

THE MOST EFFECTIVE APPROACH FOR TRUE OBJECT-ORIENTED PROGRAM DESIGN

Tanya Cashorali
College of Computer Science
Northeastern University
Boston, MA
cash21@ccs.neu.edu
Faculty advisor Viera K. Proulx

DESCRIPTION

The purpose of this project is to display the use of the Design Recipe pedagogy and the idea of focusing on the design of the structure of data, the classes, and class hierarchies, rather than the design of program actions as the core idea for program design. The Design Recipe approach is superior to other programming methods in many ways.

I started this project by first examining Carl Alphonces Java code for a calculator program in the May 2004 issue of Curricular Patterns [1]. The code was difficult to understand and did not showcase true object-oriented design. I examined another poorly designed calculator program found in Objects First with Java A Practical Introduction using BlueJ [2]. Both of these solutions contain many unnecessary if-else statements and demonstrate erroneous use of object-oriented languages. I knew if I followed the Design Recipe, which consists of a structured set of steps to follow during the design process, that I would arrive at a better solution. My first goal was to design a simple calculator with the four basic arithmetic functions (add, subtract, multiply, divide). I organized all of the input and output data into a table, which led me to discover the structure of the available data. The knowledge of these data definitions allowed me to create a comprehensible class hierarchy and diagram. A crucial step before development is writing examples for every function. The examples guided me to the realization that I could abstract the arithmetic operations into one abstract Operator class. This allows for polymorphism because the operation method will respond differently to each instance of the Operator class. Polymorphism is a requirement of any true object-oriented programming language. The next step was to arrange

all of the available data into a template. A vast majority of the program was already written at this point before actual coding even took place. I finished by filling in the necessary functions and testing them all. Finally, I modified this basic four-function calculator into a more complex calculator easily due to the organized design and abstraction as well as my understanding of the structure of the data.

The utilization of the Design Recipe will lead a programmer to the best object-oriented design possible. It provides a systematic approach to solving any problem by focusing on the structure of data rather than the design of program actions. Identifying the structure of data first is the key to observing similarities and abstracting over them, thus understanding the pattern. Alphonces approach focuses on the algorithms, which produces a complex solution that is a poor example of the use of design patterns. The Design Recipe also stresses documentation and testing of each method, forcing students into good design habits.

The Design Recipe pedagogy is necessary for designing a true object-oriented solution while understanding the data structures, class hierarchies, and underlying patterns. I have found it helps me in my other Computer Science courses such as Computer Organization and Design and Object-Oriented Design. My poster will highlight the specific steps of the Design Recipe and how it guided me through each phase of designing the calculator. It will also exhibit the implementation of the code into a GUI, therefore resulting in a fully functional graphical calculator.

References

- [1] C. Alphonse, Pedagogy and Practice of Design Patterns and Objects First: A one-act play. *ACM SIGPLAN Notices*, 39(5):11–14, May 2004.
- [2] D. J. Barnes and M. Koelling. *Objects First With Java: A Practical Introduction Using BlueJ* Prentice Hall, 2003.