coe318 Lab 3:
Linked Counters

Objectives

- Implement a `Counter` class.
- Learn how objects can be linked together.
- Use an "if" statement.
- **Duration: one week.**

Discussion

In mathematics, a number is expressed in positional notation to a certain base, B, a

$$d_n d_{n-1} \ldots d_1 d_{0B} = \sum d_i B^i$$

For example, the 3-digit number 123 in base 4 represents 16+8+3=27 (base 10).

In this lab each digit is represented as a Counter object. A Counter object has an optional left neighbour which is also a Counter object. (The absence of a left neighbour is indicated with the keyword `null`.

The important methods to implement are **`getCount()`** and **`increment()`**.

If there is no left neighbour, the count is the same as the digit.

If there is a left neighbour, the count is the sum of the digit and the modulus times the count of the left neighbour.

The **`increment()`** method increment's the Counter's digit and, if it reaches its maximum (modulus) value, it is reset to zero. Furthermore, if there is a left neighbour and if the Counter has rolled over, its left neighbour should be incremented as well.

The source code template for **`Counter`** can be accessed here

There is also a class containing the main method, **`CounterTry`**, which can be accessed here

Step 1: Create a Netbeans project

1. Create a Netbeans project called **`Counter`** which should be placed in a folder called **`lab3`** (all lowercase and no spaces). The **`lab3`** folder should itself be in your **`coe318`** folder.
2. Create a Java file (class library type) called **`Counter;`** set the package to `coe318.lab3`; then copy and paste the provided source code.
3. Similarly, create the Java file **`CounterTry`**. (Ensure you use the same `coe318.lab3` package name.
4. Generate the javadocs and compile and run the project.
5. It should compile correctly and produce output. Unfortunately, the output is incorrect and you have to fix it.

Step 2: Add instance variables and fix constructor and getters

1. Add instance variables for the two components of a counter.
2. Modify the constructor so that they are properly initialized.
3. Fix the remaining methods so that they work for a simple counter without a left neighbour.
4. Compile and run your project. As least some of the output should now be correct.

Step 3: Fix methods so it works when there is a left neighbour

1. Fix the remaining methods.

Step 4: Submit your lab

1. Submit your lab by zipping it to a file called **`lab3.zip`**
2. Then use the command **`submit coe318 lab3 lab3.zip`** to complete the submission.