CS 251 Intermediate Programming using Java
Lab 1: FizzBuzz and Command Line Arguments

Brooke Chenoweth

Spring 2024

Problem Specification

The “FizzBuzz Problem” is a well-known interview question.[1] It usually is given as something like the following:

> Write a program that prints the numbers from 1 to 100. But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”. For numbers which are multiples of both three and five print “FizzBuzz”.

In this assignment, you’ll be implementing a variation of this problem to make sure you are comfortable writing a simple Java program from scratch and to introduce you to using command line arguments.

Command Line Arguments

The `String` array parameter in the `main` method contains the command line arguments given to the program by the user. When running the program, the arguments are given after the class name and are separated by whitespace. For example, if I ran a program named “MyProg” using the command `java MyProg Hello Everybody! 2 4 6`, the `args` array would contain five values.

Array Index	0	1	2	3	4
Value	`"Hello"`	`"Everybody!"`	`"2"`	`"4"`	`"6"`

Command line arguments are not the same thing as reading input from standard input! You may recall using a `Scanner` in CS152 to read input typed into the console as your program runs.[2] Command line arguments are not the same thing. The values of the command line arguments are given to the program when the program starts, not read in interactively later.

[1] This is the sort of question asked if there are concerns that the candidate is unable to code their way out of a paper bag, so you’ll likely encounter something more complicated in a real interview.

[2] We will be using the `Scanner` class again on future assignments.

What you have to do

Create a class named `FizzBuzz` that contains a `main` method that expects three command line arguments. The first argument will be the upper limit for the numbers you will print, the second argument will be the number whose multiples will be replaced with "Fizz", and the third argument will be the number whose multiples will be replaced by "Buzz". Numbers that are multiples of both the second and third argument will be replaced by "FizzBuzz".[3]

The program should print the numbers from 1 to the limit (including the limit value), each on a separate line, replacing the multiples of the first and second arguments with "Fizz", "Buzz", or "FizzBuzz" as appropriate.

If the program is run with the command `java FizzBuzz 100 3 5`, it will be the classic FizzBuzz problem described earlier.

Hints and Assumptions

- You may assume the user will provide you with valid arguments (3 positive integer values). If you would like to make your program more robust, feel free to provide additional error checking, of course.
- Java's `Integer` class contains a `parseInt` method that takes a `String` representation of an integer value and returns that value as an `int`. You'll probably want to use this.
- Remember that a mod b equals zero if a is a multiple of b.
- The program should get its input from command line arguments, not from interacting with the user as it runs. I know many of you are comfortable using the `Scanner` class, but that is not what I want you to use for this assignment.

Compiling and Running

1. Create `FizzBuzz.java` in your favorite text editor.
2. Compile using `javac`.[4]
3. Run using `java`. Make sure you give 3 positive integer values as command line arguments.[5]
4. Test your program with a variety of inputs to make sure it works. You will not receive full credit if you do not properly parse and use the command line arguments!

[3] "FizzBuzz" should be printed on a single line in the output, not as two separate "Fizz" and "Buzz" lines.

[4] Even if you are using an IDE to edit your code, please compile using `javac` and run using `java` on the command line to be familiar with using the terminal.

[5] Really, it's a lot easier to test different command line arguments by running in a terminal instead of having to navigate several levels in the IDE settings to specify the command line arguments.

I provided you with a transcript of my program testing a variety of possible command line argument values. I strongly recommend that you look at this and make sure your output is the same.

Turning in your assignment

Once you are done with your assignment, use Canvas to turn in the `FizzBuzz.java` file that you have created. Note that each source code file should contain a header comment with your name, class, and other related information. Failure to properly comment your files and follow the class coding standards may lead to point deductions.