Abstract Factories and the Shape Calculator

Eric Cheng, Dung “Zung” Nguyen, Mathias Ricken, Stephen Wong
Dept. of Computer Science
Rice University
Houston, TX 77005
+1 713-348-3835

ericc@rice.edu, dxnguyen@rice.edu, mgricken@rice.edu, swong@rice.edu

ABSTRACT

The Shape Calculator is an assignment targeted at CS1 students in an objects-first curriculum. It can serve as a powerful yet entertaining example of the advantages of object-orientation.

Categories and Subject Descriptors

D.1.5 [**Programming Techniques**]: Object-oriented Programming

General Terms

Design.

Keywords

CS1/CS2, objects-first, design patterns, abstract coupling, loose coupling, pedagogy, assignments, software quality.

1. INTRODUCTION

In our submission, we describe the Shape Calculator, an assignment that has been used in our institution’s COMP 201 objects-first introductory course. It is sufficiently complex yet entertaining for the students, which makes it suitable as a compelling example for the benefits of object-oriented design and programming.

The Shape Calculator is a GUI application that can compute the area of an arbitrary shape. The class representing this shape can be compiled at a later time and loaded into the program at runtime.

2. ASSIGNMENT DESCRIPTION

Create a GUI application that computes the area of an arbitrary shape that can be dynamically loaded at run time. The shape can have an arbitrary set of configuration properties (e.g. width, height, color, etc.). This assignment is designed as an introduction to GUI programming in Java, where the GUI programming, which besides being a valuable skill unto itself, is used to emphasize and illustrate compelling usage of encapsulation, inheritance and polymorphism.

3. WHAT MAKES IT SO NIFTY?

This assignment is interesting and useful for several reasons:

- The assignment mixes fun with theory--motivates the students to learn important concepts and techniques.
- With straightforward code, the assignment demonstrates powerful OO capabilities that are very difficult to replicate using procedural techniques.
- The exercise covers a broad spectrum of skills and concepts without being overwhelming (see the list of skills list in section 5).
- The assignment focuses on the thought processes involved with the design of OO systems.
- It is orchestrated step-by-step process that gradually gives more rein to the students.
- Students create a fully functional, complete, non-trivial GUI program from scratch, not just a "toy" function in a pre-built system.

4. TARGET AUDIENCE

The target audience for this assignment is CS1 students in an objects-first curriculum. The Rice students working with this assignment are nine weeks into CS1 and have already seen polymorphism, several elementary design patterns (see prerequisites in section 7), lists and recursion.

5. IDEAS AND SKILLS INVOLVED

When working on this exercise, students gain understanding in the following areas:

- How to create a simple window frame and populate it with panels, buttons, labels and text fields.
- Using anonymous inner classes as event listeners in GUIs.
- Using abstract factories to manufacture products whose concrete type is immaterial to the client.
- Using anonymous inner classes to define concrete instantiations, especially in the context of factories.
- Using the closure properties of anonymous inner classes to directly access both instance and final local variables.
- Using graphics calls to paint a simple shape onto a component.
- Understanding the service nature of a component in a

OOPSLA’04, Oct. 24–28, 2004, Vancouver, British Columbia, Canada.
ACM 1-58113-833-4/04/0010.

framework, e.g. painting service of a GUI component.

- Numerous demonstrations of polymorphic behavior, especially those that cannot be replicated with conditional statements.
- Using an incremental process to build a complex software system by gradually adding components and capabilities.
- Using simple exception handling.

6. LENGTH OF ASSIGNMENT

At our institution, the assignment is a laboratory exercise (1.5 hrs) that extends into a week-long homework assignment (3-6 hrs).

7. PREREQUISITE MATERIAL

The students should be familiar with and had some programming experience with:

- Polymorphism and inheritance
- Abstract structure and behavior
- Singleton design pattern
- Delegation model programming

They need only be acquainted with and not necessarily adept at:

- Factory design pattern
- Composite design pattern
- Syntax for anonymous inner classes

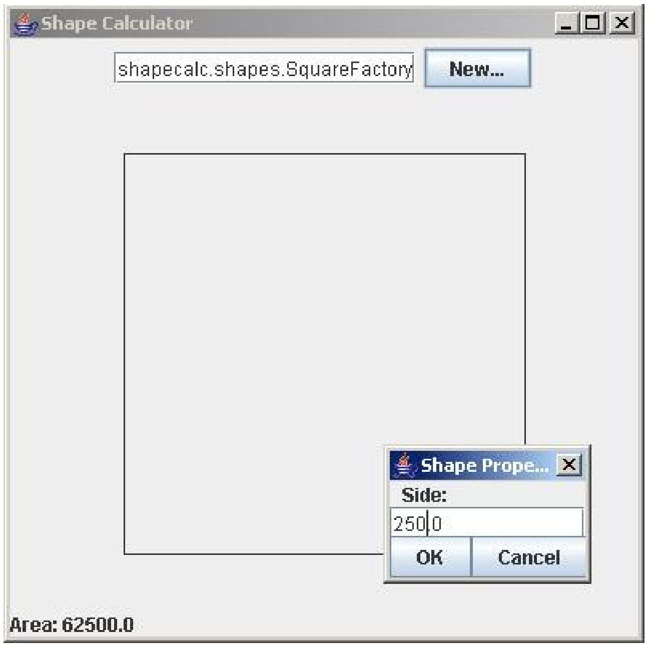
8. DIFFICULTIES TO WATCH FOR

Instructors using this assignment should be aware of the following difficulties:

- Understanding the service-oriented aspect of screen painting. Students often struggle with the inverted control aspects of a component in the GUI framework.
- Understanding the communications aspects of closures in anonymous inner classes.
  - Why local variables are required to be final.
  - Utilizing direct access to variables that are outside the anonymous inner class but within its closure.

9. INSTRUCTION MATERIALS

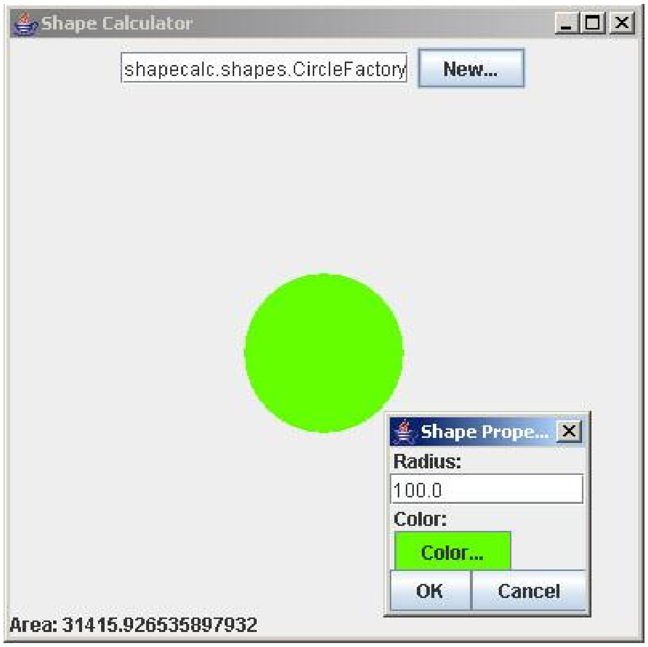
The assignment, a demo, the solution, and supporting lectures have been submitted to the Educator's Symposium. The actual assignment at our institution can be found at http://www.owlnet.rice.edu/~comp201/04-spring/labs/lab09/ .



shapecalc.shapes.SquareFactory

Side: 250.0

Area: 62500.0



shapecalc.shapes.CircleFactory

Radius: 100.0

Color:

Area: 31415.926535897932