ELEC 2543 Object-Oriented Programming and Data Structures

Exercise 5

Topics: Methods and Assignment 1

Due Date: Mar 2, 2017

Overview: This lab exercise aims at helping you to get prepared for Assignment 1. You will develop an extended version of the Random Walk Game (a very simplified Monopoly game) that you developed in Lab 3a.

The ERW (extended random walk) Game

*Overview*

There are three players, each with the same amount of cash when the game starts. They all walk on a circular path. In each round, they all make a move. The distance to move is determined by rolling a die. Depends on what location a player moves to, the player may have to pay rent or get some cash. The game ends when at least one player is broke (with –ve cash balance). In this ERW game, there is no “bank”.

*The Path*

The path composes of a “GO” lot, and *size*-1 other lots with properties on them. The “GO” lot is numbered as zero, while the other lots are numbered from 1 to *size*-1. It is a circular path, and the players move from a smaller numbered lot to a higher numbered lot. When a player is now at lot *size*-1 and it moves one step, it will be in the GO lot.

*The Players*

There are three players, and their ID’s are 0, 1, and 2, respectively. Each player starts at the GO lot. Each has 300 dollars in the beginning.

*In each Round*

In each round, P0, P1, and P2 take turn to make a move. In each move, the player first rolls a die. The range of distance is [1, 6]. It then advances to another lot according to the face value of the die.

* If the new position is the GO lot, the player gets 100 dollars.
* If the new position is not the GO lot, the player has to pay rent. The rent is 10\**k* dollars where 1 ≤ *k* ≤ 10 (*k* is randomly generated). That is, the cash amount of the player is reduced.

*Game Termination*

The game ends when one or more players have negative cash balance after each making a move in that round.

*Output*

Before the game starts, after each round, and after the game ends, the positions and the cash balances of the players have to be printed out. The path is printed in a linear manner from the GO lot, with the player position indicated by the player ID. A separate line for each player.

In each round, apart from the above information, the rent/money each player pays/gets should be printed out as well. Sample outputs of *size* = 10 are provided in Moodle.

Implementation

*Class Files*

ERWPlayer.java: Class file for players. The instance variables and constructor have been defined. The printStatus() method prints out the information of the player. You do not have to use it if you do not want to.

ERWPath.java: Class file for the path. The instance variable, getter method, and the constructor have been defined.

RandomDie.java: Class file for rolling dies. It is the same as the one you used in Lab3a. The classname has been simplified for convenience. You should not modify this file.

ERWGame.java: Class file for the game. The instance variables and constructor have been defined. The skeleton of method play() has been provided.

ERWGameDriver.java: Driver program of the game for you to test your game.

*Make a move*

**You have to develop a method that simulates a certain player makes a move.** That is, after calling this method, the cash amount and the position of the player will be updated appropriately. You should apply object-oriented design to determine where to put this method. You may develop other methods if needed.

*Instance variables*

The instance variables provided should be sufficient for developing the whole game. In case you want to declare other instance variables, you have to provide a document to explain what they do.

Handin

Submit your ERWPlayer.java, ERWPath.java, and ERWGame.java to Moodle before the deadline. In case you have declared other instance variables in any of the class files you submit, you have to submit another pdf file that explains what it does. You do not have to submit any pdf file if you do not declare any new instance variable.

We will use our own driver program to test your submission.