UNIVERSITY OF ALBERTA

Augustana Computing Science 112

Assignment #5

Goals:

1. Understand linked lists. You should be able to draw what is happening in RAM with your code and in the provided code.
2. Work in an abstraction. You should be able to have a picture, in your mind, of what is happening in your code.
3. Use a library where you can see the Java code, to Bartelop a clear understanding of the connections between files and data types.

Reference:

Heise, Chapter 5

Instructions:

In this assignment you must implement a simulation of the children's game "Duck, Duck, Goose." This is an INDIVIDUAL assignment: you must write entirely your own program. It must contain the doubly-linked, circular list structure available on eClass, `DCircLinkList.java`). Regular documentation and modularity requirements are in effect. No libraries, except Scanner and Random, may be imported.

How to play "Duck Duck Goose":

A group of children sit in a circle (ok, adults have been known to do this too). One child is "it" and remains standing on the outside of the circle. The "it" child will walk clockwise around the outside of the circle, touching the head of each child that is passed and calling out "Duck". At some point in time, the "it" person calls out "Goose". This creates the game situation: the "it" person and the "Goose" person now race around the circle to see who can get back to this position first. The "it" person runs clockwise. The "Goose" person runs counter-clockwise. Whoever returns to the position where the goose was sitting wins the spot. The other child becomes the "it" person for the next round. Note that there are many variations on this game, including that the "it" person and the "Goose" person run in the same direction. For this assignment, the "it" person and the "Goose" person must run in opposite directions, as described.

Specific Requirements:

Create a "Duck Duck Goose" project that will prompt the user for the number of players and their names. You must create a doubly-linked, circular list with these players, so they are in the order in which they are entered and the last person entered becomes the first "it" person. The player entered first must be in the location where the "it" person starts, and the player entered 2nd will be directly clockwise from there, and so on. You may assume that the number of players and their names will be entered correctly (as a positive integer and Strings, respectively), so you do not need to consider exception handling.

Ask the user how many rounds of the game they want, and then make that happen. Generate a random number between 1 and 20 (inclusive) to give the number of times the "it" person calls "Duck" before calling "Goose". Print the name of each person that the "it" person passes, in a clockwise direction, and the word that is called out (either "Duck" or "Goose"). Show the game circle at the start of each round, with the assumption that the "it" person is located at the name that is listed first in the game circle.

When “Goose” is called, the following must be done:

- Remove the goose from the game circle, because the goose gets up and starts running
- Print the new game circle (with the goose removed)
- Simulate the goose running counter-clockwise and the “it” person running clockwise. Do this by generating random numbers between 1 and the number of players in the circle, and making the person with the higher random number run “first” (since we are not doing parallel computing in this course). Keep repeating this process of generating and running until someone gets back to the position. During the running, print out the generated random numbers and who has run past whom. A sample of the output that is expected is given below.

When all rounds are finished, print the final game situation.

Make maximum use of the `DCircLinkList` class. Only write your own code when necessary. Write all of your own code in one file, called `Main.java`. We are expecting you to be creating several methods within this file, not writing it all in `main`.

Sample Output (You must have this format):

```
How many players?
7
Please enter Player 1's name:
Nikolett
Please enter Player 2's name:
Isaac
Please enter Player 3's name:
Zara
Please enter Player 4's name:
David
Please enter Player 5's name:
Bart
Please enter Player 6's name:
Keven
Please enter Player 7's name:
Gilbert

How many rounds?
5

++++++++++++++++++++++++++++++++++++++++++++

Round 1
Game circle:  (Nikolett, Isaac, Zara, David, Bart, Keven)
It-Person:  Gilbert
========================================
Random number generated is: 19
Nikolett duck; Isaac duck; Zara duck; David duck; Bart duck; Keven
duck; Nikolett duck; Isaac duck; Zara duck; David duck; Bart duck;
Keven duck; Nikolett duck; Isaac duck; Zara duck; David duck; Bart
duck; Keven duck; Nikolett duck; Isaac GOOSE

Up jumps:  Isaac
Game circle:  (Zara, David, Bart, Keven, Nikolett)
   **Speeds:  It-Person 3, Goose 2
      It-Person running past: Zara David Bart
      Goose running past:     Nikolett Keven

   **Speeds:  It-Person 4, Goose 3
      It-Person running past: Keven Nikolett
```

Prompt for number of players. Guaranteed this will be a positive integer.

Tell the user which player is being entered by numbering the prompts.

Last player entered becomes the first “It Person”.

Prompt for number of rounds. Assume valid positive integer.

Tell round number and game state.

Show random number and who gets called “duck” and then “GOOSE” (19 ducks then one GOOSE).

Show all players run past.

Here is the game play. Show speeds (which are also how many players are run past).

```
      Goose running past:     Bart David Zara

It-person (Gilbert) wins
++++++++++++++++++++++++++++++++++++++++++++
```

Show winner of each round.

```
Round 2
Game circle:  (Gilbert, Zara, David, Bart, Keven, Nikolett)
It-Person:  Isaac
========================================
Random number generated is: 13
Gilbert duck; Zara duck; David duck; Bart duck; Keven duck;
Nikolett duck; Gilbert duck; Zara duck; David duck; Bart duck;
Keven duck; Nikolett duck; Gilbert duck; Zara GOOSE

Up jumps:  Zara
Game circle:  (David, Bart, Keven, Nikolett, Gilbert)
   **Speeds:  It-Person 3, Goose 5
      Goose running past:     Gilbert Nikolett Keven Bart David
      It-Person running past: David Bart Keven

Goose (Zara) wins
++++++++++++++++++++++++++++++++++++++++++++
```

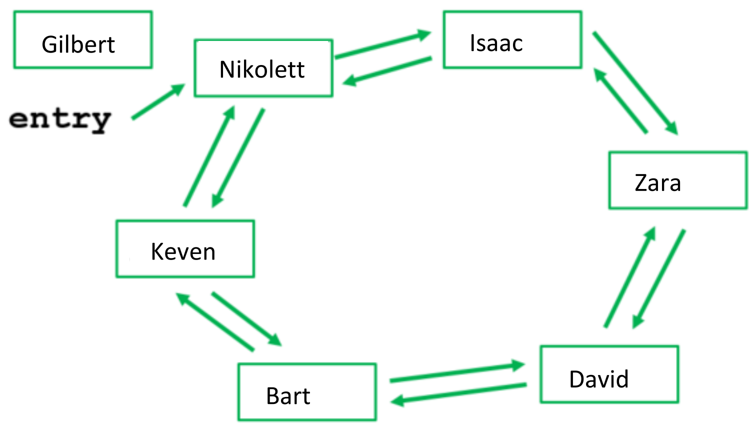
The player with the highest speed goes 1st, but both players run.

Separate rounds clearly.

```
Round 3
Game circle:  (Zara, David, Bart, Keven, Nikolett, Gilbert)
It-Person:  Isaac
========================================
Random number generated is: 7
Zara duck; David duck; Bart duck; Keven duck; Nikolett duck;
Gilbert duck; Zara duck; David GOOSE

Up jumps:  David
Game circle:  (Bart, Keven, Nikolett, Gilbert, Zara)
   **Speeds:  It-Person 5, Goose 3
      It-Person running past: Bart Keven Nikolett Gilbert Zara
      Goose running past:     Zara Gilbert Nikolett

It-person (Isaac) wins
++++++++++++++++++++++++++++++++++++++++++++

Round 4
Game circle:  (Isaac, Bart, Keven, Nikolett, Gilbert, Zara)
It-Person:  David
========================================
Random number generated is: 6
Isaac duck; Bart duck; Keven duck; Nikolett duck; Gilbert duck;
Zara duck; Isaac GOOSE

Up jumps:  Isaac
Game circle:  (Bart, Keven, Nikolett, Gilbert, Zara)
   **Speeds:  It-Person 1, Goose 4
      Goose running past:     Zara Gilbert Nikolett Keven
      It-Person running past: Bart

   **Speeds:  It-Person 3, Goose 5
      Goose running past:     Bart
      It-Person running past: Keven Nikolett Gilbert

Goose (Isaac) wins
++++++++++++++++++++++++++++++++++++++++++++
```

```
Round 5
Game circle:  (Isaac, Bart, Keven, Nikolett, Gilbert, Zara)
It-Person:  David
========================================
Random number generated is: 19
Isaac duck; Bart duck; Keven duck; Nikolett duck; Gilbert duck;
Zara duck; Isaac duck; Bart duck; Keven duck; Nikolett duck;
Gilbert duck; Zara duck; Isaac duck; Bart duck; Keven duck;
Nikolett duck; Gilbert duck; Zara duck; Isaac duck; Bart GOOSE

Up jumps:  Bart
Game circle:  (Keven, Nikolett, Gilbert, Zara, Isaac)
   **Speeds:  It-Person 2, Goose 1
      It-Person running past: Keven Nikolett
      Goose running past:     Isaac

   **Speeds:  It-Person 1, Goose 1
      It-Person running past: Gilbert
      Goose running past:     Zara

   **Speeds:  It-Person 2, Goose 5
      Goose running past:     Gilbert Nikolett Keven
      It-Person running past: Zara Isaac

Goose (Bart) wins
++++++++++++++++++++++++++++++++++++++++++++

Done.  Final State.
Game circle:  (Bart, Keven, Nikolett, Gilbert, Zara, Isaac)
It-Person:  David
========================================
```

Initial Picture of Game Entered in the Sample:



Hand in (remember this is an INDIVIDUAL assignment):

1) Your Java code: **`DuckDuckGoose.java`**
2) A **`.pdf`** of your Java code, called **`DuckDuckGoose.pdf`** (Do NOT have lines wrapping)
3) A **`.pdf`** of your program running 15 rounds with 11 players, called **`output.pdf`**. Make sure this is highly readable, and demonstrates what your program is capable of doing. Do not edit your output, but make sure that the font you submit is fixed size (e.g. courier).

Did you understand?

These are self-check questions; do not hand in solutions.

1. Can you draw what is happening to the linked list?
2. Can you draw what memory (RAM) looks like with a linked list?
3. Can you explain each of the methods in the `DCircLinkList` class?
4. Explain how to write a method that is given a particular node (not the entry node) of a `DCircLinkList` and removes that node from the list. Ensure that entry is not changed, unless it happens to be the node removed.
5. In which file would the previous question's method best reside?
6. How is the `size` field in `DCircLinkList` related to the `length` of an array?
7. How is the `getSize` method in `DCircLinkList` related to the `length` method of Strings?
8. Describe what a dot ('.') does in a Java command.
9. How does a `DNode` differ from a node that is only singly-linked?
10. What is an element?
11. What is a generic type? Explain using an example from this project.
12. What is null? Give two examples.
13. How many methods did you create in `Main.java`? Could you see places where you could make even more methods?
14. What is the handshake for a method?
15. What is the difference between a parameter and a local variable?
16. What is the difference between a local variable and an instance variable?
17. Did you make sure that variables are local as much as possible?