Computer Science 205

Project #1

The BINGO! Game

Due Date : Friday, September 22nd, 11:59 PM

50 Points

Objective

The purpose of this assignment is to acquaint ourselves with processing multi-dimensional arrays. This program will utilize loops, external file i/o, random numbers, and methods with array parameters.

The Game of *BINGO!*

Bingo is a popular game played on a 5 row by 5 column grid, called a card. There is one letter of the word B-I-N-G-O at the top of each column. In each space under each letter are randomly selected numbers :

- under **B**, 1–15
- under **I**, 16–30
- under **N**, 31–45
- under **G**, 46–60
- under **O**, 61–75.

As numbers are selected, they are marked off the card. When a line of five squares, horizontal, vertical, or diagonal, is marked out, the card is a winner. A sample bingo card is shown below :

12	28	31	49	66
3	26	45	53	75
10	17	33	59	67
7	19	42	55	74
2	23	37	46	70

If the numbers 3, 45, 53, 75, and 26 are picked at random, then there will be a horizontal bingo on the second row. Diagonal wins may be from top left corner down to bottom right corner or top right corner down to bottom left corner. No free space will be utilized.

Assignment Specifications

1. Write a program to play a game of bingo on a single card. The input to your program will be from the file `bingo.in`.

   The input will consist of a set of exactly 25 integers in row major order on separate lines representing the numbers on the bingo card.

   You will use the built-in `random()` method from the `Math` class to generate your random integers between 1 and 75 corresponding to the numbers being picked from a bingo bin. (i.e., `Math.round((Math.random() * 74 + 1))`) This process will continue until a winning pick is "called."

2. The bingo numbers that are picked should be displayed on the screen. This should be followed by a display of the winning bingo card with `X`'s used in place of called numbers. In addition, print a message to the screen indicating whether the win was horizontal, vertical, or diagonal, and how many picks from the bin it took to find a win.

3. Your program should be named `Bingo` and should be in a `Prog1` directory on your account.

Method Notes

You will need to use at least four class methods in this program. Below are some good candidates. You are strongly encouraged to break one or more of these methods into other methods. As you'll see, some of them can get long.

- A `fillCard` method to input the values from the file into your 5 by 5 card.

- A `printCard` method to print your card values out to the screen in tabular form. Any number that has been picked in the current game should be represented by an `X`. Drawing the game board and including the name "BINGO" lined up above the board are both required.

- A `playGame` method that acts as the main driver for all of your game action. It will need to contain a loop which continues to run until a win is found.

  You may want to have it call a separate method which picks a random integer between 1 and 75 that has not already been picked, and then "marks" that integer if it finds it on your card. The easiest way to mark a card is to simply change its value to 0. It should print to the screen each random number picked from the bingo bin.

- After picking a number and marking it if it is on the card, you will want to call a method called `checkForWin`. This method will sum up each row, each column, and then the two diagonals. If, at any point, you find a zero sum, you should leave this method and return your type of win. I recommend storing your win type as an integer. A non-win can be a 0, and your other wins can be global constants such as :

```
public static final int HORIZONTAL = 1;
public static final int VERTICAL = 2;
public static final int DIAGONAL = 3;
```

One potential problem spot with this method is in your summations. For your horizontal and vertical summations for this method, you are going to need to flush your sum variable to zero before each inner loop is processed. With your diagonal sums, you've only got one loop, so just reset the value before each loop.

General Notes

1. There is a sample input file and executable version of this program which you can use. You can copy them over into your area by typing:

   `cp /pub/digh/CSC205/Prog1/* .`

   You can run the sample executable program by typing `java Sample`.

2. If you obtain an array out of bounds error that usually indicates that you are trying to access an array component outside the defined indices available for your array size. Try inserting `System.err` statements at various points throughout your program to help track down the location where your program is crashing.

3. Use meaningful variable names, proper indentation, and documentation throughout. Provide comments for all important statements and variables. Below each method header, provide a precondition and postcondition.

Analysis & Design

Describe the problem at hand, and show and describe a sample input file as well as the corresponding output produced on the screen.

Break down your algorithm, and plan out all of the different class methods you plan on using. Be sure you give a plan for each of the following three tasks :

- marking a card
- checking for a duplicate pick
- determining if a card is a win

Mention the type and size of all data structures used. No module structure chart is required. Type this up into a file named `OOD.txt` in your current directory.

Sample Run

```
YOUR BINGO CARD :

   B    I    N    G    O
-------------------------
| 12 | 28 | 31 | 49 | 66 |
-------------------------
|  3 | 26 | 45 | 53 | 75 |
-------------------------
| 10 | 17 | 33 | 59 | 67 |
-------------------------
|  7 | 19 | 42 | 55 | 74 |
-------------------------
|  2 | 23 | 37 | 46 | 70 |
-------------------------

BINGO NUMBERS PICKED AT RANDOM FROM BIN :
68      18      75      10      41      9       71      57      35
20      46      43      14      32      51      36      26      66
62      28      67      65      24      59      58      23      17
54      22      29      7       12      37      4       56      19

YOU WIN WITH A VERTICAL BINGO AFTER 36 PICKS!

YOUR BINGO CARD :

   B    I    N    G    O
-------------------------
|  X |  X | 31 | 49 |  X |
-------------------------
|  3 |  X | 45 | 53 |  X |
-------------------------
|  X |  X | 33 |  X |  X |
-------------------------
|  X |  X | 42 | 55 | 74 |
-------------------------
|  2 |  X |  X |  X | 70 |
-------------------------
```

Finishing Up

We need to create a typescript file. So, at your prompt in your program directory, type `script`. Now, perform each of the following steps: display your program using `cat`, compile your program, run your program twice, and exit script by typing `exit`.