Java II: Making Dice

Let's write a bean style class that we will name **`TheDie`**.

We want to simulate a game die. A die is an object that can return the values 1, 2, 3, 4, 5 or 6 and are usually used in pairs where we refer to them as dice.

Your bean will have a single private instance variable called **`oneDie`** of type integer.

The bean will have a default constructor.

The bean will have a getter and setter for the instance variable **`oneDie`**.

Question: Should there be a setter? Why or why not?

The bean will have a method called **`rollTheDie`** that generates a random number with a value between 1 and 6 and assigns it to the **`oneDie`**. Here is the code to generate a random number:

```
Random rand = new Random();
oneDie = rand.nextInt(6) + 1;
```

The bean will have an **`equals`** and **`hashCode`** method that you can have NetBeans write for you. This will be used to determine if two die have the same value, commonly called a pair.

Next, let's write a test class called **`DieTester`**.

This class will have two methods. The first is the standard **`main`** method and it will be written as:

```
public static void main(String[] args) {
    DieTester dieTester = new DieTester();
    dieTester.perform();
    System.exit(0);
}
```

The second method is **`perfom()`** and it should create a **`TheDie`** object and then use it 100 times, displaying the value of the roll each time.

Run the program.

Have a bit more time? Add the ability to determine the frequency with which each value appears on the die.