**AC12001**

Name: …Miklos Mayer…………………………………………………………………….…..

Matric number: …190017778 …………………………………………………….……….

Lab Title: …AC12001 assignment 2: Sets………………………………..…………....

Test number/date/version: 17/02/20 ……………………………………….…..…..

Test Notes: …Tests run with JUnit5 ……………………….……………………….……

Summary:

The program is a simulation of a lottery system. Multiple people can play it, they can define the range of the bettable numbers and the duration of their ticket (how many weeks they would like to use the ticket).

Requirements:

* The program shall take inputs from the user(s) for:
  + The range for the lottery (0 - ...), bigger than 6
  + How many weeks they would like to play
  + For a name for each player
  + 6 numbers for each lottery tickets
* Lottery ticket shall not have duplicate numbers or numbers out of the lottery’s range
* The lottery system shall generate 6 random winning numbers for each week and check if any of the tickets has won
* The system shall display who won and how much
* At the end of the game the system shall calculate and display how much the player(s) earn or lose based on their ticket prices and winnings.
* The system should reward a match 3 with 25, a match 4 with 100, a match 5 with 1000 and a match 6 with 1000000 pounds
* A ticket’s price should be 2 pounds

Class designs:

* Menu
  + fields:
    - pound: Currency – The currency used in the lottery
    - in: Scanner – The stream through the user communicates with the program
    - lotterySystem: LotterySystem – The lottery game
    - weeks: int – The number of weeks to run the lottery for
    - players: Set<Player> - The players of the lottery
  + methods:

Pseudocodes:

1. Take the range for the lottery from the user until they provide a valid whole number bigger than 6
2. Take input from the user for the number of weeks to play, until they provide a whole number bigger than 0
3. While there is a next player:
   1. Take input from the player for their name
   2. Take input from the player while they provide 6 whole number in the lottery system range for their ticket
4. Run the lottery for the weeks defined previously by the user
   1. Generate 6 different numbers within the lottery’s range
   2. Check every bought ticket if they have 3 or more match with the week’s winning numbers
   3. Display the winners
5. Display the total costs and earnings of each player

Test results:

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Description** | **Test Data** | **Expected result** | **Worked?** |
|  | For all tests if not stated other:  lotteryMax = 10 |  |  |
| Create a new ticket with 6 different numbers | 1, 2, 3, 4, 5, 6 | The new Ticket is created | Y |
| Create new ticket with 5 different numbers | 1, 2, 3, 4, 5 | Exception’s thrown | Y |
| Create a new ticket with 7 different numbers | 1, 2, 3, 4, 5, 6, 7 | Exception’s thrown | Y |
| Create a new ticket with a negative number | -1, 1, 2, 3, 4, 5 | Exception’s thrown | Y |
| Create a new ticket with a greater number than lotteryMax | 1, 2, 3, 4, 5, 11 | Exception’s thrown | Y |
| Create a new ticket with a 0 number | 0, 2, 3, 4, 5, 6 | Exception’s thrown | Y |
| Generate a random ticket 100 times | None | The System creates a random Ticket with a set of 6 numbers between 1 and lotteryMax | Y |
| Run the lottery 10 times with lotteryMax = 10 | 10 10 Arcfej 1 2 3 4 5 6 n | The system should display in the end how much the user has spent and won. | Y |
| Run the lottery 100 times lotteryMax = 10 | 10 100 Arcfej 1 2 3 4 5 6 n | The system should display in the end how much the users have spent and won. | Y |
| Run the lottery for 3 users 10 times lotteryMax = 10 | 10 10 Arcfej 1 2 3 4 5 6 y Second 2 3 4 5 6 7 y Third 10 9 8 7 6 5 n | The system should display in the end how much the users have spent and won. | Y |
| Run the lottery for 3 users 100 times lotteryMax = 10 | 10 100 Arcfej 1 2 3 4 5 6 y Second 2 3 4 5 6 7 y Third 10 9 8 7 6 5 n | The system should display in the end how much the users have spent and won. | Y |
| Run the lottery 10 times with lotteryMax = 100 | 100 10 Arcfej 1 2 3 4 5 6 n | The system should display in the end how much the user has spent and won. | Y |
| Run the lottery 100 times lotteryMax = 100 | 100 100 Arcfej 1 2 3 4 5 6 n | The system should display in the end how much the users have spent and won. | Y |
| Run the lottery for 3 users 10 times lotteryMax = 100 | 100 10 Arcfej 1 2 3 4 5 6 y Second 2 3 4 5 6 7 y Third 10 9 8 7 6 5 n | The system should display in the end how much the users have spent and won. | Y |
| Run the lottery for 3 users 100 times lotteryMax = 100 | 100 100 Arcfej 1 2 3 4 5 6 y Second 2 3 4 5 6 7 y Third 10 9 8 7 6 5 n | The system should display in the end how much the users have spent and won. | Y |

Self-evaluation:

Actually, I don’t have much to report. One of the challenges was to mock user inputs in JUnit tests. Unfortunately, I didn’t have time to implement my own Set, I just extended HashSet and added a new function for looking up for intersections between the instance of the class and another set.