Chris Piech
CS 106A

Handout #24
February 6, 2017

Section Handout #4: String Processing

Portions of this handout by Eric Roberts, Patrick Young, Jeremy Keeshin and Nick Troccoli

Codingbat practice using Strings

To give you more practice working with Strings, a set of new problems on Strings have been made available on the Codingbat website:

```
http://codingbat.com/exp/CS106A-Codingbat
```

We encourage you to work through some of those problems to get more hands-on experience with Strings.

1. Adding commas to numeric strings (Chapter 8, Exercise 13, page 290)

When large numbers are written out on paper, it is traditional—at least in the United States—to use commas to separate the digits into groups of three. For example, the number one million is usually written in the following form:

```
1,000,000
```

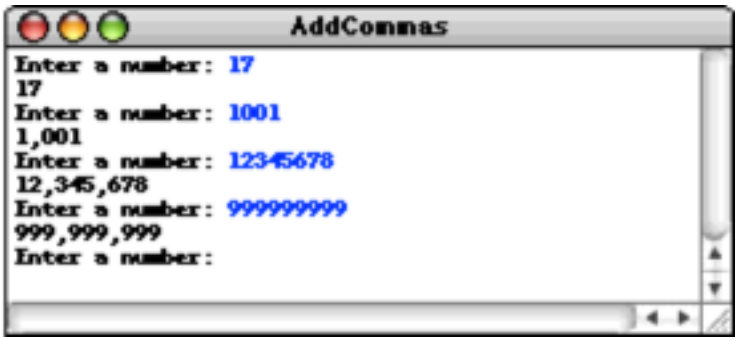
To make it easier for programmers to display numbers in this fashion, implement a method

```
private String addCommasToNumericString(String digits)
```

that takes a string of decimal digits representing a number and returns the string formed by inserting commas at every third position, starting on the right. For example, if you were to execute the main program

```
public void run() {
   while (true) {
      String digits = readLine("Enter a numeric string: ");
      if (digits.length() == 0) break;
      println(addCommasToNumericString(digits));
   }
}
```

your implementation of the `addCommasToNumericString` method should be able to produce the following sample run:



2. Deleting characters from a string

Write a method

```
public String removeAllOccurrences(String str, char ch)
```

that removes all occurrences of the character **`ch`** from the string **`str`**. For example, your method should return the values shown:

```
removeAllOccurrences("This is a test", 't')  returns  "This is a es"
removeAllOccurrences("Summer is here!", 'e') returns  "Summr is hr"
removeAllOccurrences("---0---", '-')         returns  "0"
```

3. Converting a string to alternating capital letters

Write a method

```
public String altCaps(String str)
```

which converts a string to alternating capital letters, meaing you alternate between uppercase and lowercase. This style of typing was prevalent on the internet in the late 90s. For example:

```
altCaps("hello")               returns  "hElLo"
altCaps("section is awesome") returns  "sEcTiOn Is AwEsOmE"
```

Note that characters that are not letters are not changed and do not affect the alternating sequence of uppercase and lowercase letters.

4. Tracing method execution

For the program below, show what the graphics canvas looks like when it runs.

```
/*
 * File: Collage.java
 * ------------------
 * This program adds a collage of non-overlapping images to the screen.
 */

import acm.program.*;
import acm.graphics.*;

public class Collage extends GraphicsProgram {

   private GImage tree = null;

   public void run() {
      GImage star = new GImage("star.png");
      GImage smiley = new GImage("smiley.png");
      tree = new GImage("stanfordTree.png");
      add(tree, 0, 0);
      tryToAdd(star);
      tryToAdd(smiley);
      GImage otherImage = star;
      tryToAdd(otherImage);
   }

   private void tryToAdd(GImage image) {
      if (addImage(image, 0, 0)) {
         return;
      } else if (addImage(image, 200, 20)) {
         return;
      } else {
         addImage(image, 200, 300);
      }
   }

   private boolean addImage(GImage image, double x, double y) {
      GObject obj = getElementAt(x, y);
      if (obj == image) {
         return true;
      } else if (obj != null) {
         return false;
      } else {
         add(image, x, y);
         return true;
      }
   }
}
```