Monte Carlo Door problem

```java

public class DoorGame {
    private int trials;
    private Random random;

    public DoorGame(int trials) {
        this.trials = trials;
        this.random = new Random();
    }

    // Monte Carlo simulation for not switching doors when given the chance
    public double monteCarloStay() {
        int correctPicks = 0;

        for (int i = 0; i < trials; i++) {
            int winningDoor = random.nextInt(3) + 1; //randomly chooses the winning door
            int chosenDoor = random.nextInt(3) + 1;//randomly chooses the door the contestants picks
            //if the winning door is logicly equivelent to the door the random generator picks
            if (winningDoor == chosenDoor) {
                correctPicks++;
            }
        }
        //creates the percentage
        return (double) correctPicks / trials * 100;
    }

    // Monte Carlo for switching the door when given the chance
    public double monteCarloSwitch() {
        int correctPicks = 0;

        for (int i = 0; i < trials; i++) {
            int winningDoor = random.nextInt(3) + 1; //randomly chooses the winning door
            int chosenDoor = random.nextInt(3) + 1;//randomly chooses the door the contestants picks

            //a non winning door is revealed
            int revealedDoor;
            do {
                revealedDoor = random.nextInt(3) + 1;
            } while (revealedDoor == winningDoor || revealedDoor == chosenDoor);

            // the contestant switches the door when given the chance
            int switchedDoor = (1 + 2 + 3) - chosenDoor - revealedDoor;

            if (switchedDoor == winningDoor) {
                correctPicks++;
            }
        }
        //creates the percentage
        return (double) correctPicks / trials * 100;
    }
}

```

```
Console
<terminated> Tester (8) [Java Application] C:\Users\Jaiden Nunez\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (Oct 23, 202
Correct Picking Percentage without Switching: 33.3193%
Correct Picking Percentage with Switching: 66.59570000000001%
```

```java

public class Tester {

    public static void main(String[] args) {
        int trials = 1000000;
        DoorGame game = new DoorGame(trials);

        double stayPercentage = game.monteCarloStay();
        double switchPercentage = game.monteCarloSwitch();

        System.out.println("Correct Picking Percentage without Switching: " + stayPercentage + "%");
        System.out.println("Correct Picking Percentage with Switching: " + switchPercentage + "%");
    }
}

```

```
Console
<terminated> Tester (8) [Java Application] C:\Users\Jaiden Nunez\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (Oct 23, 2024, 9:00:24 AM
Correct Picking Percentage without Switching: 33.3193%
Correct Picking Percentage with Switching: 66.59570000000001%
```