Visitor

10-25-2005
Do you have any questions about the reading for today?
Visitor

- This pattern provides a way to perform an operation on all of the elements of some structure. This allows you to change the operation without having to alter the code for the structure itself.
- To do this we define a Visitor interface with a method that is called when an object is “visited” that method also accepts the object being visited as an argument.
- The ConcreteVisitor can collect information, or might be like a simple functor.
Example

- GoF uses an example of a compiler where the code is compiled to a meta-format and then we pass through that format several times doing different things.
- Without visitor, we need a different method for each type of pass we want to do. That adds a new method to every type in our meta-format. With visitor we just create a new ConcreteVisitor.
- More familiar to you would be having a binary tree and wanting to do different things to the elements.
Benefits and Drawbacks

- The primary benefit is that it is very easy to add new operations. It is also easy to change that functionality for a single visitor because it sits in a single visitor class.
- This also keeps things related to different operations separated into different visitors.
- When you want to add a new type to the structure, that can be hard. This is because different visitors might handle each subtype differently.
More Benefits and Drawbacks

- Visitors can work across class hierarchies. An iterator can't do this. The visitor can also be more efficient than an iterator in some cases.
- The visitor can accumulate state where data would have to be passed through if the methods were part of the objects.
- Visitors often require you to break encapsulation in some way.
Does anyone want to present anything this week for progress on the project working in Java?

It is critical to understand that sometimes you just have to pick a way to do something and do it.