**Project 2: Editable TextLine**

**Due:    July 6, 2010**

In this project you are asked to create a class TextLine representing a line of text.  Internally, the line of text *must* be represented as an array of characters.  You may not use (directly or indirectly) the String class to represent the content of TextLine.  Also, you are asked to implement a derived class EditableTextLine by implementing the Editable interface. You must test your project by providing a Tester class, as described later.

In this project, you are also required to provide suitable JavaDoc comments for your classes, methods, and constants.

While logged in to odin, submit your project directory, called Project2, to cs1302a, using the submit command.  The submitted directory should contain:

* All the source code, i.e. .java files. Please do NOT include .class files.  Note that you are required to provide javadoc comments for your classes, constants, constructors, and methods.  Other comments explaining the logic of your code are also required!  Also, your source code should be well formatted, as usual.
* A README file telling us how to compile your program and how to execute it.

**Project description:**

* Except for the Tester class (described later), your program must be placed in package edu.uga.cs1302.txtbuff.
* Define a public class TextLineIndexOutOfBoundsException which must be derived from java.lang.IndexOutOfBoundsException and must have the following public constructors:

TextLineIndexOutOfBoundsException()

constructs a TextLineIndexOutOfBoundsException object with no detailed message. It simply calls the default constructor of its superclass.

TextLineIndexOutOfBoundsException(String errMsg)

constructs a TextLineIndexOutOfBoundsException object with the specified detailed message. It should call the one parameter constructor of the superclass passing errMsg as the argument.

TextLineIndexOutOfBoundsException(int index)

constructs a new TextLineIndexOutOfBoundsException object with an argument indicating the illegal index. It should call the one parameter constructor of the superclass passing the following string as the argument: "TextLine index out of range: " + index

* Define a public class TextLine including the following:

    public int constant:

DEFAULT\_SIZE

specifies the default capacity of a TextLine. It should be set to 80.

   public constructors:

TextLine()

constructs an empty TextLine. The line length should be set to 0, but the capacity of the internal array should be set to DEFAULT\_SIZE.

TextLine(String line)

constructs a TextLine object initialized to contain the same characters as in the argument string line. The line length should be set to the length of the arguments string, but the capacity of the internal array should be set to the smallest  multiple of DEFAULT\_SIZE, sufficient to store the line.

   public methods:

int       indexOf(String fragment)

returns the index position of the first occurrence of the fragment in this TextLine, or -1, if the fragment is not found.

int       indexOf(String fragment, int fromIndex)

returns the index position of the first occurrence of the fragment string in the line, starting at the specified index fromIndex, or -1, if the fragment is not found.

int       length()

returns the length of this TextLine.

int       capacity()

returns the current capacity of this TextLine.

String    toString()

returns the contents of this TextLine represented as a Java String.

**BONUS method** (also public):

int       indexOfMatch(String regex, int fromIndex)

returns the index position of the first occurrence of characters in this TextLine, starting at the specified index fromIndex, which match the given regular expression regex.  If a match is not found, -1 should be returned.

* Define a public interface Editable containing the following methods:

void     append(String fragment)

appends the given string fragment at the end of a TextLine.

void      insert(int index, String fragment)

              throws TextLineIndexOutOfBoundsException

inserts the given string fragment at the specified index of a TextLine. A TextLineIndexOutOfBoundsException exception is thrown if the supplied index is out of bounds of a TextLine.

void      replace(int start, int end, String fragment)

              throws TextLineIndexOutOfBoundsException

replaces the characters between start and end in this TextLine with characters in the specified string fragment. A TextLineIndexOutOfBoundsException exception is thrown if the supplied start or end are out of bounds of a TextLine.

* Define a public class EditableTextLine, which extends TextLine and implements the Editable interface.  You must provide the following constructors:

EditableTextLine()

which constructs an empty EditableTextLine. You should rely on the parent class constructor.

EditableTextLine(String line)

which constructs a EditableTextLine object initialized to contain the same characters as in the argument string line. You should rely on the parent class constructor.

Note, that in implementing the methods of the Editable interface, you will have to increase the internal capacity of the EditableTextLine, if the length of the resulting line (after appending, inserting, or replacing) exceeds the current capacity.  I suggest doubling the capacity whenever this occurs.

* Create a public class Tester in the default package, which should:
  + prompt the user for a line of text;
  + read the entered line of text and create an EditableTextLine object called eLine and initialized to the entered characters;
  + echo it back (print) using *toString*();
  + print its: *length()* and *capacity();*
  + prompt the user for a string of characters and read it and save it in variable str;
  + print index positions of *all* occurrences of str in eLine using *indexOf*;
  + append str in eLine and print eLine using *toString*();
  + insert str in eLine at index position 0 and print eLine;
  + insert str in eLine at index position str.length() and print eLine;
  + replace all occurrences of str in eLine with the string ”abc” and print eLine.

All reading should be from System.in and writing to System.out.

* Run javadoc to create the API documentation for your project. Place the resulting files in directory html, a subdirectory in your project directory Project2.

**Things to note:**

* Your classes, except for the Tester, must be in package edu.uga.cs1302.txtbuff.
* As usual, follow a good coding style.
* You must provide suitable JavaDoc comments for your classes, methods, and constants.
* Your design should be reasonably efficient, and in accordance with object oriented design principles (encapsulation, information hiding, inheritance, etc.). Should the array representation and other fields in class TextLine be private or protected?
* If you are defining a field as public or protected, briefly mention the reason for that.
* If you are defining a field as static or final, briefly mention the reason for that, as well.
* If you need private or protected methods, briefly mention the reason for that