinates. | Inefficient for computations requiring polar coordinates. |
| Design 4 | Provides direct access to both coordinate types. | Uses more memory, slightly more complex code. |
| Design 5 | Common methods shared among subclasses, code reusability. | Requires subclass-specific implementations, slight overhead. |



By editing the PointCPTest.java file, we ran the program testing, taking the time to create points (same value) by using two designs (1 and 5).

Turns out Design 1 takes 6293800 nanoseconds, in comparison to design 5, which only costs 1266700. It indeed fits our hypotheses.

*Run Results Shown*

