Homework 6: Mock objects

The following class `YahooQuery` sends a query string to `www.yahoo.com` and retrieves the results.

```
import java.net.*;
import java.util.*;
import java.io.*;

public class YahooQuery {

   protected int count;
   protected Vector<String> queryResult;

   public YahooQuery() {
      count = 0;
      queryResult = new Vector<String>();
   }

   protected void readFromURL(String urlStr) {
      queryResult = new Vector<String>();

      try {
         URL url = new URL(urlStr);
         BufferedReader in = new BufferedReader(
            new InputStreamReader(url.openStream()));
         String str;
         while ((str = in.readLine()) != null)
            queryResult.add(str);
         in.close();
      } catch (MalformedURLException e) {
      } catch (IOException e) {
      }
   }

   public void query(String query) {
      String encodedQuery = "";

      try {
         encodedQuery = URLEncoder.encode(query,"UTF-8");
      } catch(Exception e) {
         encodedQuery = "";
      }
      String urlStr = "http://search.yahoo.com/search?p=" + encodedQuery;
      readFromURL(urlStr);
   }

   public int resultCount() {
      // TODO: write this method
      return 0;
   }
}
```

An example of the use of this class is in the following code:

```
            YahooQuery yq = new YahooQuery();
            yq.query("j2ee");
            System.out.println(yq.resultCount());
```

As you might have noted, class YahooQuery has no unit test, and also it is incomplete. (E.g., method resultCount is only a stub.)

Your tasks

1. It is very difficult to write a unit test for this class, because there is a dependency in method `readFromURL`. Rewrite this method, so that it is possible to do unit testing without actually connecting to the real yahoo.com, by using mock objects.
(Hint: try to encapsulate all the input/output calls.)

2. Now, since your class is testable, write a Junit test for it. You should use EasyMock or jMock to create mock objects for your test. It should contains a few interesting testcases. Note that the class with the original implementation of `resultCount` should definitely fails this test.

3. Implement method `resultCount` so that the class passes all testcases. You can change the data structures (currently a Vector) for storing the html data.

Submission

You should send your `YahooQueryTest.java` and `YahooQuery.java` to me via e-mail. (Look for my e-mail at http://www.cpe.ku.ac.th/~jtf .) The subject should be "`219343 Homework 6`."

Due date

Thursday, December 20th, 2007