Java Loops and Collections

CSC 210 Practice Exercises

FizzBuzz Array

Write a method that takes in an integer `k` as argument and returns an array with strings as values, for every integer from zero to `k`. The string should be `"FizzBuzz"` if `n` is divisible by 3 and 5, `"Fizz"` if `n` is divisible by 3, and `"Buzz"` if `n` is divisible by 5 or `n` (as a string) if none of the conditions are true.

Prime

Write a method that returns `true` if an integer `n` is prime, `false` otherwise.

You can determine if a number n is NOT prime if it is divisible by any number between 2 and n / 2.

The integer 1 is not a prime number.

Fibonacci Sequence

Fibonacci numbers are a sequence of numbers where every number is the sum of the preceding two numbers.

Write a method that takes in an integer `k` as argument and returns an array with integer as values, representing a Fibonacci sequence of size `k`. Assume `k >= 2`.

Prime numbers

Write a method that returns an array with the first `n` prime numbers.

ANSWERS

FizzBuzz Array

```
public static String fizzBuzzSingle(int n) {
        String result = "";

        if (n % 3 == 0) result += "Fizz";
        if (n % 5 == 0) result += "Buzz";

        if (result.equals("")) result += n;

        return result;
    }

public static ArrayList<String> fizzBuzz(int n) {
        ArrayList<String> result = new ArrayList<String>();

        for (int i = 0; i <= n; i++) {
            result.add(fizzBuzzSingle(i));
        }

        return result;

    }
```

Prime

```
public static boolean isPrime(int n) {

        if (n == 1) return false;

        for (int i = 2; i <= n/2; i++) {
            if (n % i == 0) return false;
        }

        return true;
    }
```

Fibonacci Sequence

```
public static ArrayList<Integer> fibonacci(int k) {
        ArrayList<Integer> result = new ArrayList<Integer>();

        result.add(0);
        result.add(1);

        for (int i = 2; i < k; i++) {
            result.add(result.get(i-1) + result.get(i-2));
        }

        return result;
    }
```

Prime numbers

```
public static boolean isPrime(int n) {

        if (n == 1) return false;

        for (int i = 2; i <= n/2; i++) {
            if (n % i == 0) return false;
        }

        return true;
    }

public static ArrayList<Integer> firstPrimes(int n) {
        ArrayList<Integer> result = new ArrayList<Integer>();

        int current = 1;
        while (result.size() < n) {
            if (isPrime(current)) result.add(current);
            current += 1;
        }

        return result;
    }
```