Chapter 3 Labs

Based in part on lab exercises from https://github.com/orhs-apcs/chapter-3

Getting Started

See instructions here.

Labs

1. Rock, Paper, Scissors
2. FizzBuzz
3. Guessing Game
4. Fibonacci
5. Factors
6. Odd and Even Digits

Rock, Paper, Scissors

Create a program that allows the user to play against the computer in rock, paper, scissors. Here's the general outline of the program:

1. The user is prompted to enter their play as a single character (R, P, S, r, p, or s) in the console.
2. The computer's play is chosen at random (uniformly) and printed.
3. The winner is determined and printed out.

There is some code already in `RockPaperScissors.java`; use this as a starting point and keep the console messages consistent.

FizzBuzz

Create a program `FizzBuzz.java` that prints the numbers from 1 to 100 with the following exceptions:

1. If the number is divisible by 3, print `Fizz` instead
2. If the number is divisible by 5, print `Buzz` instead
3. If the number is divisible by both 3 and 5, print `FizzBuzz` instead

Guessing Game

Create a program `GuessingGame.java` that plays a number guessing game with the user. At the start, the computer picks a number between 1 and 100. The user then tries to guess the number in the least number of tries. Here's a sample output of the program.

```
I'm thinking of a number between 1 and 100.
Enter your guess: 27
Too high; guess again: 12
Too low; guess again: 19
Too low; guess again: 21
Too high; guess again: 20
You got it after only 5 guesses!
```

Bonus: What is the optimal strategy? What is the maximum number of guesses needed to guess any given number?

Fibonacci

Create a program `Fibonacci.java` that prints the first 20 numbers in the Fibonacci sequence. To refresh, the Fibonacci sequence begins with 0, 1 and each following number in the sequence is the sum of the previous two terms.

Factors

Create a program `Factors.java` that finds the factors of a number entered by the user. Here's a sample output:

```
Enter a number: 12
1
2
3
4
6
12
```

Now, modify it to only print out the prime factors. In this case, the new output would be:

```
Enter a number: 12
2
3
```

Odd and Even Digits

Create a program `EvenOdd.java` to count the even and odd digits in a number. Here's the general procedure:

1. Prompt the user to enter an integer
2. While there are more digits
    1. Get the next digit
    2. If it's 0, increment the zero count
    3. If it's even, increment the even count
    4. If it's odd, increment the odd count
3. Print out the counts

Here's some sample output:

```
Enter an integer: 23039
There is 1 zero digit.
There is 1 even digit.
There are 3 odd digits.
```

Note how the verb and plurality of 'digit' changes based on the count.

Hint: use `%` and `while` in a clever way