CSC 210 Program #2

A GUI of Lists of DrawableShapes
Jan 16, 2008

Logistics

Program #2 is:

- **Worth** 5 points, or 5% of your grade
- **Due** Monday Jan 28, 2008
- **Covers** a trinity of Chapter 3 inheritance stuff, some Chapter 4 list stuff, and a touch of Appendix C GUI fun

Description

Program #2 presents a nice GUI that allows your user to place any number of shapes into a window. There are three parts.

UNO: Chapter 3 Inheritance stuff

Go grab your textbook and turn to page 164. No, go ahead… I'll wait.

[pause]

In Program #2, we're going to complete and extend the "Drawable Shapes" project described on pages 164-165. For Program #2, you'll start from the code in the book. I have placed a copy on the k: drive `common_area` folder. It's also at the publisher's web site.

Please add the following to complete your program:

- Complete any missing Java code not given by the textbook
- Write the `Circle` and `RtTriangle` classes... use the book's `Rectangle` class as your guide
- Add your recursive tree from Program #1 as a `DrawableShape`. What do you think the area and perimeter of an R-tree should be?
- Add a new `DrawableShape` of your own... it has to be something you can draw, like a line (probably the easiest), a square (which would be a subclass of `Rectangle`), text, or what were we drawing in class… a rhombus or trapezoid or something.

DOS: Chapter 4 List stuff

This is probably the easiest part. Your user wants to create many `DrawableShape` objects. So, you'll probably want a list to keep track of them all.

I think that I used Java's `ArrayList` to do this, but whatever. In `paintComponent()`, you'll want to traverse the list and draw each `DrawableShape` in there.

TRES: Appendix C Java Gui stuff

You'll need a way for users to add shapes to the window as we go along. I suggest a GUI ala Appendix C. Along the lines of "Think, then code", please sketch out your GUI and then figure out how you're going to do it.

I expect a nice, clean GUI, but nothing super-elaborate. You can simplify your interface (if you want) by placing the shapes at random coordinates and giving them random colors. I got really tired of specifying that stuff when I did mine.

QUATRO: Put a bow on it

Your Program #2 should run as a Java application and an applet. Please make your applet available from your Noctrl web site.

Program #2 has plenty of creative flair already built-in with your new `DrawableShape` and GUI decisions. If you get "into" this program and want to do something more, I will accept proposals for 1-2 points of extra credit. Please get prior approval from me if you want to do something like this.

On the 28th, we'll all peek at your results in class. Good luck!

Grading

As you did for Program #1… by the due date, please place your work for Program #2 in your folder on the k: drive. I'll be looking for:

1. **Your README file** – describing the state of your program... what works, what doesn't, what your "flair" is, etc.
2. **Your Net Beans folder** – including your Java source code, class files, etc.
3. **Your Javadoc** – generate using the "Build/Generate Javadoc" menu
4. **Your applet** – create a web page at w:/index.htm with an Applet of one of your favorite trees. Please include a link to your Javadoc.
5. **Your printout** – Please print one source file… your most important one, so that I have someplace where I can scratch my comments.

I expect your code to be beautiful. "No crappy code"™

all good… yow, bill

Additional Notes

I have given you a lot of flexibility in this program, so please don't be shy about asking for clarification on something.

Remember:

- The Java API at: java.sun.com/j2se/1.5.0/docs/api/index.html
- The nice GUI examples in Appendix C
- My email: wtkrieger@noctrl.edu start early so that you can ask for help if you need it!