HW02 - Submitted

Title	HW02
Student	Lodhia, Rahmaan Mohammed
Submitted Date	Aug 29, 2009 9:49 pm
Grade Scale	Points (max 100.0)

Instructions

CS 1331 Homework 2

Due Mon Aug 31, 2009 11:55PM

Introduction

In this assignment we will begin requiring you to comment your code and we will gradually build this level up over the term. For HW2 we will begin to "javadoc" our code. Javadoc is an application that is part of the JDK that you downloaded. You can run it on the command line by typing "javadoc *.java" which will create documentation for all the classes in your current directory. If you are using JGrasp, you can invoke javadoc by using Project->Generate Documentation from the pull-down menu. Javadoc creates the nice web-page based documentation that we have been using in class (the API).

Javadoc comments are a special form of comment. Javadoc recognizes a comment that starts with /** as a javadoc comment. For each new program assignment we will introduce new javadoc features. For HW2 we will create the following javadoc comments:

The class header:
Start your class header with the javadoc comment /**. Then write a short description of the class. Note that each line starts with an asterisk (*) and space. End the header comment with */. For example:

```
/**
* This is HW2, problem 1.
* This class rids the Earth of bad guys.
*/
```

Javadoc class header comments should be placed below any import statements and right above the class header itself:

```
import javax.swing.JApplet;

/**
* This is HW2, problem 1.
* This applet draws some shapes and text.
*/
public class MyApplet extends JApplet {
```

You should also place good comments into your code when the algorithm you are using is not obvious. For the statement

```
int perimeter = a + b + c;
```

a good comment might be:

```
// calculate the perimeter by summing the lengths of the sides
```

A bad comment would be:

```
//add a b and c together
```

A frequent syntax problem developers have is mismatched braces. One commenting technique that can really help you is to comment the close brace with a short description of what it is enclosing. For example:

```
} // end switch(x)
 } // end main method
} // end class def Demo
```

We'll be looking at more javadoc and commenting issues in future HWs. For now, if you have more questions, check out the javadoc info page under the Java Resources page on the class website, or talk to your TA.

Assignment 2.1: Movie Poster Applet

You are applying for a job with the marketing division of a film company. To assess your technical skills, the interviewer has asked you to create an a movie poster in the form of a Java applet. The movie can be real or imaginary, and the applet must advertise the movie in some way. Name your applet MoviePosterApplet. The applet can be as simple or as elaborate as you want.

Since this is a test of your technical skills, the interviewer have given you some restrictions:

- Use at least three different drawing methods (excluding drawString) in the applet. Filled and unfilled count as two separate methods, so drawRect and fillRect (for example) count as two different methods, and there are others in your textbook. drawLine also counts as a valid method.
- Include some text somewhere in the applet (like the movie title and advertising slogans). Remember, drawString does *not* count towards your three drawing method requirement.
- Use at least three different colors in your applet (the default white background doesn't count).
- One of your three (or more) colors should be a custom-created color (i.e. use the Color constructor).
- Do not use external images--the art in this applet should be all programmed by you (although you are free to re-create a real logo using applet drawing methods).
- The size of the applet should be something reasonable, and fit on a typical screen.

When you've finished your applet, create the html file to display it. Call the file "MoviePosterApplet.html". The code for creating this page is in your book and is covered in lecture and recitation. Make sure you edit the file in plain text (ie, Notepad for Windows).

Note for Mac users: If you're using TextEdit to create your .html file, be sure you are editing in plain text mode:

1) Open TextEdit
2) Click Format -> Make Plain Text.
3) type in the html code
4) click save
4a) type in a filename with .html on the end
4b) click save
4c) click "Use .html"

Be sure to javadoc your applet with a brief description.

Turn-in Procedure

Turn in the following files on T-Square. When you're ready, double-check that you have *submitted* and not just saved as draft.

- MoviePosterApplet.java
- MoviePosterApplet.html

All .java files should have a 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

MoviePosterApplet.html (1 KB; Aug 29, 2009 9:49 pm)
MoviePosterApplet.java (1 KB; Aug 29, 2009 9:49 pm)