**Java Code Challenge**

In the 17th century, the discipline of probability theory got its start when a gambler asked a mathematician friend to explain some observations about dice games.

Why did he, on average, lose a bet that at least one six would appear when rolling a die four times? And why did he seem to win a similar bet, getting at least one double-six when rolling a pair of dice 24 times?

Nowadays, it seems astounding that any person would roll a pair of dice 24 times in a row, and then repeat that many times over.

Let’s do that experiment programmatically instead.

Simulate each game a million times and print out the wins and losses, assuming each bet was for $1.

- Simulate 1000000 plays of the first game: You win if you get one six in four rolls of one dice.

- Simulate 1000000 plays of the second game: You win if you get double sixes in twenty four rolls of two dice.

Given a payout of $1 when a win is hit calculate the:

* Mean
* Variance
* Standard deviation

For each game's payouts.

Be prepared to discuss the design and implementation of your solution if you move into the next round of interviews. Points will be awarded to your solution based on:

* Whether the solution compiles and works as expected
* How clean and refactored the solution code is
* How would you scale your solution so that it can handle even larger simulations?
* Whether the statistics have been extracted correctly

Please do use any information you find online to help you with the solution.