Challenge - FizzBuzz

1 Hour

The Assignment

In this challenge you will write a Java program, called *FizzBuzz*. You may use your TRAINS for Reference. You may not use any other reference materials (NO GOOGLE, NO STACKOVERFLOW etc.). You have one hour. (Time yourself!)

FizzBuzz

Write the class `FizzBuzz` in the file `FizzBuzz.java`. The main method of the `FizzBuzz` class must take input from the user with the prompt "Enter the limit:" The program must then create a `String` array containing all numbers from 1 up to and including the limit. The catch is that the program must:

- put "Fizz" in the place of any multiple of 3
- put "Buzz" in the place of any multiple of 5
- put "FizzBuzz" in the place of any number that is a multiple of both 3 and 5

It must then *print* the entire array to the console. Print an entire Array by importing `java.util.Arrays` and using `System.out.println(Arrays.toString(<ArrayName>))` where `<ArrayName>` is the name of your array. Hint: read all the Tips!

Expected Output

Example 1

After you run your program, and enter `5` when prompted, your console should display

```
Enter the limit:
5
[1, 2, Fizz, 4, Buzz]
```

Example 2

After you run your program again, and enter `5` when prompted, your console should display

```
Enter the limit:
15
[1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz, 11, Fizz, 13, 14, FizzBuzz]
```

Submitting to the Autograder

1. Complete your assignment, making sure your program's output matches the expected output stipulated by the assignment brief.
2. Make sure that your program compiles and runs without any errors.
3. Create a `zip` file containing the `FizzBuzz.java` file.
4. Upload the `zip` file to Athena.

Tips

1. You must use `scanner` for user input and you can read numbers using `sc.nextInt()`
2. `Integer.toString(i)` returns the `String` version of the number `i`

3. `a%b` returns the remainder of dividing `a` by `b`; so `10%5` returns `0`
4. Print an entire Array by importing `java.util.Arrays` and use `System.out.println(Arrays.toString(<ArrayName>))` where `<ArrayName>` is the name of your array.
5. Every printed String must end with a newline character.
6. The `System.out.println` method automatically appends a newline character to the string it prints.
7. Remember that your class names needs to be exactly the same as the name of the files in which they were defined.
8. Make sure that all of your input and output exactly matches the strings shown in the examples.
9. Do not include any `package` declarations in your submission file.
10. DO NOT CLOSE YOUR SCANNER