Assignment 2 – Lottery Numbers – Documentation (Max Fyall)

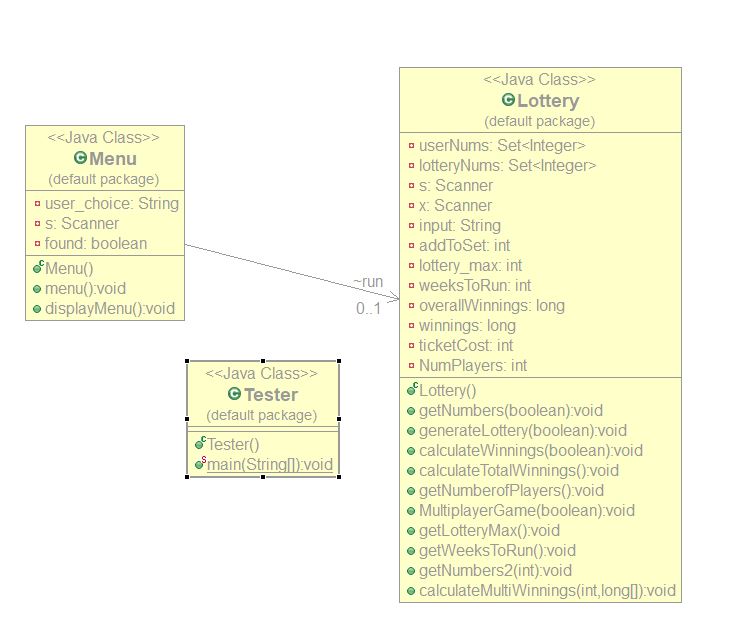
Introduction to the problem:

The aim of this assignment is to implement a functional lottery game using the set interface from the java collection classes.

Requirements:

The main 5 requirements are satisfied with the code I have submitted. The code for these requirements was successfully implemented. Requirement six was attempted and is partially complete with the submitted code. The code only allows for multiplayer capabilities if they play on a one week draw. I was unable to implement a multiple week-based draw with more than one player. This also means that I was unable to complete requirement 7 because I would have run out of time if I had wanted to implement it.

Class Design:



Pseudo Code:

getNumbers()

* Get lottery\_max from keyboard
* Validate input from keyboard
  + If input from keyboard is less than 6
  + Display error message
* Display instructions
* Get user numbers from keyboard, store in string
* Use string.split() to split the string into an array
* Start loop
* Store data in array into field
* Add to the set by using setName.add(field)
* Validate input
  + If size of set (setName.size()) is less than or bigger than 6
  + Display error message
  + Clear the set (setName.clear())
* Else
* If option 1 has been selected (found = false)
  + Print the set containing the users numbers
* If option 2 has been selected (found = true)
  + Get weeks to run from keyboard
  + Start loop
  + Call generateLottery method
  + Call calculateWinnings method
  + Clear lottery numbers set (setName.clear())

GenerateLottery()

* Create array on integers
* Create local variable count = 0
* Create object reference of random class
* Begin while loop ( set condition to size of lottery numbers set not being 6 )
  + Store random value (between 1 and lottery\_max) in array
  + Add to lottery numbers set
  + Add one to count
* End loop
  + If size of lottery set is not equal to count ( i.e. duplicate number has been added)
  + Take one away from count
* If option 1 has been selected (found = false) print the set

calculateWinnings()

* Create copy of user numbers set and store in another set
* Perform set intersection on copied set and lotteryNums set (use retainAll() method)
* Use if statements to determine what prize the user has won, by looking at the size of the copied set after the set intersection.
* Prize money is stored in a field
* If option 1 has been selected
* Then clear the user numbers set
  + If the users winnings is 0
  + Output message telling them they won nothing
  + Clear the lottery set
  + Set both fields that store the winnings to 0
  + Else
  + Output their winnings
  + Clear the lottery set
  + Set both fields that store the winnings to 0
* clear the copied set

calculateTotalWinnings()

* clear the lottery set
* calculate cost of tickets bought
* calculate total winnings
* if winnings is less than 0
  + display message that the user has won nothing
* else
* display the user’s winnings
* set fields to 0
* clear the user numbers set

MultiplayerGame()

* create array of longs called playerWinnings with the limit being the number of players
* call generate lottery method
* start loop
* create local variable count set to the counter in the for loop
* call getNumbers2 method
* call calculateWinnings method
* call calculateMultiWinnings method
* end loop
* clear lottery numbers set

calculateMultiWinnings()

* calculate ticket costs
* set int i to count from MultiplayerGame()
* calculate winnings player by taking ticket prices away from winnings
* add one to i
* clear the user numbers set
* set fields containing winnings to 0
* if i is equal to the length of the array
  + display the winnings of the players by printing out the array

Tests:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected result** | **Worked?** |
| Add data to set then print the set | 1,2,3,4,5,6 | Data is added to the set and the data is printed | Y |
| Add data to set then remove it and print it. | 1,2,3,4,5,6 | Data is added then removed and prints a blank set | Y |
| Generate Lottery Numbers, add them to the set then print them | 1,2,3,4,5,6 range of 6 | Generates lottery numbers with no duplicates and print them to the screen | Y |
| State a range | 1,2,3,4,5,6  Range of 6 | Data is processed, and the max range of the lottery is 6. Prize should be the jackpot | Y |
| Set Intersection | 1,2,3,4,5,6  Range of 6 | Print both sets, work out that reward should be the jackpot | Y |
| Set Intersection | 1,2,3,4,5,6  Range of 15 | Print Both sets, calculate what reward the user should get according to however many numbers match in each set | Y |
| Set Intersection | 1,2,3,4,5,6  Range of 30 | Print both sets, calculate if user has won anything and if they have won nothing it prints to the screen | Y |
| Run for a number of weeks | Enter 2 weeks to run for.  1,2,3,4,5,6  Range of 10 | Program keeps the same numbers for the user but generates different numbers for every week. Prints the amount the user won | N |
| Run for a number of weeks | Enter 2 weeks to run for.  1,2,3,4,5,6  Range of 10 | Program keeps the same numbers for the user but generates different numbers for every week. | Y |
| Calculate Total Winnings after number of weeks | Enter 2 weeks to run  1,2,3,4,5,6  Range of 15 | Program keeps the same numbers for the user but generates different numbers for every week. Prints the amount the user won after calculating the price of tickets | Y |
| Run a multiplayer game | Enter weeks to run 2  Enter players 2  Player 1: 1,2,3,4,5,6  Player 2: 1,2,3,7,8,9  Range of 10 | Output results after doing calculations | N |
| Run a multiplayer Game without the weeks function | Players 2  Player 1:  1,2,3,4,5,6  Player 2:  1,2,3,7,8,9  Range of 10 | Output result after doing calculation | Y |

Evaluation:

This assignment was another good test of my coding skills. Getting to use the java library classes made this week’s assignment a whole lot easier. It made it so I had to do less coding. This did mean however I had to think more carefully about how I was using the methods provided to me by the java class libraries. This has assignment has once again broadened my programming knowledge which can allow me to use many different data structures in my programming future. The main section of this assignment was completed relatively easily. The optional extras were always challenging as they are meant to be, I only managed to partially complete one of the optional extras. I do believe if I put more time into the optional extras I would have completed them. When I was implementing my code, I was coming across a lot of logical errors which were very frustrating to fix as I was finding it very difficult to spot them in my code. This was most certainly the case when I was trying to implement the multiplayer draw over a certain number of weeks. The experience with these logical errors will no doubt help me spot them easier in the future when I am programming far more complex algorithms.