Programming Assignment #02 – King of the Stacks

Due: Tuesday, October 20th at 5:00 PM

You must upload a **single ZIP file** containing all of your **JAVA source code** files on the course Moodle site by the start of class on the due date.

Suppose you are designing a game called King of the Stacks. The rules of the game are as follows:

- The game is played with **two (2)** players.
- There are **three (3)** different Stacks in the game.
- Each turn, a player pushes a **disk** on top of exactly one of the three Stacks. Players alternate turns throughout the game. Each disk will include some marker to denote to whom it belongs.
- At the end of certain turns, spaced at regular intervals, the top disk is automatically popped from the Stacks. The pop timer is staggered so that disks are popped off from different Stacks at different times.
- The game is played for a set number of turns (**N**), but there must be at least 20 turns.
- After **N** turns have elapsed, the game is over. The player who has the most disks remaining on the three Stacks combined is the winner.

Your task is to write a program that will implement and simulate the King of the Stacks game.

Some requirements for your simulation:

- Your program should ask how many turns should be played (N >= 20)
- The game will be simulated according to the rules listed above
- Your program should simulate the entire game **without any user intervention** after the number of turns is entered. Each player's turn will be simulated by having a random integer (0, 1, 2) generated. Based on the random integer drawn, the player will push her disk onto the appropriate Stack.
- The pop timers should be staggered so that a disk is popped from each Stack at the end of every 3rd, 5th, and 7th turns, respectively. (In other words, Stack A will pop off a disk every 3 turns, Stack B will pop off a disk every 5 turns; and Stack C will pop off a disk every 7 turns).
- Your simulation should include a way to handle a potential EmptyStackException that is thrown by attempting to pop an empty Stack.

- Your program should include print statements stating which turn it is and describing the events of each turn (Player _ push disk onto Stack _; A disk was popped from Stack __; etc.)
- At the end of the game, all of the disks should be popped off from each Stack and scores tallied for each player. The results should be announced via a print statement.

Some ground rules:

- Your simulation should work for **any** of the implementations of the Stack ADT. If one were to change the driver to use a different Stack implementation class, your simulation should still work.
- Your program should be generalized where possible so that the simulation can easily be modified (e.g., number of turns played, pop timers for each stack, etc.)
- Object-oriented principles must be followed; define classes to represent game objects as needed.
- Functional decomposition must be followed and helper methods be used where appropriate.
- Your program should be well organized and properly commented.
- You must test your program to confirm that it works.

Upload your source code files (.java files) to the course Moodle.