Programming Lab #5

Rolling Dice

Background

This lab was developed by Maria and Gary Litvin and is used with permission.

This lab focuses on technical concepts we already know – the `if` and `switch` conditionals. But it introduces some new ideas you have not yet experienced. Because of this, you should read through the document "RollingDiceLab" up to the page numbered 25 now (if you have not already) and then return to this lab. There are some ideas that we will revisit later so don't panic if there's a lot of stuff you don't understand. Work to get an idea of what's going on and register that there is "stuff" you don't get and move on.

Don't go any further until you've read the RollingDiceLab document!

Objects

A deeper discussion of objects is at the very beginning of the reading. It's important for you do understand the thinking behind the design but don't worry about the discussion of `JFrame` and `Swing`. These are related to GUI development and we will not be responsible for any of that in this lab. It's important to understand that an object can represent something that is abstract – like the rules to a game like Craps.

Unified Markup Language (UML)

On the page numbered 19 there is a picture of classes and how they are related. This is an example of UML. The two main concepts to understand here is that the arrow indicates an is-a relationship (inheritance). For example DisplayPanel is-a JPanel. The diamond end indicates a has-a relationship (service). For example CrapsTable has-a RollingDie. The overlapping diamonds indicates that CrapsTable has more than one RollingDie.

Test Harness

The concept of a test harness is introduced with CrapsTest1.java. This is a small "utility" that allows the programmer to test one component of the program independent of the fully functional application. This could also be referred to as a "sandbox" but sandboxes tend to be more informal.

Setting Up

Note that you will need to unzip the files prior to using them. If you do not already have software to unzip files, you can download the free 7zip application from http://www.7-zip.org/ .

In JCreator:

1. Download and extract the zipped file containing the files and unpack them into a directory named "RollingDie". You may have to use a separate program like 7zip to extract the ZIP file (see link above).

   CrapsGame.java
   CrapsStats.java
   CrapsTest1.java
   RollingDie.java
   PL5-RollingDiceInstructions.pdf
   RollingDiceLab.pdf
   CrapsDemo.jar
   Craps.jar

2. Copy Craps.jar and put it in the same folder as you put the other jar files for this class. It should be: C:/ProgramFiles/Java/jdk1.8.0_11/jre/lib/ext

In DrJava:

1. Download and extract the zipped file containing the files and unpack them into a directory named "RollingDieFiles". You may have to use a separate program like 7zip to extract the ZIP file (see link above).

   CrapsGame.java
   CrapsStats.java
   CrapsTest1.java
   RollingDie.java
   PL5-RollingDiceInstructions.pdf
   RollingDiceLab.pdf
   CrapsDemo.jar
   Craps.jar

2. To create a project file,

   a. Run DrJava.
   b. Use the **New** command in the **Project** menu.
   c. Pick a name for the new project file, select a place to save it, and press the **Save** button. I suggest you save it in the RollingDieFiles directory.
   d. In the **Project Properties** dialog, edit the project settings as described below.
   e. Set the **Project Root** to the directory that corresponds to the default package: RollingDieFiles

f. Click on the **Add** button in the **Extra Classpath** section, find the Craps.jar file and press the **Select** button to add the Craps.jar file to the "Extra Classpath".
g. Press the **OK** button to close the **Project Properties** dialog.
h. Use the **Open Folder...** command in the **File** menu and select the directory you selected as project root in step 3.e., then press the **Select** button.
i. The source files are now displayed on the left side of the editor.

You are now ready to edit, run, and compile the project with DrJava.

In Eclipse:

1. Download and extract the zipped file containing the files and unpack them into a directory named "RollingDie". You may have to use a separate program like 7zip to extract the ZIP file (see link above).

   CrapsGame.java
   CrapsStats.java
   CrapsTest1.java
   RollingDie.java
   PL5-RollingDiceInstructions.pdf
   RollingDiceLab.pdf
   CrapsDemo.jar
   Craps.jar
2. Make the Java Project "CrapsProject"
3. Copy the following files and put them in your CrapsProject/src folder:
   Craps.java
   CrapsGame.java
   CrapsStats.java
   CrapsTest1.java
   RollingDie.java
4. Put Craps.jar in your CrapsProject folder.
5. Add Craps.jar as an external jar to the build path. If you've forgotten how to do this refer back to the DrawHouse instructions. Warning! Once you do this you will not be able to run the Craps game anymore by double clicking on the Craps.jar file.

~ ~

Note that you can double click on CrapsDemo.jar and play a few games of Craps. It is highly recommended that you do this until you completely understand the rules.

Coding Part 1

Starting on the page numbered 25 there are instructions to for programming the CrapsGame and Die classes. The notes below are intended to augment the

instructions on that page. Do NOT move forward before you have completed each step:

1. **Step 1:** - be sure to *carefully* read and understand the JavaDoc above each method you are required to write. It will guide you through exactly what you need to do to successfully write the method!
2. **Step 2** – note that you are to write the Die class AND its test bed (the console app) from scratch. Follow the instructions carefully. A description of the Die class was in the reading. Be sure to document all of the classes you create and the methods you write!
3. **Step 3** - another good example of a test harness. CrapsStats allows you to test your implementation of the rules quickly and easily without needing to worry about a GUI.

Coding Part 2

Read the text starting on the page numbered 31. Again, there may be some things you don't understand. Don't get bogged down with them! Read through the text until you get to section 7.13 (you don't have to read that part!).

1. **Carefully** review the code in the `drawDots` method of RollingDie.
    a. What do the `x1`, `x2` & `x3` variables represent in the context of a die? What about the `y1`, `y2` & `y3` variables?
    b. The case to draw a single dot representing '1' is shown. Use this as the framework to draw the dots for 2, 3, 4, 5, & 6.

What to turn in

When you are done with this lab upload the source code for the following files:

CrapsGame.java
Die.java
RollingDie.java

to the Programming Lab #5: RollingDice assignment in Canvas. Be sure to review both this document and the assignment rubric for a clear understanding of how the assignment will be graded.