Seneca College

Mar 13, 2019

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444 **Demo Due date: Mar 22, 2019**
Final Code Submission Date: Mar 22, 2019

Workshop 5

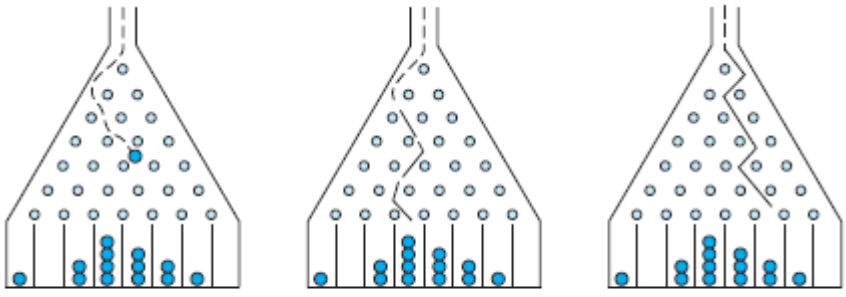
Notes:

i. Each task should be presented during the lab, demo worth 70% of the workshop marks and code uploading worth the other 30%.
ii. Make sure you have all security and check measures in place (with proper use of Exceptional Handling where ever needed), like wrong data types etc.
iii. Make your project in proper hierarchy; introduce proper class coherence in your project. Proper packages and **your project should be handled by only one main method which should be in a TesterClass**.
iv. Given output structure is just for student to have a glimpse what the output can look, students are free to make the output better in any way.

Other inputs can be given during demo, so make sure you test your program properly.

Task 1: (No JavaFX required)

The bean machine, also known as a quincunx or the Galton box, is a device for statistics experiments named after English scientist Sir Francis Galton. It consists of an upright board with evenly spaced nails (or pegs) in a triangular form, as shown in Figure.



3 Each ball takes a random path and falls into a slot.

Balls are dropped from the opening of the board. Every time a ball hits a nail, it has a 50% chance of falling to the left or to the right. The piles of balls are accumulated in the slots at the bottom of the board.

Write a program that simulates the bean machine. Your program should prompt the user to enter the number of the balls and the number of the slots in the machine.
Simulate the falling of each ball by printing its path. For example, the path for the ball in Figure (b) is LLRRLLR and the path for the ball in Figure (c) is RLRRLRR. Display the final buildup of the balls in the slots in a histogram. Here is a sample run of the program:

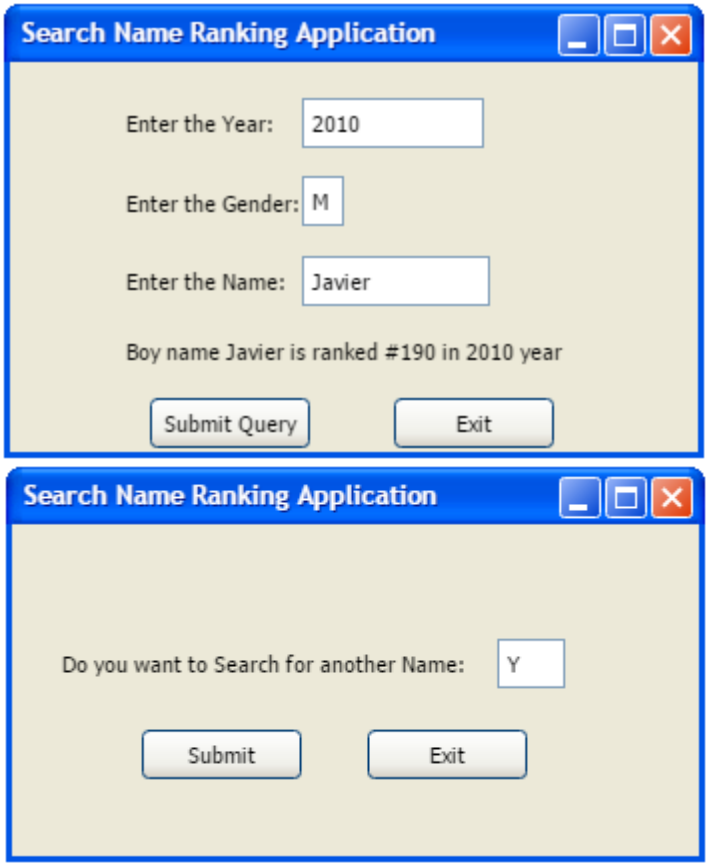
```
Enter the number of balls to drop: 5 Enter
Enter the number of slots in the bean machine: 8 Enter

LRLRLRR
RRLLLRR
LLRLLRR
RRLLLLL
LRLRRLR

   0
   0
  000
```

Task 2: (JavaFX Required)

The popularity ranking of baby names from years 2001 to 2010 is downloaded from www.ssa.gov/oact/babynames and stored in files named **babynameranking2001.txt**, **babynameranking2002.txt**, . . . , **babynameranking2010.txt**. Each file contains one thousand lines. Each line contains a ranking, a boy's name, number for the boy's name, a girl's name, and number for the girl's name. For example, the first two lines in the file **babynameranking2010.txt** are as follows:

1. Jacob 21,875 Isabella 22,731
2. Ethan 17,866 Sophia 20,477

So, the boy's name Jacob and girl's name Isabella are ranked #1 and the boy's name Ethan and girl's name Sophia are ranked #2. 21,875 boys are named Jacob and 22,731 girls are named Isabella.
Write a program that asks the user to enter the year, gender, and followed by a name, and displays the ranking of the name for the year. Here is a sample run:


Search Name Ranking Application
Enter the Year:
2010
Enter the Gender:
M
Enter the Name:
Javier
Boy name Javier is ranked #190 in 2010 year
Submit Query
Exit
Search Name Ranking Application
Do you want to Search for another Name:
Y
Submit
Exit