

Section Handout #4 - Strings

Based on handouts by Keith Schwarz and Marty Stepp

Strings

1. Adding commas to numeric strings. Write a method named `addCommas` that takes a `String` representing a number and returns a `String` with a comma at every third position, starting from the right. (e.g. one million is written as 1,000,000). For example, your method should return the values shown below:

```
addCommas("147")           returns "147"
addCommas("2014")          returns "2,014"
addCommas("37897987")      returns "37,897,987"
```

2. Removing Characters. Write a method named `removeAll` that takes a `String` and a character as parameters, and removes all occurrences of the character (without using the `String` `replace` method.) For example:

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

3. AIT cApS. Write a method named `convertToAltCaps` that takes a `String` as a parameter and returns a version of the `String` in alt caps (where alternating letters are uppercase and lowercase). For example, your method should return the values shown below:

```
convertToAltCaps("hello")           returns "hElLo"
convertToAltCaps("section is GREAT") returns "sEcTiOn Is GrEaT"
```

Note: For each of the problems above, we ask you to write a method that takes input as parameters and returns some result to the caller. For extra practice, try writing code to call each of these methods (and pass in the appropriate parameters) within the `run()` method!

File Reading

4. Boy/Girl. Write a method called `boyGirl` that accepts as a parameter a `BufferedReader` that is reading a file containing a series of names followed by integers. The names alternate between boys' names and girls' names. Your method should print out the number of boys' and girls' names, and the absolute difference between the sum of the boys' integers and the sum of the girls' integers. The input could end with either a boy or girl; you may not assume that it contains an even number of names. However, you can assume that each line consists of a name followed by a single space and then the age. For example, if the input file `counts.txt` contains the following text:

```
Jared 3
Kim 7
Paul 14
Alisa 13
Cody 4
Jenny 12
Ben 6
Coral 7
Trent 6
```

Then your method could be called in the following way:

```
try {
    BufferedReader rd = new BufferedReader(new FileReader("counts.txt"));
    boyGirl(rd);
    rd.close();
} catch (IOException ex) {
    println("Can't open that file.");
}
```

In this file, the boys' sum is 33 and the girls' sum is 39. Thus, your method should produce the following output:

```
5 boys, 4 girls
Difference between boys' and girls' sums: 6
```