CSC 103 Programming Assignment #3 02/18/16

Due date – **Friday November 18th at Noon**

This project is to be done by a pair (2) of students. No groups of three or one without the instructor’s permission. The phrase “I like to work alone” is not an acceptable reason for not having a partner. You will need to learn to work together. Make sure that both of you understand all aspects of the assignment. Note: If the work is not evenly done, this will no doubt be seen during exams.

**Project # 3 Airport Runway Simulation**

Your task is to create the classes that are discussed in chapter 7, programming project 9 on pages 407-408 of your textbook.

You are to create a Runway Simulation class, this should be partly modeled by the Car Wash simulation that was shown in class and is discussed in the textbook on pages 369-382.

Your simulation will start with 6 input values (which the user will need to input). These six inputs are:

* Total length (in minutes) of simulation
* Total time needed for a single takeoff (in minutes)
* Total time needed for a single landing (in minutes)
* Average number of minutes between planes arriving for landing (this means that the percentage of probability that a plane arrives at any given minute is one divided this number)
* Average number of minutes between planes arriving for takeoff
* Maximum number of minutes before landing planes will run out of fuel and crash.

Your simulation should report output of status for each minute of simulation (see example output). At the end of the simulation you should report the following:

* Number of planes that tookoff
* Number of planes that landed
* Number of planes that crashed
* Average waiting time for takeoff
* Average waiting time for landing

You should also create a stack of the crashed planes storing the time during the simulation that they crashed. They should be printed out in reverse order because you placed them on a stack. ( again see output)

You will create two Queues, one for planes that wait to takeoff, and one for planes that wait to land. Each minute of the simulation, you will check if new planes arrives for either status and put them in the Queues. For each minute that the runway becomes available, choose a plane waiting from one of the queues. But always first look to any planes waiting to land, as they have priority, because they may crash. If there are none in landing queue, then take from takeoff list. Once a plane is on the runway, it is unavailable for use until takeoff or landing is completely finished.

You are to make use of the following classes that we have discussed for the last two classes without any changes. These classes are stored in the project folder.

* **LinkedQueue**
* **LinkedStack**
* **Node** – used by the first two
* **BooleanSource**
* **Averager**

You will create these classes to finish the project:

* **Runway** (see below)
* **Plane** (mostly complete - see below)
* **RunwaySimulation**- main driver of program.

The Runway class is modeled after the Washer class for the carwash, but will have additional features. You will use this class for both takeoffs and landings. The template of this class is below:

The Plane class allows us to store info about each plane in line, and this is the Object type that will be stored in both Queues. The code for this class is below:

Class Runway{  
 private int timeForLanding;  
 private int timeForTakeoff;  
 private int runwayTimeLeft;   
 private char operation;   
// operation can be: I – Idle, L-Landing, T-takeoff  
  
 public Runway ( int time\_takeoff, int time\_landing){…}  
//set the time for landing, time for takeoff, and the //operation to idle.   
  
 public boolean isBusy() {…}  
  
 public void reduceRemainingTime(){…}  
  
 public void startUsingRunway(char typeOfUse){…}  
// if typeOfUse is 'T' - then the operation is take off //and set the runway time left   
// to the time it takes for takeoff.  
// if typeOfUse is 'L' - then the operation is landing and //set the runway time left   
// to the time it takes for landing  
// if typrOfUse is ‘I’ – then the runway is idle, set the //runway time left to zero  
  
 public char kindOfOperation() {…}   
// returns the type of operation the runway is used for.   
// returns 'L' if the runway is used for is landing.   
// returns 'T' if the runway is used for taking off.   
// returns ‘I’, if the runway is idle   
  
}//end Runway

class Plane {  
 static private int planeCount = 0;   
 // the plane number arrived to the queue   
 // should be in incrementing order  
 private int time; //the time the plane arrived in queue  
 private char operation; // the kind of operation the //plane is doing 'L" is for landing  
 // 'T' is for taking off   
 private int planeNo; // plane number  
  
 public Plane( int aTime, char landingOrTakeOff)  
// operation is the type of operation the plane is doing. // If landingOrTakeOff is 'L' it means the plane is landing // If landingOrTakeOff is 'T' it means Taking off.   
 {  
 time = aTime;  
 operation = landingOrTakeOff;  
 planeNo = ++planeCount;  
 }  
  
 public int getTime() {  
 return time;

}  
 public int getPlaneNo () {  
 return planeNo;

}  
 public char getOperation () {  
 return operation;

}  
   
 private static int getPlaneCount()  
 {  
 return planeCount;  
 }  
  
}// end Plane

Tips for good grades:

* Make sure you use comments where needed and use variable names that make sense, some of your grade will depend on program style as well as the use of your program.
* Update the comments in the class file, to include your names and any new information
* You will lose points for things like not indenting, or naming variables in non-descriptive ways. Do no leave in debugging code, or commented out code.
* I use jGrasp and the java version that is in the lab computers. So make sure that your programs work with this.
* Test your own projects thoroughly before you hand them in.
* Late projects will not be accepted so plan ahead.

The five classes that are completely created have to be used as is. I will be running your programs with my own copies of these classes, so if you make any changes to them it will not work when I run it. These classes are:

* **LinkedQueue**
* **LinkedStack**
* **Node** – used by the first two
* **BooleanSource**
* **Averager**

The other three classes are yours to write as you like, each one in a separate file. Name them **RunwaySimulation.java, Runway.java** and **Plane.java**. You are allowed to create additional classes also if you feel that you need them. If you do not name these files correctly, you will lose points. Use Javadoc to create the documentation for your ADT class.

Hand in electronically – (NOT E-mail!!!)

In S-drive CSC 103 folder:

1. Create folder called **projectthree\_firstname\_lastname**
2. Place these three script files named above in folder.
3. Place any other class files that you created.
4. Place any other documentation such as a readme file in here as well.

Sample run of working program:

The time of simulation: 30 minutes  
The time needed for a takeoff: 2 minutes  
The time needed for a landing: 3 minutes  
The average time between arrivals for takeoffs: 8 minutes  
The average time between arrivals for landings: 4 minutes  
The maximum time a plane can stay in air before crashing: 4 minutes

During minute 1 :  
 Runway: Idle  
During minute 2 :  
 Arrived for Takeoff : Plane # 1  
 Runway: Plane # 1 is taking off   
During minute 3 :  
 Runway: Plane # 1 is taking off (finishing)  
During minute 4 :  
 Runway: Idle  
During minute 5 :  
 Arrived for Takeoff : Plane # 2  
 Arrived for Landing : Plane # 3  
 Runway: Plane # 3 is landing   
During minute 6 :  
 Arrived for Landing : Plane # 4  
 Runway: Plane # 3 is landing   
During minute 7 :  
 Arrived for Landing : Plane # 5  
 Runway: Plane # 3 is landing (finishing)  
  
.  
.  
.

.  
  
During minute 27 :  
 Arrived for Takeoff : Plane # 12  
 Runway: Plane # 12 is taking off   
During minute 28 :  
 Runway: Plane # 12 is taking off (finishing)  
During minute 29 :  
 Runway: Idle  
During minute 30 :  
 Runway: Idle  
  
Crashed Planes:  
 Plane # 8 crashed at time : 16  
 Plane # 6 crashed at time : 12

Number of planes that came to runway for takeoff: 4  
Number of planes that came to runway for landing: 8  
Number of planes that crashed: 2  
Average time waiting in takeoff queue: 3.75  
Average time waiting in landing queue: 1.67