HW03 - Submitted

Title	HW03
Student	Lodhia, Rahmaan Mohammed
Submitted Date	Sep 9, 2009 12:47 pm
Grade Scale	Points (max 100.0)

Instructions

Assignment 3

Due Wednesday September 9, 2009 11:55PM

3.1: Tip Calculator

For homework 3, you'll be creating a tip calculator to help with all the tricky math one has to do at restaurants. You'll make use of the Scanner class, loops, and if statements.

The program should work as follows:

1. Print out instructions for the user.
2. Prompt the user for the prices of the items, one by one. You do not have to worry about validating user input (that is, you can assume they enter correct dollar / cent amounts).
3. When the user enters -1 for a price, that means they are done entering prices.
4. Prompt the user for the percentage they want to tip.
5. Print the subtotal (sum of all the item prices), the tip (subtotal times tip percentage), and the total (subtotal + tip). You do not have to worry about rounding to the nearest cent.
6. Ask the user if they would like to split the check
7. If yes, then ask how many people will be splitting the check, and print the per person cost (total from step 5 divided by the number of people)

You should also javadoc your class, main method, and any additional methods you create.

Example Output

```
Tip Calculator

Enter the prices of your items. Enter -1 to enter tip percentage.

Item 1: $1.99
Item 2: $1.99
Item 3: $2.99
Item 4: $10.99
Item 5: $8.99
Item 6: $5.39
Item 7: $-1

Enter the tip percentage: 18

Subtotal: $32.34
Tip:      $5.8212
Total:    $38.1612

Split the check? (y/n) y
Number of people: 2

Per person: $19.0806
```

3.2: Dice Applet

For the 2nd part of the assignment, you'll create an applet that draws the faces of a die based on a random number.

For your reference:

en.wikipedia.org/wiki/Dice

Specs for the applet:

1. Call it DiceApplet
2. You should base your dice drawing on the following: en.wikipedia.org/wiki/File:Die_Faces.png
3. You may make them as fancy or simple as you want, but it needs at least a square face and the dots.
4. You must write a method (separate from paint()) that draws a die. It should take:
    1. the graphics object to draw with
    2. the die's number (1 through 6)
    3. the x and y location of the die
    4. the side length of the die
    5. the color of the die
5. In your paint() method, draw 3 dice using randomly generated numbers for their sizes, colors, positions, and roll values, using the above method. All of the dice must be drawn on screen (ie, no part of a die cut off), even if the applet is resized. The dice may overlap.
6. The width and height of the applet should be reasonable, around 500x500 pixels (big enough to draw 3 dice)
7. Create an HTML file called DiceApplet.html to display your applet.

Don't forget to javadoc your class and methods. In particular, the method in step 4 should have all of its parameters properly javadoc'ed. A (modified) example from the Java API:

```
/**
 * Fills the specified rectangle.
 * The left and right edges of the rectangle are at x and x + width - 1.
 * The top and bottom edges are at y and y + height - 1.
 * The resulting rectangle covers an area width pixels wide by height pixels tall.
 * The rectangle is filled using the graphics context's current color.
 *
 * @param    x      the x coordinate of the rectangle to be filled.
 * @param    y      the y coordinate of the rectangle to be filled.
 * @param    width  the width of the rectangle to be filled.
 * @param    height the height of the rectangle to be filled.
 *
 */
public void fillRect(int x, int y, int width, int height)
{
    ......
}
```

Hint: Since we don't know yet how to trigger a repaint event, if you're viewing the applet through AppletViewer, if you "wiggle" the size of the viewer's window, it will cause the applet to repaint, and redraw the dice.

Turn-in Procedure

Turn in the following files on T-Square. Double-check that you have *submitted* and not just saved as draft (if the submission is successful, you will receive an e-mail from T-Square within a few minutes). Also, be sure all of your files compile and run.

- TipCalculator.java
- DiceApplet.java
- DiceApplet.html

All .java files should have a brief descriptive javadoc comment.

Don't forget your collaboration statement. You should include a statement with every homework you submit, even if you worked alone.

Submitted Attachments

- TipCalculator.java (3 KB; Sep 7, 2009 12:32 pm)
- DiceApplet.html (1 KB; Sep 7, 2009 12:32 pm)
- DiceApplet.java (4 KB; Sep 9, 2009 12:47 pm)