Code review Lab3

First thing why I set properties of the class Dice private?

I could set them up public, but the problem here, everyone can access to them from everywhere outside the class et that is VERY BAD!!!, because what will happen if someone decides to modify these properties directly?

Example

```
class Dice {
    // properties
    public int sideUp;
        --------
    // can control that if sideUp is declared public
```

```
class DiceGame{
    public static void main(){
        Dice dice = new Dice();
        Dice.sideUp = -1; // no one
```

Then to stop things like that, we must declare properties of the class private (I wonder why it is not made by default, I searched that without finding any response)

The developer of the class is the one who makes roles inside the class and the users of this class must follow these roles.

We can imagine a class like a black box, the users of this class are not supposed to know exactly how things are made inside, the maker of the class gives us only setters to modify properties inside the class but following his roles and getters to get information from this class.

Before we can use a class, we must create an object from this class, this is why we use constructors.

According to w3schools.com, a constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created.

A class is like a blueprint for creating objects.

In this lab we have three constructors.

The first one is without parameters, it creates a dice with six sides (The common one) by default, and its side up is randomly set up. Dice dice = new Dice();

The second constructor is with one parameter, the number of sides, we need this one to create a dice object if the number of sides is not equal to six. For example, Dice dice20 = new Dice(20); will create a dice with twenty sides.

I added another condition inside this constructor (if numSides < 3 || numSides > 100 return;)

because in real life we can't find a dice with number of sides less than 3 nor greater than 100.

The third constructor is with two parameters, number of sides and dice type which nothing else but a name given to the dice.

I think we do not really need this constructor, because we can set a name to the dice by only knowing the number of sides like this.diceType = “{” + numSides + “}”; except if we must set a particular name to the dice.

Each type of dice created (constructor called) set the side up randomly, then I thought creating a private method that can return a number randomly between 1 and the number of sides and all constructors will call this method to ovoid redundancy.

This method must be private because we will never call it outside the class.

The last method created is roll() that must be public because once an object of a dice is created we can call it from outside the class that simulates a dice roll up.

We have a concrete example of that when we simulate a Yahtzee case.

We create five dices with six faces, here we need only the constructor without any parameter.

The goal here is to count the number of rolls until we get all faces having the same number.

Of course, the first thing that we think of is to use a loop, but which one?

In Java we have three types of loops, the for loop, the while loop and the do while loop.

For loop is automatically eliminated because with this loop we have to know the number of loops that we will make, or here we don’t know in advance how many loops it will take to have five dices having the same side up, this number will certainly change at each essay.

I have chosen do while loop because we must roll dices up at least once to know if we have a Yahtzee or not.