Programming Assignment #1

Building a Single-person-Social Network Application

Due date: Feb,2, 2:00PM
URL: http://www.cs.colostate.edu/~cs200

A. Objective

In this assignment, you will build classes to manage information for a single user. Your system will maintain basic information for a user such as their name and postings by the user on their *wall*.

B. Description of Task

You will be creating 5 classes: `Member, Edge, EdgeStack, Item, InformationParser,` and will make modifications to one class: `PA1`.

(1) `Member`

The `Member` class includes the following information for each user:

- `userID`: unique id for this user
- `first name`: first name of this user
- `last name`: last name of this user
- `edgestack`: a class maintaining a stack of edges created by this user.

(2) `EdgeStack`

EdgeStack is a class that maintains a data structure to store and retrieve edges. This class should implement the following stack operations: create(), isEmpty(), push(), peek(), pop(), and destroy().

(3) `Edge`

The `Edge` class maintains information about an edge. This includes:

- `item`: object produced by the current action (e.g. a newly added comment)
- `reference`: the target that the posting refers to (e.g. a posting that a comment has been added) For the "create posting" type, the homepage will be the target object. For the "click like" edge, comment object or wall posting object can be a target object.

(4) `Item`

The `Item` class maintains information about any items shown on the News Feed page. This class is used for the newly posted item and related items. This class should include the following information:

- `itemID`: unique id for this posting object.
- `type`: (1) wall posting, (2) comment, (3) homepage (4) status change (5) like (6) visit
- `creator`: member ID of the original creator of this posting
- `message`: the content of this posting. If it is a wall posting or comment, it will be a text that is posted on the wall or added to the comment. If it is a homepage, it will be the URL of the homepage. If it is a status change, it will contain a new status. For the "like" posting (e.g. clicking like button) or "visit", this will be empty.
- `age`: the age of this posting (in minutes)

(5)InformationParser

The `InformationParser` class is for loading and parsing the example input file to build aforementioned objects. **The first edge specified in our example file is stored on the bottom of the stack and subsequent edges are stacked on top of previous ones.**

The example input file of this assignment follows the following format:

```
member [member id] [first name] [last name]
edge
item[itemID][type][creator][message][age]
reference[itemID][type][creator][message][age]
edge
item [itemID][type][creator][message][age]
reference[itemID][type][creator][message][age]
.
.
.
```

`member, edge, item,` and `reference` are indicators of starting a record. Multiple records are separated by a linefeed. Each of the records contain one or more fields. Multiple fields are separated by a **space**.

For example, the input file below represents a record for the user with userID:eh0721, first name: Earnest, last name: Hemingway and one edge created by this user. This edge includes two item objects: item and reference. First, item has itemID: 2, type: 4-status change, creator: eh0721, message: "I have arrived in LAX.", and age (3600 seconds, 1 hour). The reference follows a similar structure. The edges are listed under a member based on the age of the items: the older messages will be listed first.

```
member eh0721 Earnest Hemingway
edge
item 2 4  eh0721 "I have arrived in LAX" 3600
reference 1 3 eh0721
http://www.cs200classmatebook.org/~eh0721 3500000
```

For the posting comments and status change on the wall, the target object is specified as the URL of the page. You can use the java.lang.String.split() method to parse the line you have to read.

(6) `PA1`

The class that includes the main for this assignment.

C. Requirements

The test cases listed below are provided to assist you in verifying the correctness of your software. You are also required to test your software with different test cases that you will build yourself. Actual grading will be done by test cases using the same commands; however, the values that the specified input arguments take will be different.

(1) Test case 1

Objective: Print user's last name

Command: `java PA1 PA1-exampleData.txt print_fullname`

Output: `Earnest Hemingway`

(2) Test case 2

Objective: Print user's ID

Command: `java PA1 PA1-exampleData.txt print_userID`

Output: `eh0721`

(3) Test case 4

Objective: Pop an edge and print it out. Repeat this step n time and print out the size of the current edge stack.

Command: `java PA1 PA1-exampleData.txt pop_printEdges 2`

Output:

```
edge
item 111 4 eh0721 Message111 10
reference 1 3 eh0721
http://www.cs200classmatebook.org/~eh0721 30525
edge
item 90 4 eh0721 Comment900 200
reference 48 4 eh0721 Message48 3600
8
```

(4) Test case 4

Objective: Peek an edge that was the nth record (from top of the stack) in your edge stack, and print the edge. In this test case, you need to pop 2 edges and peek the stack. Then print the peeked item and the current size of the edge stack.

Command: `java PA1 PA1-exampleData.txt browse_printEdge 3`

Output:

```
edge
item 59 6 eh0721 3300
reference 48 4 eh0721 Message48 3600
8
```

(5) Test case 5

Objective: Scan EdgeStack and find item having the itemID. If you find it, print the item, otherwise print nothing.

Command: `java PA1 PA1-exampleData.txt find_printItem 48`

Output:

```
item 48 4 eh0721 Message48 3600
```

PA1 file, input file, and submission instructions will be posted on the class web site with this document. **DO NOT MODIFY the main() of PA1.java that has been provided to you.**

E. Grading

This assignment will account for 5% of your final grade. The grading itself will be done on a 50 point scale. The grading breakdown is as follows:

Test case 1:	5 pts
Test case 2:	5 pts
Test case 3:	3 x 5 pts
Test case 4:	3 x 5 pts
Test case 5:	2 x 5 pts

If your software did not implement stack operations, there will be a 20 point deduction. **Use of java.util.Stack is expressly disallowed**. You can use arrays and LinkedList. Ask the GTA before you use some other class in the java API or a third-part class.

Note: You are required to work alone on this assignment.

G. Late Policy

Please check the late policy available from the course web page.