HW 1 - Functional Programming in Java

Goals:

- Learn basic Java syntax for control and logic structures
- Create a new Java command line project in InteliJ
- Build and run a simple Java command line application
- Version the project in GitHub

End of Chapter Exercises

Before beginning, you'll want to read through Chapters 1 through 3 in your textbook. These chapters introduce basic Java language syntax for how to use program control and logic structures. We will use these throughout the semester, but will not be spending a lot of time on them, as it is assumed you have a foundation in at least one other programming language and have worked with basic program control and logic structures.

You should be familiar with how to use conditional statements and boolean logic, `if-else` statements and the logical operators `and`, `or`, `not`, as well and control such as a `for` or `while` loops. You should also understand how to work with strings and arrays (lists of items). These are the foundation of most programming languages and translate pretty easily between them. Once you learn one programming language, it's generally pretty easy to apply what you know to learning the next.

As you complete each chapter, work through the Self Test Exercises at the end of each Chapter to measure what you have learned. The answers are in Appendix A at the back of the book, so you can evaluate and correct your work when you are done. If you have questions on anything covered, please ask. The class chat room will be an excellent spot to discuss these.

Part I - Counting Game

When teaching kids division, they often will play a word game where the students will sit in a circle and count up from 1 to some number (say 10 for example), but with a twist. If a number is divisible by another number, that number is replaced with a word.

For example, a rule might be if a number is divisible by two, then say "woof". So it would go like this:

1
woof
3
woof
5
woof
7
woof
9
woof

The game grows in complexity as more rules are added. If we add a new rule to say “meow” when a number is divisible by three, we get:

1
woof
meow
woof
5
woof meow
7
woof
meow
woof

Notice that in place of 6 we said both woof and meow because 6 is divisible by both 2 (woof) and 3 (meow).

Assignment

Your first assignment is to write code for this number game in Java. Recall that you can tell if a number is divisible by another by using the modulus operator (%) which gives you the remainder from division. If the result is zero (there is no remainder) then the first number is divisible by the second.

For your number game, we will count from 1 to 30 using the following rules:

1. If a number is divisible by 5, then print "beep" instead of the number
2. If a number is divisible by 8, then print "buzz" instead of the number
3. If a number is divisible by 12, then print "zap" instead of the number

Remember: if the number is divisible by both numbers, both words should print on the same line.

GitHub Info

Repository: https://github.com/htc-ccis2595/counting-game

You need to fork this repository and submit a pull request to turn in the assignment. Please post a screenshot of the pull request to the D2L dropbox. The dropbox is mainly used to communicate due dates, not for storing the completed work. You do not need to upload the project to D2L, only to GitHub.

Part II - 99 Bottles

The second part of the assignment is to write a program that will print out the lyrics to 99 Bottles. Bottles of what? Make a variable for what is in the bottle (so it can be easily changed) and start it out as “beer” or another beverage of your choice, if you’d rather.

You will also want to have a variable for the initial number of bottles, as it gets tedious to scroll through 99 verses each time. I recommend working with this set to 3 for starter, then changing it to 99 when you have everything working. Remember to update this, as I am expecting 99 Bottles, not 3 Bottles...

Assignment

Your job is to write a command line program in Java that prints the song lyrics, nicely formatted as follows:

> 99 bottles of beer on the wall.
> 99 bottles of beer!
> Take one down. Pass it around.
> 98 bottles of beer on the wall.
>
> 98 bottles of beer on the wall.
> 98 bottles of beer!
> Take one down. Pass it around.
> 97 bottles of beer on the wall.
>
> ... and so on until ...
>
> 2 bottles of beer on the wall.
> 2 bottles of beer!
> Take one down. Pass it around.
> 1 bottle of beer on the wall.
>
> 1 bottle of beer on the wall.
> 1 bottle of beer!
> Take one down. Pass it around.
> No more bottles of beer on the wall.

Be very careful with the singular / plural of the word bottle(s) as it appears in the lyrics. There are two things I am looking for in this assignment:

1. Can you use a loop to control the number of times code is run
2. Can you use that loop counter to alter the program behavior

You want to keep an eye on the number of bottle to get the correct singular or plural format of the word bottle, and you also need to alter that last line of the verse when we run out of bottles.

While this may seem a trivial or silly assignment at first, using a loop counter and altering program behavior based on the value of that counter is a very common requirement in many programs. So pay attention to the details here and write the most elegant code that you can to solve this seemingly silly assignment.

GitHub Info

Repository: https://github.com/htc-ccis2595/99-bottles-java

You need to fork this repository and submit a pull request to turn in the assignment. Again, please post a screenshot of the pull request to the D2L dropbox. There will be only one D2L dropbox for both parts of the assignment, so both screenshots will go to the same place.