

Assignment #1 – GamePieces

Date assigned: Thursday, February 4, 2010
Date due: Thursday, February 11, 2010
Points: 25

For this assignment, you are to implement a die and a pair of dice as Java classes. Each of the classes is to be implemented in a single project called PUNetIDJava (where PUNetID is your 8 character PUNetID) using Eclipse3.5. Further each of the game piece classes is to exist in a package called edu.pacificu.cs.GamePieces. The behavior of each class is defined as below. Add additional state as necessary for each class definition.

```
public class Die
{
    private int randomSeed = 1234;
    private int sideValue;

    // uses a predefined random seed value of 1234 to get a random number between 1 and 6
    // which is placed into sideValue
    public Die () {}

    // using the random value passed in for a seed for the random number generator, get
    // a random number between 1 and 6 which is placed into sideValue
    public Die (int randomSeed) {}

    // set the field sideValue to the value passed into set
    public void set (int sideValue) {}

    // return the sideValue
    public int get () {}

    // roll a single die producing a random value between 1 and 6
    // and store it into sideValue
    public void roll () {}

    // display the sideValue of a single die
    public void display () {}
}

public class Dice
{
    private Die die1;
    private Die die2;

    // creates two random die objects using the predefined randomSeed value from Die
    public Dice () {}

    // creates two random die objects using the random seed value passed in for a single die
    // the first die will use randomSeed and the second die will use randomSeed + 1
    public Dice (int randomSeed);

    // uses two die to initialize each die field of Dice
    public Dice (Die die1, Die die2) {}

    // initializes the field die1 to the value passed in
    public void setDie1 (Die die1) {}

    // initializes the field of die2 to the value passed in
    public void setDie2 (Die die2) {}

    // returns the total side value of both die
    public int getTotalDiceValue () {}

    // returns the side value of die1
    public int getDie1Value () {}
}
```

```

// returns the side value of die2
public int getDie2Value () {}

// roll a pair of dice
public void roll () {}

// display the value of each die separated by exactly one space
public void display () {}
}

```

For each class you are to create a JUnit test that has methods that will logically test all of the constructors and methods of the class.

For class Die implement the following methods in a JUnit test called DieTest.java:

1) `public final void testDie () {}`

This method is to do each of the following in the exact order given (1) create two Die objects using the default randomSeed value, (2) create two Die objects using the method **currentTimeMillis** as the random seed value, (3) display each Die object sideValue properly labeled, (4) roll Die1 one time and display the result properly labeled, (5) set Die2 to the value 6 and display the result properly labeled. The output of executing this method must look exactly like the following (your exact values will vary):

```

*****
*           testDie Execution           *
*****
Die1 is 3
Die2 is 4
Die3 is 6
Die4 is 2
Die1 is 1
Die2 is 6

```

2) `public final void testMultipleRolls ()`

This method is to create a single Die object using the predefined randomSeed and roll the Die 100,000 times. Use an array with six elements to collect the number of times a 1, 2, 3, 4, 5, and 6 were rolled. Output your results exactly as follows:

```

*****
*   testMultipleDieRolls Execution   *
*****
Side 1 occurred xxxxx times
Side 2 occurred xxxxx times
Side 3 occurred xxxxx times
Side 4 occurred xxxxx times
Side 5 occurred xxxxx times
Side 6 occurred xxxxx times

```

For class Dice implement the following methods in a JUnit test called DiceTest.java

1) `public final void testDice () {}`

This method is to do each of the following in the exact order given (1) create a Dice object using two new die objects that are created using the randomSeed value from currentTimeMillis, (2) display the dice value, (3) roll the Dice exactly one time, (4) display the value of the dice, (5) display the total of both die summed together. Output your results exactly as follows:

```

*****
*           testDice Execution           *
*****

Dice is 2 3
Dice is 6 5
Total Dice is 11

```

2) **public final void** testMultipleRolls ()

This method is to create a single Dice object and roll the Dice 100,000 times. Use an array with twelve elements to collect the number of times a 1, 2, 3, ..., 11, and 12 were rolled. Output your results exactly as follows:

```

*****
*   testMultipleDiceRolls Execution   *
*****

Value 2 occurred xxxxx times
Value 3 occurred xxxxx times
Value 4 occurred xxxxx times
Value 5 occurred xxxxx times
Value 6 occurred xxxxx times
Value 7 occurred xxxxx times
Value 8 occurred xxxxx times
Value 9 occurred xxxxx times
Value 10 occurred xxxxx times
Value 11 occurred xxxxx times
Value 12 occurred xxxxx times

```

Goals for Assignment #1:

1. Write a couple of simple Java classes using basic Java programming constructs
 2. Use the concept of a package to organize our classes
 3. Learn how to debug and test our classes using the JUnit framework
 4. Use SVN for revision control
 5. Become comfortable developing programs using Eclipse
 6. Use the Javadoc tool to create online documentation for your code
-

Specifics:

1. Save your program in a project folder called PUNetIDJava. Then drop the entire folder into the CS360AndroidDrop folder by 1pm on the day in which the assignment is due.
2. Your code is to be written using the development tools specified in the syllabus.
3. If you come to me with a question regarding your solution, I will have you load your project onto a machine in the CS lab. I will not look at your code on your computer or on paper as it just takes me too long to get at the problem. Further, I want you to bring in your textbook and lecture notes in case I want you to look up something. Remember, I'm not just a tell you the answer guy.
4. If you want help with a compiler error, you must be able to tell me exactly what statement you put in your code that caused the error and be able to isolate the error. If you have typed in a bunch of code and have not tested your code as you've gone along, I'm not going to help you sort out the mess. You've been warned!!
5. Print out your classes in a logical order such as the Die class followed by the Dice class which uses the Die class. Also, your output is to be double-side in color with a single staple in the upper-left corner.
6. Create the Javadocs for all of the public information in your PUNetIDJava folder as we discussed in class. The folder is to be at the same level as the src and bin folders.