CS 235 - Week 2 Lab Exercise - 2021-09-03

Deadline

Due by the end of lab on 2021-09-03.

How to submit

- Submit your files `DieSetPlay.java` and `ArgsPlay.java` on https://canvas.humboldt.edu

Purpose

To practice with some of the Java features discussed in class this week and last week.

Important notes

- You are required to work in **pairs** on this lab exercise. If you are not pair-programming, then you may not receive full credit for your lab exercise.
  - If there are an odd number of students attending lab, one team may have 3 students.
  - **Be sure to TYPE BOTH OF YOUR NAMES** in the beginning comment of EACH of your `.java` files

Lab Exercise set-up

- FIRST: in the CS50 IDE, in your folder for today's lab exercise, create a local copy of:
  - `GameDie.java` from the Week 1 Lab

Problem 1

- Start a new Java application class `DieSetPlay.java`,
  - Include both of your names in an `@author` line in a `javadoc`-style opening comment for this class
  - Also include today's date in a `@version` line in this comment
- Class `DieSetPlay`'s `main` method should also start with a `javadoc`-style comment (see the comment used for class `GameDieTest` in the posted Week 1 Lab version of `GameDieTest.java`)
- Class `DieSetPlay`'s `main` method should do at least the following (although you are welcome to add MORE if you would like!):
  - Ask the user for a desired number of sides they want for a set of game die, and read in their entry.
  - Ask the user for a desired *quantity* of game die they want in that set, and read in their entry.
  - Create an array of that many `GameDie` instances.
  - Loop through that array, creating a new `GameDie` with the specified number of sides for each element in that array.
  - Ask the user if they want to roll their set of game die, and read in their entry.
  - If they answer yes, roll each game die in their set of game die, printing the roll results to the screen in a tasteful fashion.
    - (you get to choose what counts as a yes answer)

Problem 2

FUN FACT: remember how a Java application's `main` method has `String` array argument?

If you call a Java application as follows:

```
java MyApp moo 3 oh my
```

...then the `moo 3 oh my` are considered to be four **command line arguments** for this call of this Java application `MyApp`,

...and, in this case, its `main` method's `String` array argument will have 4 string values in it, in this case `"moo"`, `"3"`, `"oh"`, and `"my"`.

- Start a new Java application class `ArgsPlay.java`,
  - Include both of your names in an `@author` line in a `javadoc`-style opening comment for this class
  - Also include today's date in a `@version` line in this comment
- Class `ArgsPlay`'s `main` method should also start with a `javadoc`-style comment (see the comment used for class `GameDieTest` in the posted Week 1 Lab version of `GameDieTest.java`) --

  ...BUT in this case you should not say that `@param args` is NOT used here, because you are GOING to use it!
- Its `main` method should do at LEAST the following (although you are welcome to add MORE if you would like!):
  - Print to the screen a message saying HOW many command line arguments were given. (Hint: how can you find out how many elements are in a Java array?)
  - Using either `String` method `toLowerCase` or `toUpperCase` -- your choice -- print to the screen each of this call's command line arguments in either all-uppercase or all-lowercase, one command-line argument per line.

- When you are done, or before you leave lab, use Gmail to
  - MAIL a copy of your `.java` files to BOTH/ALL of you, and
  - EACH of you should SUBMIT the required files on Canvas