CSE 201: Advanced Programming Monsoon 2017

Lab Assignment-6
(Exception Handling)

IIIT-Delhi. 7th Sept 2017. Due by 23:59pm on 8st Sept 2017

Instructor: Vivek Kumar

No extensions will be provided. Any submission after the deadline will not be evaluated. If you see ambiguity or inconsistency in a question, please seek a clarification from the teaching staff.

NOTE: Attendance in the lab is mandatory.

Plagiarism: All submitted homeworks are expected to be the result of your individual effort. You should never misrepresent someone else's work as your own. In case any plagiarism case is detected, it will be dealt as per IIITD policy for plagiarism.

NOTE: You will have to create a "PRIVATE" git repository to manage your code for this lab assignment. This is a part of evaluation criteria and will be checked by TA during the demo. The timestamp on the git log should be before the deadline.

Problem Description:

In this lab, you need to help the knights to find the queen in 2D grid. Each knight has a name which is unique and a magic box. There are rules so that knights can move along the grid. Also there are total number of iterations such that at each iteration knights can make a move -

1. The starting coordinates of every knight is known. Also the coordinates of the queen is known.
2. The magic box that each knight have is a **stack** (inbuilt stack is allowed) which can contain any value - String, Float, Integer or Coordinate. Only coordinates are considered to be valid credentials. For the rest of the values popped you need to **throw an exception.**
3. At each iteration the knights start popping the value from their stack (assume stack to be filled with values) . If the popped value comes out to be coordinates then the knight moves to that coordinate else it stays at its current position.
4. If the stack of the knight is empty and he has not found the queen, you again need to throw an exception and that knight is removed from the grid.
5. If a knight (say knight B) pops the coordinates at the ith iteration and at that coordinate already a knight is present (say knight A) (from the i-1th iteration or if knight A made a move before knight B in the same iteration). Then the knight currently entering the coordinates will tell the other knight to leave the grid. (Knight B's coordinates will be updated and knight A will leave the grid). You need to throw an exception here as well.

6. At each iteration the order in which knights will make a move (a value is pop from their stack) is according to the lexicographic order of their name. (first knight A then knight B and so on).
7. If a knight finds the queen (that is his updated coordinates coincide with the queen's coordinates) then you need to throw an exception and then abort the iterations.

Note: Game might end even if Queen is not found, if number of iterations are completed or there is no Knight left.

Input:

For each knight you need to read the .txt file. If there are n knights, then there are n **.txt** files each having the description of a knight. Names of file are 1.txt , 2.txt and so on.

The format of the knight description in **.txt** file is:
First line contains a string which denotes the name of the knight.
Second line contains start coordinates of knight x and y
Third line contains number of values (m) in the magic box of the knight.
Next m lines contain type of the value and value to be pushed in the stack.
Note: The values are pushed in the stack in the order they are provided in m lines.

From user, following input must be taken:

1. The number of knights
2. The total number of iterations so that knights can make a move at each iteration.
3. Coordinates of queen x and y

Output:

Output has to be printed in a .txt file.
For each knight at every iteration you need to print the following:
First line is as follows:
<Iteration number> <Name Of Knight> <current_location (x y)>
The second line will depend on the exception. For the exceptions you have to print the custom exception class for that exception and then the exception message.
Hint: You need to throw the exception message from the corresponding custom exception class:

1. If no exceptions print **"No exception" <coordinates popped (x y)>**.
2. If popped value not a coordinate, "**NonCoordinateException: Not a coordinate Exception**" <**Value popped**>.
3. If Stack gets empty "**StackEmptyException: Stack Empty exception**".
4. If the knights new coordinate overlap with other knights coordinates "**OverlapException: Knights Overlap Exception**" <**Name of the other knight**>
5. If queen is found "**QueenFoundException: Queen has been Found. Abort!**"

Test Case:

A test case has been provided. Take the following user input from user for input files in the test case:

1. The number of knights - **3**
2. The total number of iterations - **10**
3. Coordinates of queen x and y **- 4 5**