CSCI 125 Project 2: Mail System

As we all know, spam (unsolicited junk email) is clogging up the arteries of the internet. In this project you are going to modify the mail-system project from chapter 3 to filter spam, as well as to store MailItems for more efficient retrieval. Here are the specifications for this project. Read them carefully:

1) Do Exercise 3.54 if you haven't already (add a subject to MailItem).

2) Next, we want to modify the MailServer to store and retrieve MailItems more efficiently. The current version of the MailServer stores everyone's MailItems in one potentially huge ArrayList. Modify the MailServer so that every user has their own ArrayList for just the MailItems sent to them. The MailServer should use a HashMap, where the keys to the HashMap are the names of the users, and each user is mapped to an ArrayList containing all the MailItems sent to that user. Once you have finished modifying the MailServer as described, verify that you can send and receive individual MailItems using the MailClient. Each MailItem should be removed from the MailServer when it is retrieved, so that the user never sees the same MailItem twice.

3) Any email whose subject starts with the exact word "SPAM" or whose message contains the word "viagra" (with any capitalization) is spam, and should be ignored by the *post* method in the MailServer so that the intended recipient doesn't see it when he or she retrieves MailItems. (Hint: you will find some very helpful methods in the javadoc for the *String* class).

4) Currently, the MailServer only allows a MailClient to retrieve one MailItem at a time using `getNextMailItem`. Add a second method to the MailServer class that can retrieve any number of MailItems at a time, up to the maximum number of MailItems stored for that user:

```
public ArrayList<MailItem> getNextMailItems(String user, int howMany)
```

   a) If there are fewer than *howMany* items present, they should all be returned in the ArrayList.
   b) Remember that each MailItem should be removed from the MailServer when it is retrieved, so that the user never sees the same MailItem twice.
   c) If there are no messages, return null just as `getNextMailItem` did.

5) Next, add a *getNextMailItems(int howMany)* method to the MailClient that can be used to retrieve a particular number of MailItems. It should invoke the new method on the MailServer like this:

```
server.getNextMailItems(user, howMany)
```

6) Add a method to the MailServer class called `printAllMessages` that iterates through the users, prints out each user name, and prints out all the emails associated with that user. Format this nicely, e.g.,:

```
Here are the MailItems for matt:
From: kim
To: matt
Subject: Stammtisch
Message: Meet me at 4:30 PM at the Curragh
----------------------------------------
Here are the MailItems for Suzie:
…
```

7) Write a test class named *MailTester* that thoroughly tests your modified mail-system:
   a) Test `howManyMailItems` in cases when a user
      1. has some mail,
      2. has 0 mail but is in the HashMap (because previous messages were downloaded, for instance),
      3. is not in the HashMap.
   b) Test `getNextMailItem` for both when there is mail and when there is not mail for a user.

c) Test `getNextMailItems(user, howMany)` when `howMany` is greater than, equal, or less than the number of messages. Also ensure it works when a user is not in the HashMap. Make sure to not only test that the proper number of messages were retrieved in each case, but that they are the correct ones—an easy way to do this is give them different subjects and test them against the expected subjects.
d) Also test `getNextMailItems(howMany)` in MailClient.
e) Test that `post` filters message with subject beginning with SPAM.
f) Test that `post` filters message with message containing Viagra.
g) Test `printAllMessages()` in MailServer.

When you have completed Project 2, select "Create Jar File ..." from the File menu in BlueJ. Check "include source" and "include Bluej project files". Save the file with the name bassett-project2.jar (except use your name instead of bassett). Then upload the "jar" file into Moodle.

Also, bring a stapled printout (double-sided if possible) of your MailServer, MailClient and MailTester classes to class if you would like me to mark it up with comments.