Lab 9: That's a loaded question

Objectives

- To learn about how to achieve polymorphism in Java

Step 0: setup

1. Create a project in IntelliJ
2. Add SimpleGame.java to your src folder. This file has a partially-complete implementation of a simple Dice game using 4 dice. In the game, we roll all 4 dice, and if they total a large enough value, we win. Otherwise, we roll again.
3. Add StandardDice.java to your src folder. This file implements a standard dice, similar to what we have done in another lab.

Step 1: Making different Dice

The SimpleGame will not work yet because your program is missing two types: Dice and LoadedDice.

Dice

Dice should be an interface that specifies the methods that all Dice-implementing classes should provide. Write Dice so that it requires the following methods:

```
public void roll();
public int getValue();
```

Be sure to add appropriate javadocs for those methods.

LoadedDice

LoadedDice should implement the Dice interface, but represent a **6-sided** dice that is weighted to be more likely to roll one value more often than the others. Write LoadedDice in LoadedDice.java.

It should have the methods the Dice interface requires, and it should also have two constructors:

1. `LoadedDice()` should create a dice weighted to roll a 4 more often than anything else.
2. `LoadedDice(int meanValue)` should take as a parameter the mean value the dice should roll (i.e. the value it should roll most often), which should be between 1 and 6, inclusive.

   Here's a simple implementation of a random number generator that will give appropriate values for such a loaded dice (meanValue is the value the dice is very likely to roll, and

deviation is the standard deviation to use; 0.3 is a good value for the deviation):

```
value = (int) Math.round(new Random().nextGaussian()
      * deviation + meanValue);
if (value < 1) {
    value = 1;
}
if (value > MAX_VALUE) {
    value = MAX_VALUE;
}
```

Make sure that LoadedDice and StandardDice all implement the Dice interface.

Step 2: Finish the game

Now that you have the three types Dice (an interface), StandardDice (a class), and LoadedDice (a class), you can finish the SimpleGame class by writing the helper methods I've left out:

1. `diceTotal`
2. `rollAll`
3. `showDice`

After you have finished these helpers, you should be able to run the game, which should produce results like the following:

```
[6]  [5]  [4]  [5]    ---> 20
Press enter to continue

[5]  [3]  [5]  [5]    ---> 18
Press enter to continue

[2]  [12]  [5]  [5]    ---> 24
You WON!!!!
```

How to turn in this lab

Before turning in any program in this class, remember this mantra:

Just because it works doesn't mean it's good.

Part of your grade will also come from things like how understandable and readable your code is. You will also be graded on the neatness, presentation, and style of your program code.

Turn in your java files on gradescope.

Ask for help if you're having problems!