Design patterns
Admin

- Final exam time
- Milestone 2 deadline tomorrow!
Design Patterns

Elements of Reusable Object-Oriented Software

Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides

Foreword by Grady Booch
What are design patterns

• Solutions to specific problems in OO software design
• 23 patterns in 3 categories
  • Creational
  • Structural
    • Composite
    • ...
  • Behavioral
    • Observer
    • Interpreter
    • ...
Observer

• One to many relationship
  • The many need to know changes in “one” immediately

• Example
  • Facebook feed
  • Lines & rectangles
  • ...


Example

• If a person changes its status, how to let all his “subscriber” knows?
Example

• What if there are different types of subscribers?
for each view in views
v.update();
Can you think of some examples?
Composite pattern

- Tree hierarchy
How to build a tree and traverse it?

```c
struct node{
    struct node* left;
    struct node* right;
    int val;
    int sum(){
        ...
    }
}
```
How to differentiate leaves and others?

struct leaf{
    int val;
    int sum(){ return val;}
}
How to accommodate different types of internal nodes?

• Examples
  • struct node or struct leaf?
  • Book
  • Graphics
Class diagram

```
// Container functionality
// for each element
for (int i = 0; i < elements.length; i++) {
    elements[i].doThis();
}
```
Can you think of some examples?
Interpreter

• What is an interpreter
  • Language, compiler

• Example
  • Boolean expression
    • Abstract syntax tree

\[ a \&\& b \mid\mid !c \]

A parser will turn this into an abstract syntax tree, and then an interpreter will evaluate the tree. How to write a program to do the tree-based evaluation?
Class diagram

- Client
- Context
- TerminalExpression
- CompoundExpression

- AbstractExpression
  - +solve(inout Context)

Perform "parent" functionality then delegate to each "child" element. "Context" is data structure for holding input and output.
Strategy

• Multiple variants of one algorithm
• Different types of objects only differing in behavior

• Example
  • Different type of printing for an expression
Class diagram
-- encapsulate algorithms into class

Program to an interface, not an implementation.

Client \rightarrow «interface» Abstraction

+/doSomething()

ImplementationOne
+/doSomething()

ImplementationTwo
+/doSomething()
Alternative solutions

• If in C

• Super-class on the data side

• Template in C++
Other examples

• Different sorting
• Different rendering
• What else?
Template

• Provide a skeleton for similar algorithms

• Example
Class diagram

```
FrameworkClass

+templateMethod()
+stepOne()
+stepTwo()
+stepThree()

ApplicationClassOne

+stepTwo()

ApplicationClassTwo

+stepTwo()
```

stepOne();
stepTwo();
stepThree();
Visitor

• How to add a class of operations for a variety of data classes?

• Example
  • Different operations for AST nodes
  • Different operations for Person (Female, Male)
The concrete types of the Element and Visitor objects have been "recovered". Perform the work appropriate for their pair of types.
Visitor

• What is it good at?
  • If you add operations (Visitor classes), the interface of the Element classes remains unchanged

• What is it bad at?
  • If you add new Element sub-class, significant changes are needed for the Visitor side

• Double-dispatch
  • Imagine two dimensions of a function call
    • The exact algorithm
    • The type of data this algorithm works on
  • You will get chance to make choice along both dimensions dynamically, using visitor pattern
Creational design patterns
Factory Method

• Lets a class defer instantiation to subclasses
  • No need to decide which subclass I want to use statically

• Example
  • Date (US style, Europe style, Chinese style, ...)
  • Window
Class diagram

```
Product
  +FactoryMethod()
  +AnOperation()

ConcreteProduct

Creator
  product = FactoryMethod()

ConcreteCreator
  +FactoryMethod()

```

return new ConcreteProduct
Abstract Factory

• For creating families of related or dependent objects without specifying their concrete classes

• Examples
  • Date, currency, data
  • Window, mouse, scroll bar, …
Class diagram