Reuse at Design Level: Design Patterns – V

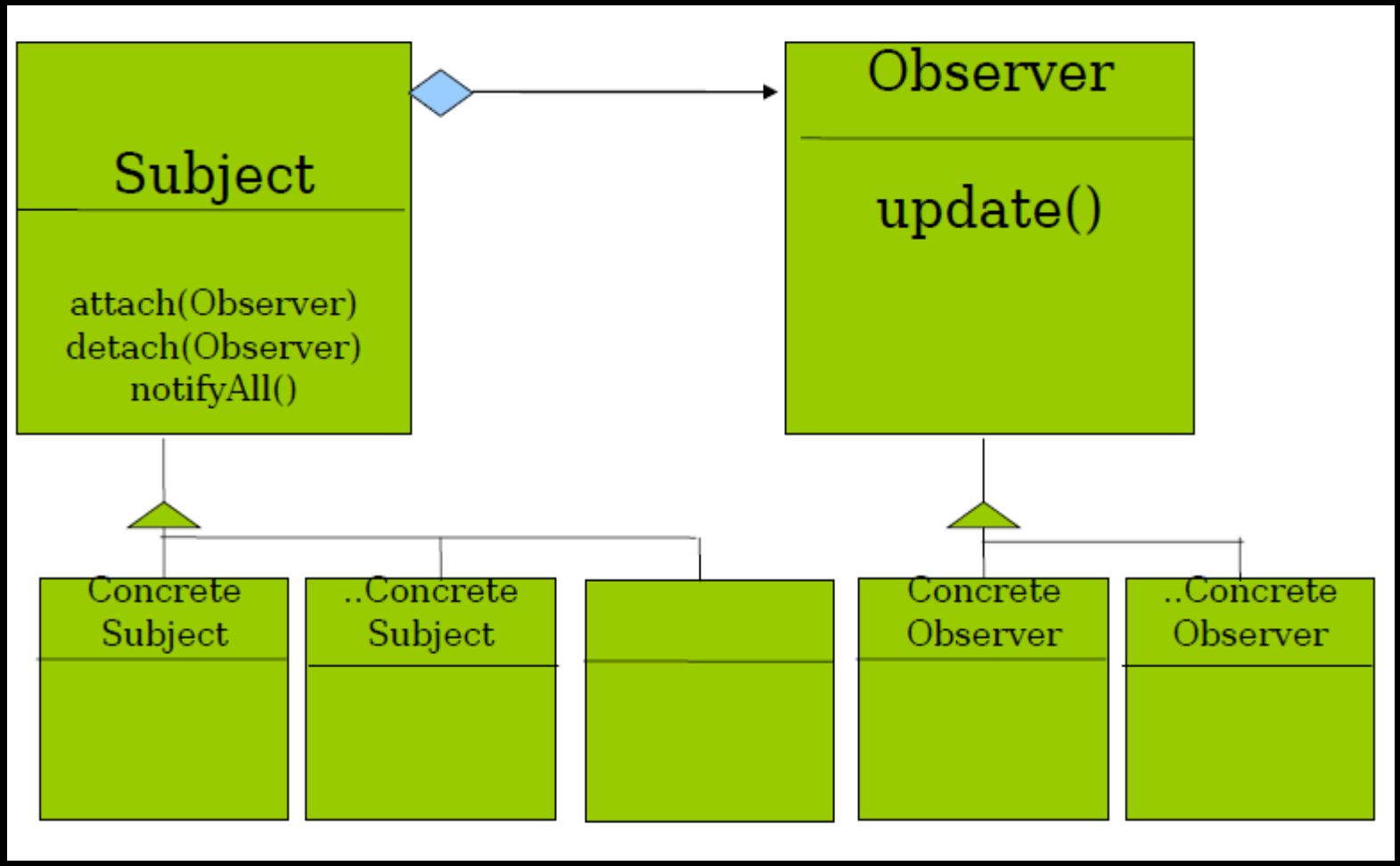
Classification of Patterns (Cont..)

		Purpose		
		Creational	**Structural**	**Behavioral**
Scope	**Class**	**Factory Method**	**Adaptor Class**	**Interpreter** **Template Method**
	Object	Abstract Factory Builder Prototype Singleton	Adaptor (Object) Bridge Composite Decorator Façade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Problem: When an object changes its state, its dependents are updated

- 1-* dependency between observed and observers
- One observed, Many observers

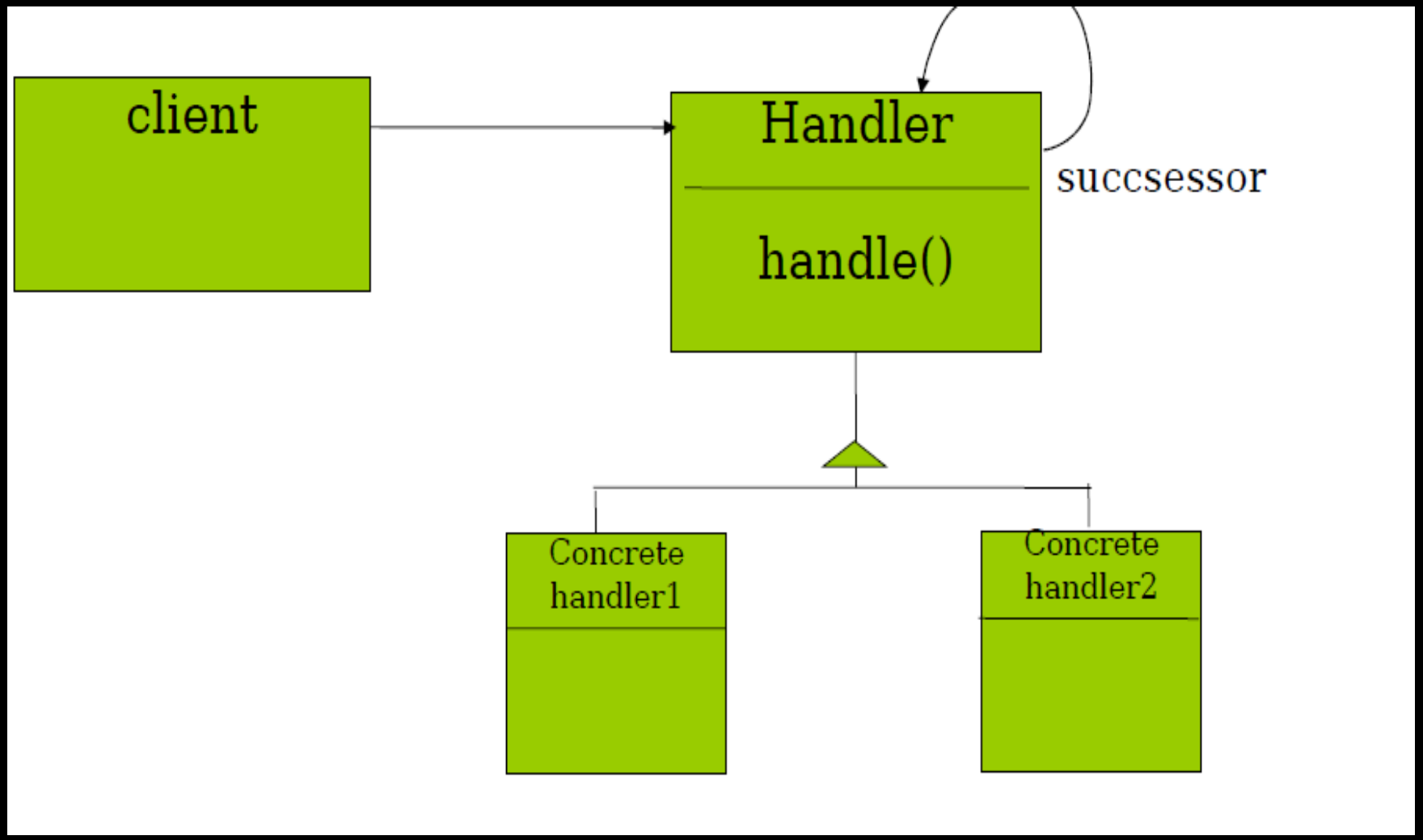
The Observer Pattern



A Problem: one of many request handlers

- A request may get handled by one of many objects
- The sender may not exactly which one can handle the request
- One way is to chain the handlers..

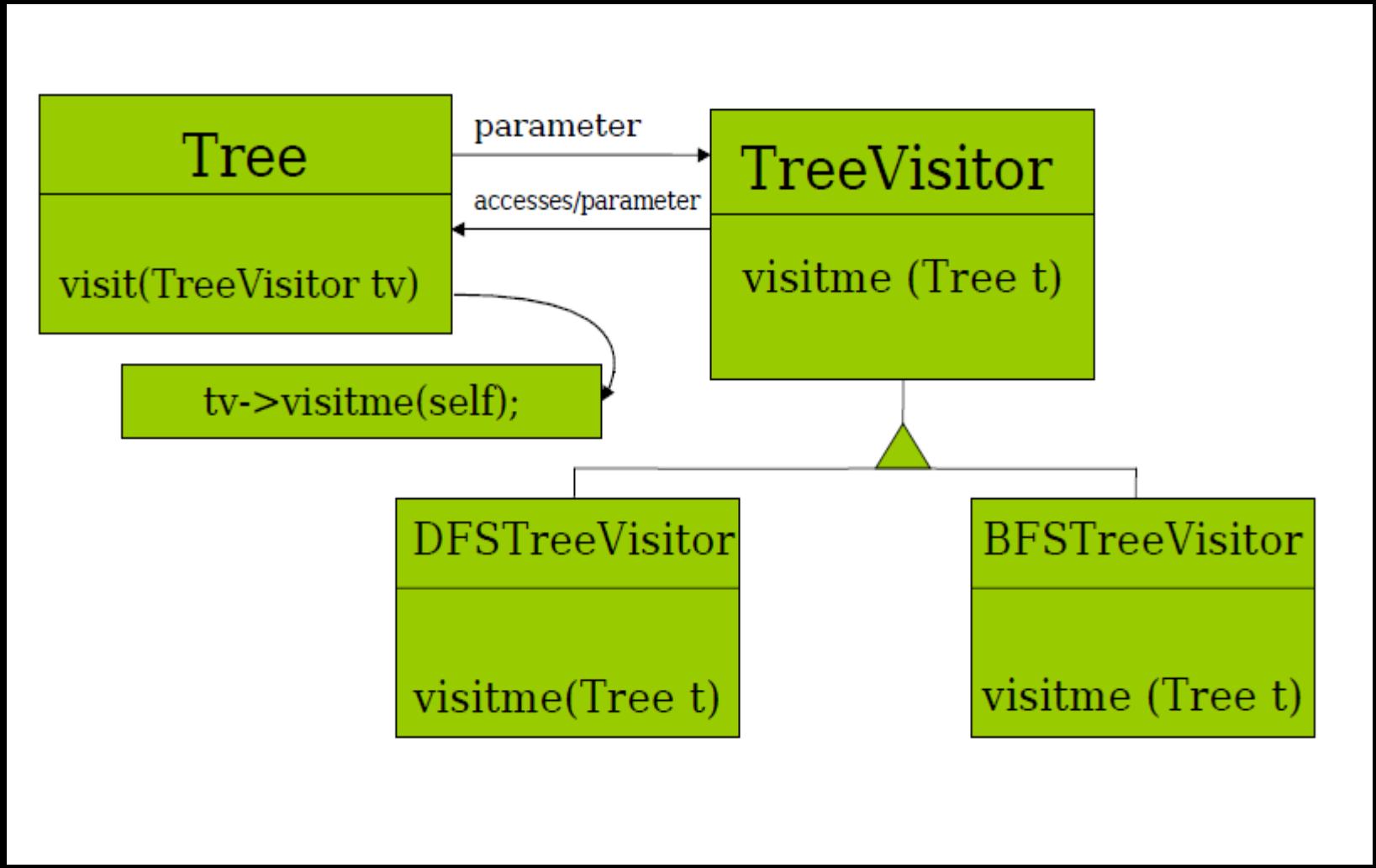
Example: Chain of Responsibility?



A Problem: modeling of operations such that they can be externally added to a class

- You have an existing class that is ready to accept new operations
- A new operation can be plugged in by means of an external object which can handle such a new operation
- For example:
  - Class Tree has operations to form a tree and a plug point through which new operations can be added

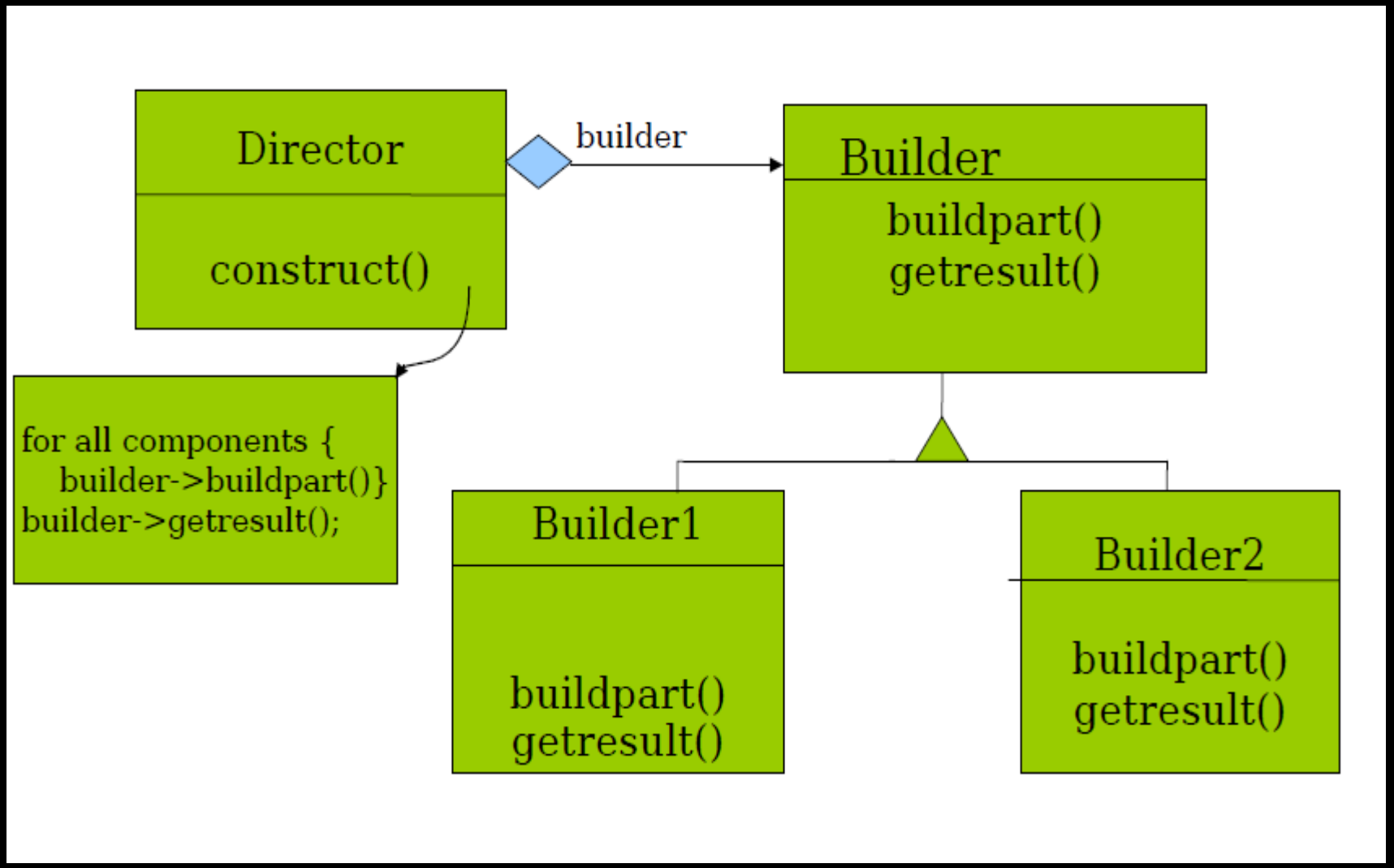
Example: Visitor Pattern



A Problem: Separate construction of objects from their representation

- E.G. transforming from one representation into another
- A transformation function for each component
- Finally you can obtain the whole transformed representation

Builder Pattern



Thanks