

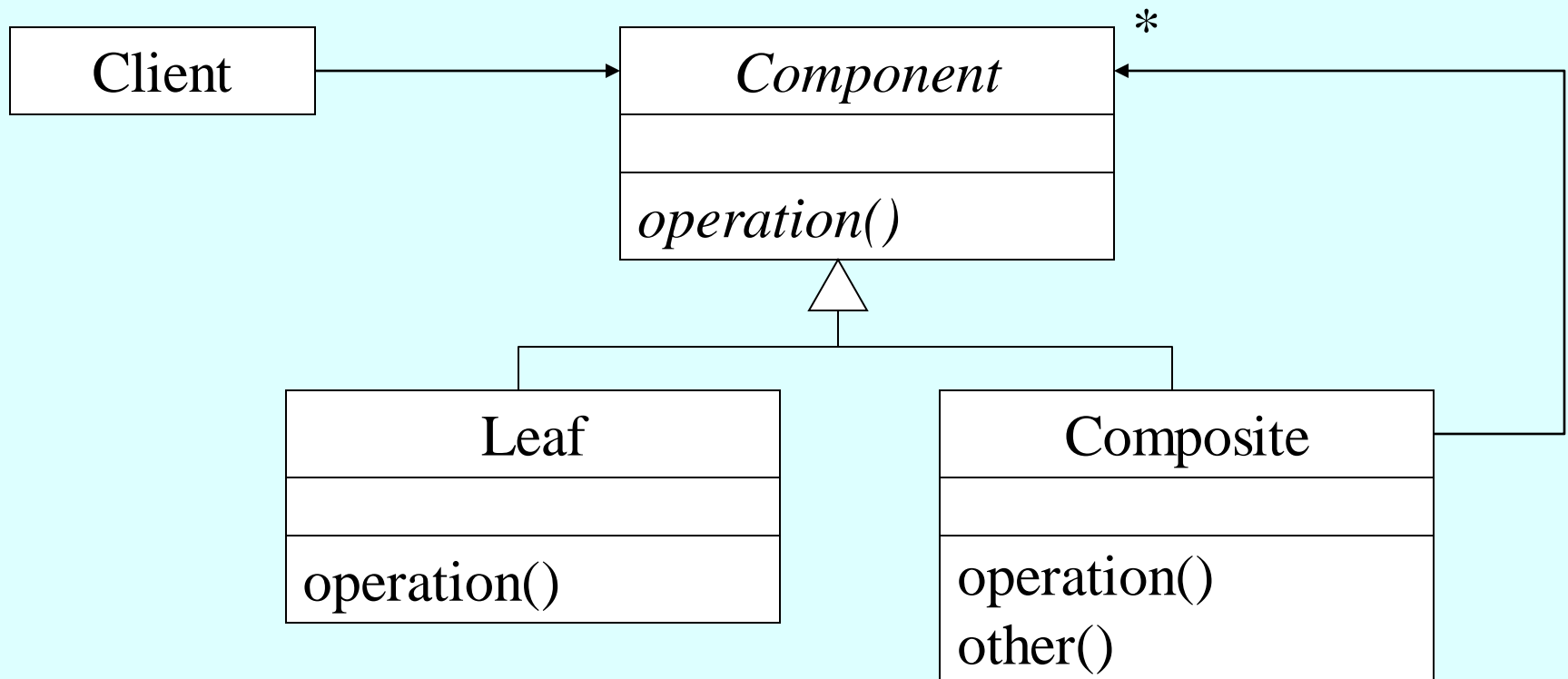
Composite Pattern

A composite is a group of objects in which some objects contain others; one object may represent collections, and another may represent an individual item, a leaf.

The composite class diagram enables a tree structure – a hierarchy.

The operations are appropriate for processing and traversing trees.

Composite Pattern



Note that Composite has references to possibly many Components.

Each node of the Component structure can respond to some common operation(s). A client can send the common operation to Component and the structure responds “appropriately”.

The client sees Component ... the client does not see the complexity of Component.

Composite Pattern

First example

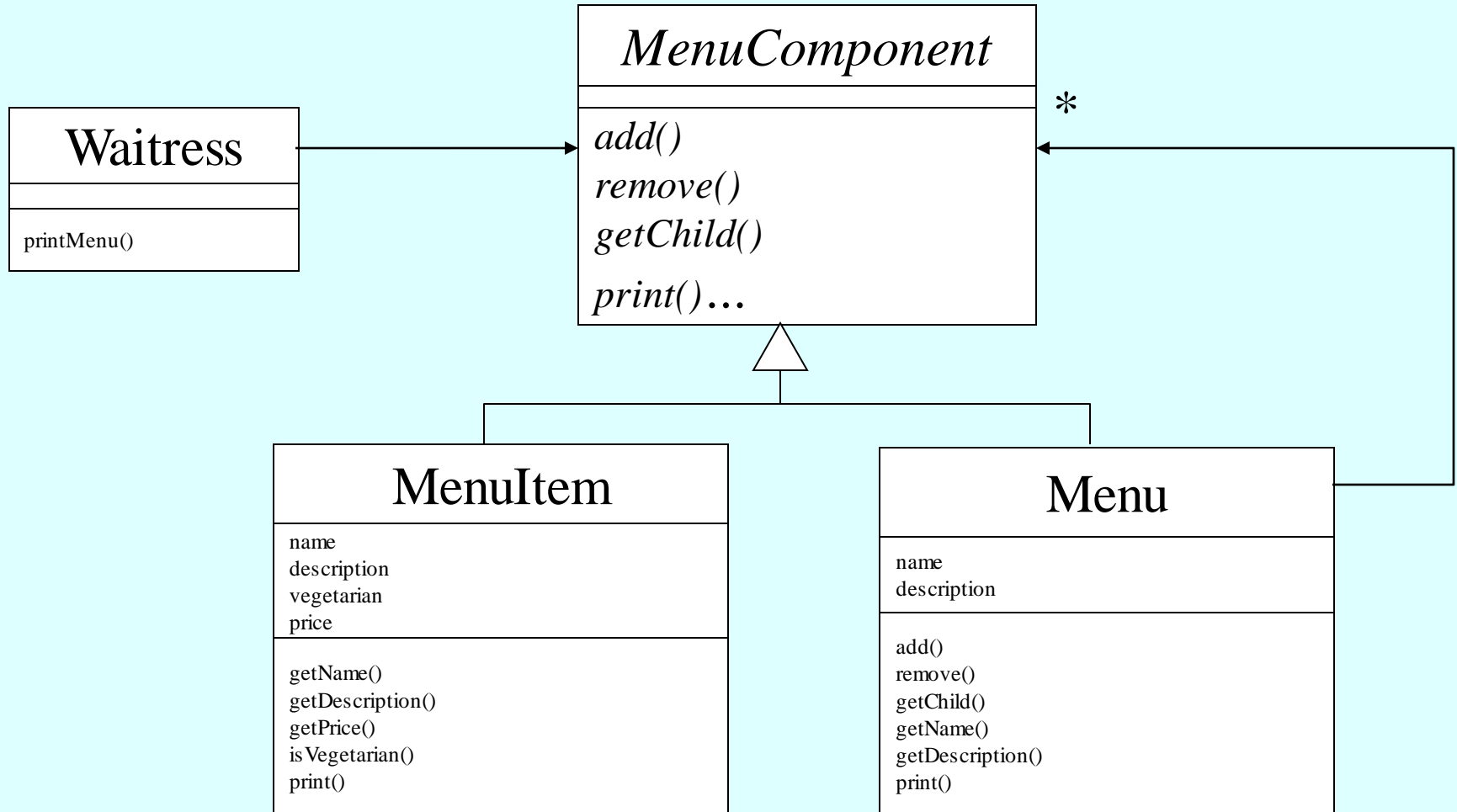
- Class diagram
- One operation, display. How is display handled by the various classes?
- How are associations implemented?
- What hierarchy is instantiated?

Composite Pattern

Consider the Head First example

- What is the class diagram?
- Who is the client?
- What operations are defined for the component, the composite, and the leaf? How are they carried out?
- How are the associations implemented in the code?
- What is the hierarchy that is instantiated?

Composite Pattern – Text example



Composite Pattern – Text example

Menu
menuComponents: ArrayList
add() remove() getChild() print()...

Note the text implements the association from Menu to MenuComponents with an array list data type.

The association from Waitress to MenuComponent ...

Composite Pattern – Text example

A waitress can ask a menu component to print itself... `print()` is a common operation.

What happens when a waitress sends `print()` to a menu component?
(Given a specific hierarchical menu, what messages are sent?)

Composite Pattern – Text example

In the Menu class, `print()` is appropriate for an internal node ... it sends `print()` to each of its children

```
public void print() {  
    System.out.print("\n" + getName());  
    System.out.println(", " + getDescription());  
    System.out.println("-----");  
  
    Iterator iterator = menuComponents.iterator();  
    while (iterator.hasNext()) {  
        MenuComponent menuComponent =  
            (MenuComponent)iterator.next();  
        menuComponent.print();  
    }  
}
```


Composite Pattern – Text example

In the MenuItem class, print() is appropriate for a leaf:

```
public void print() {  
    System.out.print("  " + getName());  
    if (isVegetarian()) {  
        System.out.print("(v)");  
    }  
    System.out.println(", " + getPrice());  
    System.out.println("    -- " + getDescription());  
}
```