Ta’Quawn Watts

Project 1 Java Program Written Documentation

Project 1’s Java program has four classes and four tester classes. The four classes are: StatsLibrary, Person, BirthdayProgram, and DoorGame. The four tester classes are: TestStatsLibrary, TestPerson, TestBirthdayProgram, and TestDoorGame.

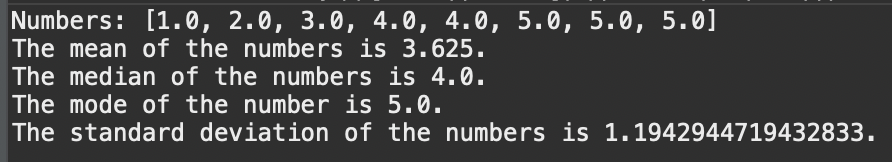
StatsLibrary is a class hosting a library of various functions used primarily in statistics. The methods included are findMean(), findMedian(), findMode(), findStdDev(), findUnion(), findIntersection(), findComplement(), factorial(), solveCombo(), and solveBinoDistrib(), solveGeoDistrib().

findMean() returns the mean of a list. findMedian returns the median of a list. findMode() returns the mode of a list and returns null if there are no mode or multiple. findStdDev() returns the standard deviation using the formula . TestStatsLibrary uses the mentioned methods generate the following output for an ArrayList named “someNumbers” that includes the numbers: [1.0, 2.0, 3.0, 4.0, 5.0].

Text

Description automatically generated

In this ArrayList, findMode() returns “There is no mode in the numbers”. If the ArrayList someNumbers is modified to contain a mode, TestStatsLibrary will generate the following output:



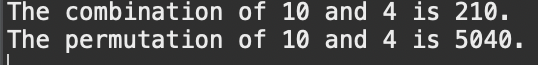
findUnion() returns the union of subset A and subset B, findIntersection() returns the intersection of subset A and subset B, and findComplement returns the complement of a subset. TestStatsLibrary uses the mentioned methods to generate the following output for the listed ArrayLists: set containing [1, 2, 3, 4, 5, 6, 7, 8, 9], listA containing [2, 4, 6], listB containing [1, 2, 5, 7, 9]

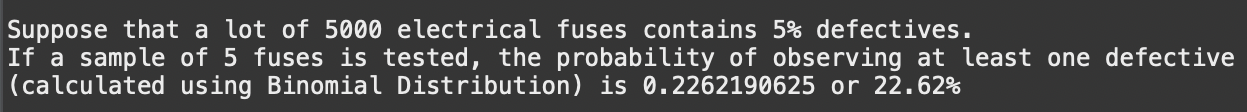
A black screen with white text

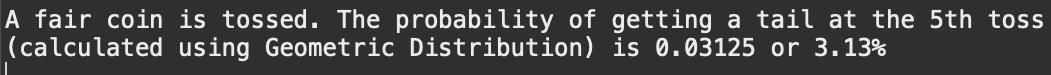
Description automatically generated with low confidence

factorial() returns the factorial of an integer. solveCombo returns the combination of two integers using the formula . solveBinoDistrib returns the binomial distribution probability using the following formula . solveGeoDistrib returns the geometric distribution using the following formulas:

TestStatsLibrary uses the mentioned methods to generate the following output:





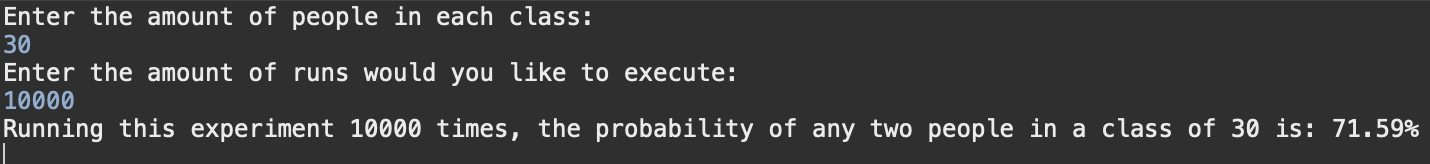


Person is an object representing a person with a randomized birthhday. It contains the method getBirthday(). To test Person and its method, TestPerson creates 30 Person objects and uses getBirthday to generate the following output:

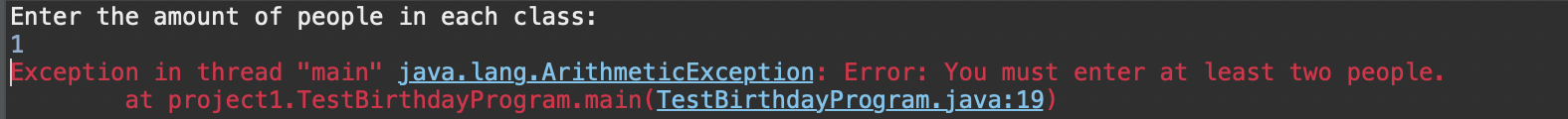
Shape

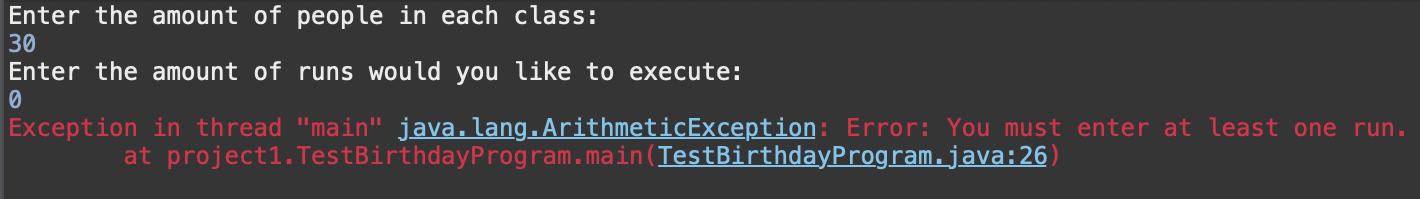
Description automatically generated with medium confidence

BirthdayProgram is a class determining the probability of any two people in the same class having the same birthday. It contains the methods personGenerator() and probCalculator(). personGenerator generates the desired amount of Person objects and returns a list of randomized birthdays. probCacluator runs the desired amount of times and returns the probability of two duplicate birthdays being generated. TestBirthdayProgram asks the user to input a number of people to generate and a number of runs to execute and output the following:



If the user inputs less than two people or less than 1 run, testBirthdayProgram will output the following errors:





DoorGame is a class that determines the probabilities involved in the Monty Hall problem. The Monty Hall problem goes as followed: "The following game was played on a popular television show. The host showed a contestant three large curtains. Behind one of the curtains was a nice prize (maybe a new car) and behind the other two curtains were worthless prizes (duds). The contestant was asked to choose one curtain. Before showing the contestant what was behind the curtain initially chosen, the game show host would open one of the curtains and show the contestant one of the duds. He then offered the contestant the option of changing from the curtain initially selected to the other remaining unopened curtain. Which strategy maximizes the contestant’s probability of winning the good prize: stay with the initial choice or switch to the other curtain?

DoorGame contains a keepDoor() method and a changeDoor() method. keepDoor determines the probability of winning the “nice prize” if the contestant decides to keep their door. changeDoor() determines the probability of winning the “nice prize” if the contestant decides to change the door. TestDoorGame runs the experiment 10,000 times using four lines of code and generates the following output:

