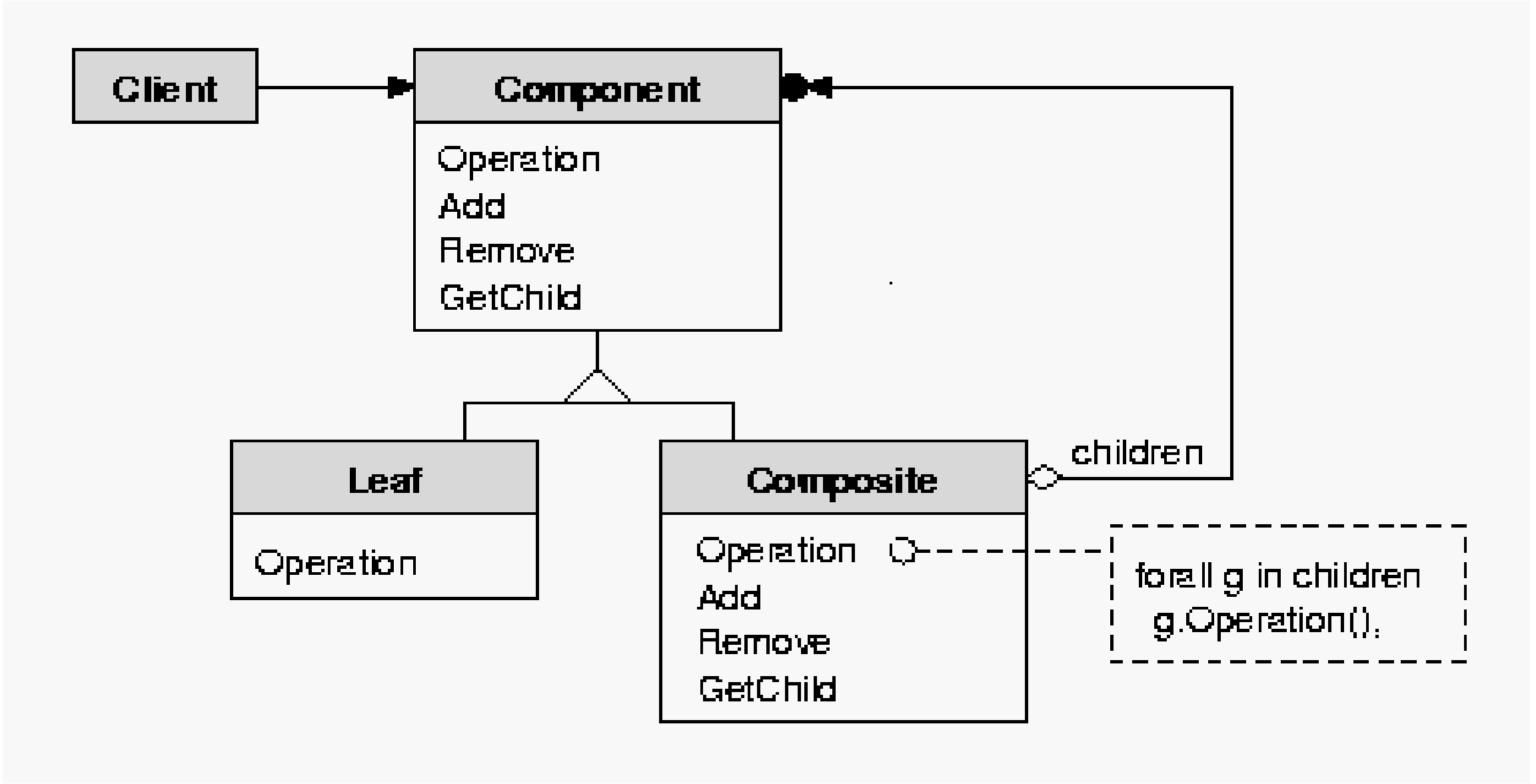
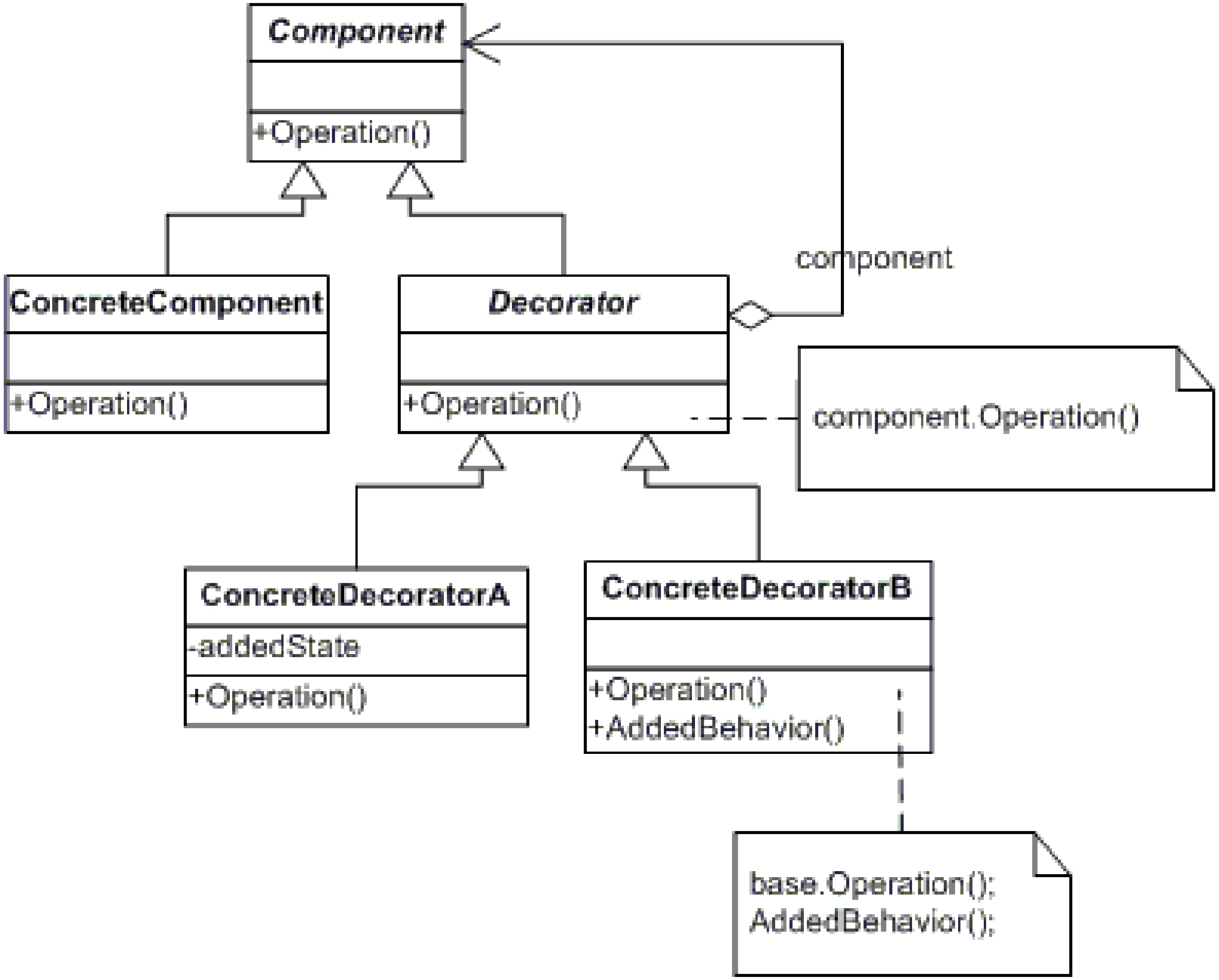
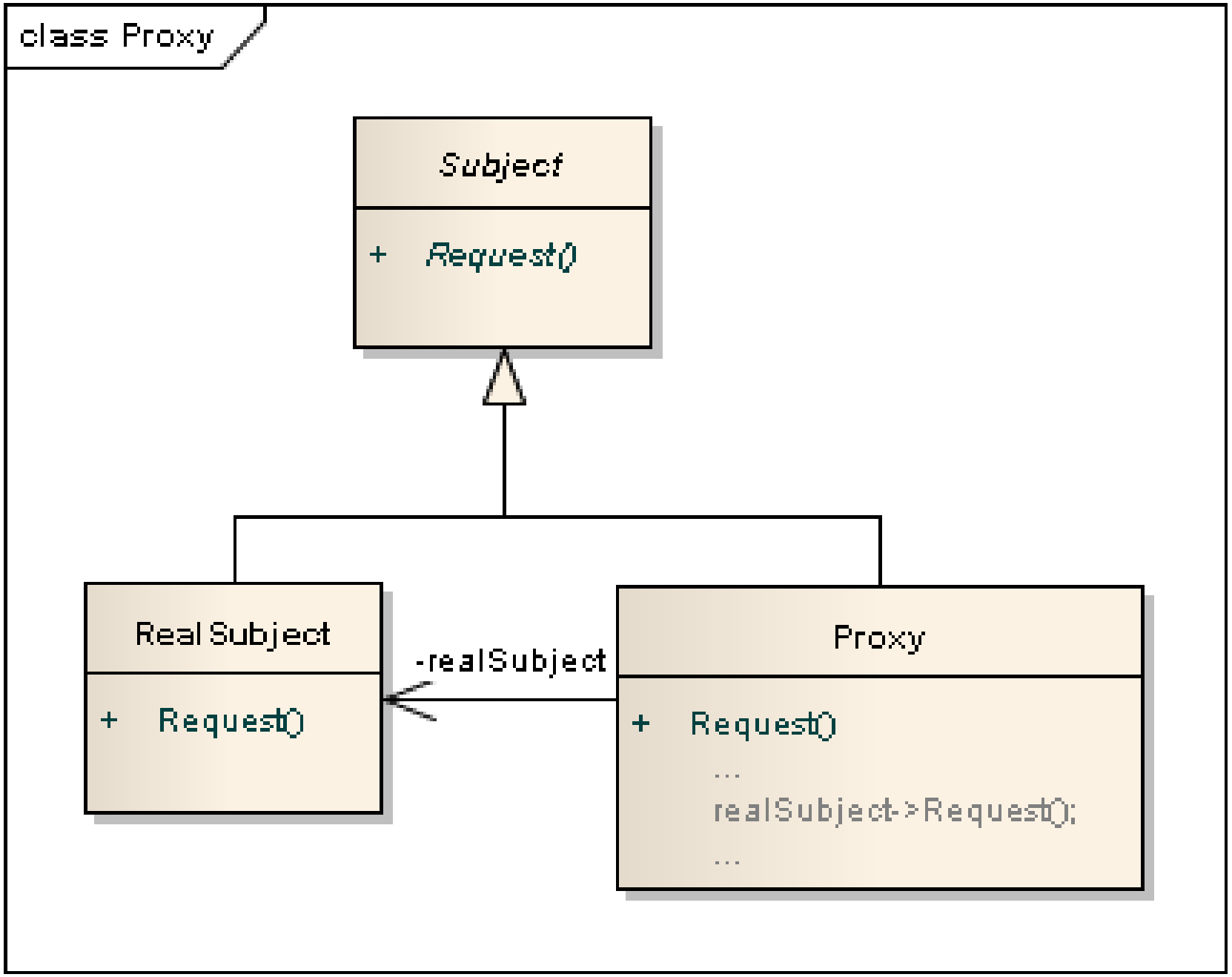
COMPOSITE


Client
Component
Operation
Add
Remove
GetChild
Leaf
Operation
Composite
Operation
Add
Remove
GetChild
children
forall g in children
g.Operation().

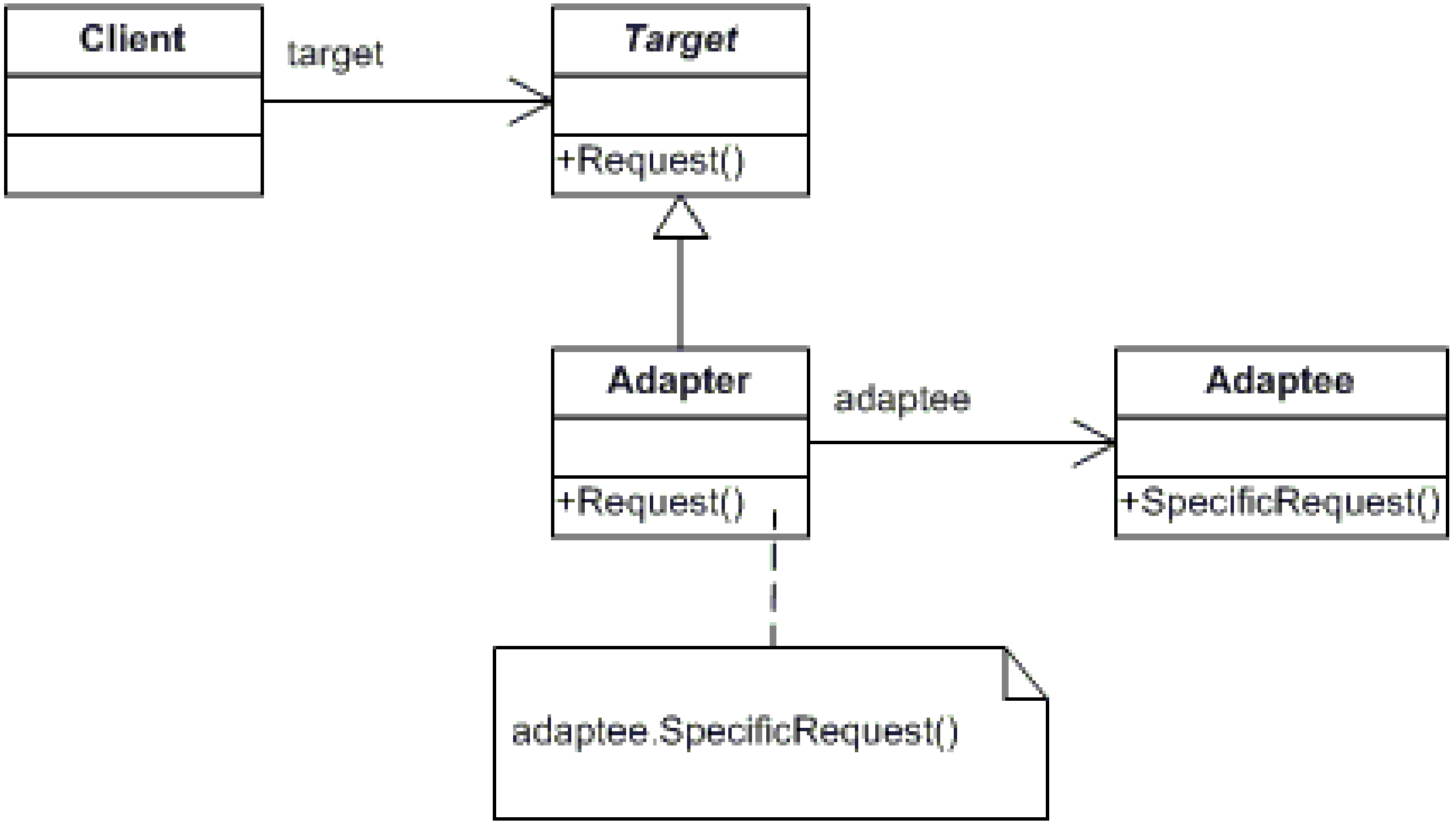
DECORATOR


Component
+Operation()
component
ConcreteComponent
+Operation()
Decorator
+Operation()
component.Operation()
ConcreteDecoratorA
-addedState
+Operation()
ConcreteDecoratorB
+Operation()
+AddedBehavior()
base.Operation();
AddedBehavior();

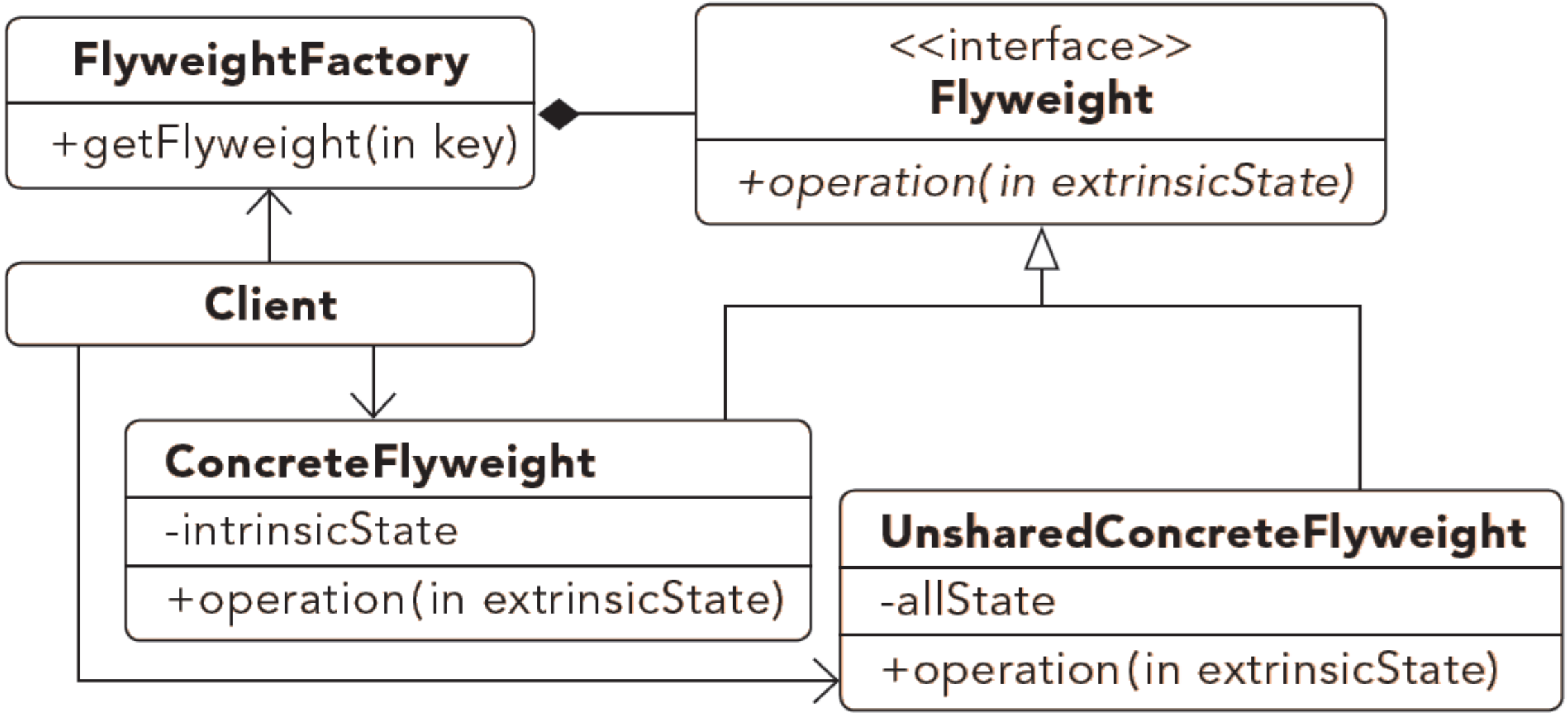
Proxy


class Proxy
Subject
+ Request()
Real Subject
+ Request()
-realSubject
Proxy
+ Request()
...
realSubject->Request();
...

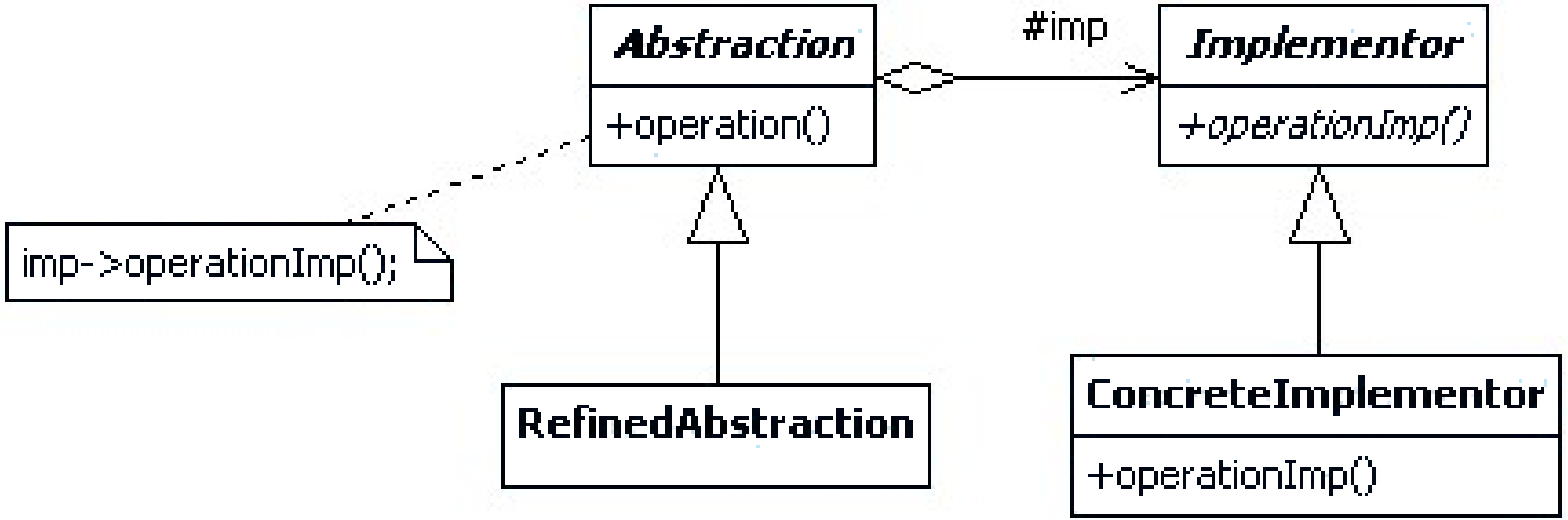
ADAPTER


Client
target
Target
+Request()
Adapter
+Request()
adaptee
Adaptee
+SpecificRequest()
adaptee.SpecificRequest()

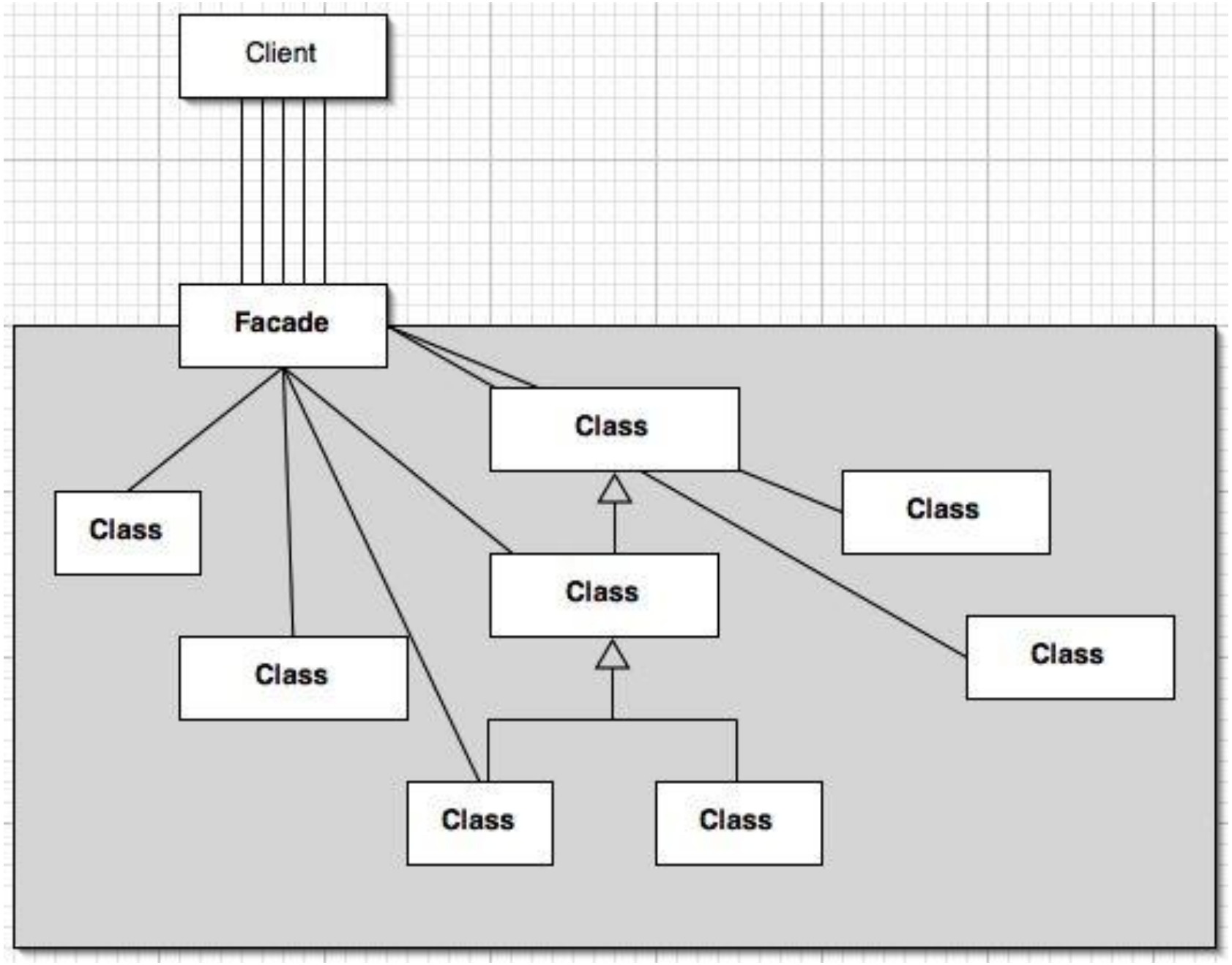
FLYWEIGHT


FlyweightFactory
+getFlyweight(in key)
<<interface>>
Flyweight
+operation(in extrinsicState)
Client
ConcreteFlyweight
-intrinsicState
+operation(in extrinsicState)
UnsharedConcreteFlyweight
-allState
+operation(in extrinsicState)

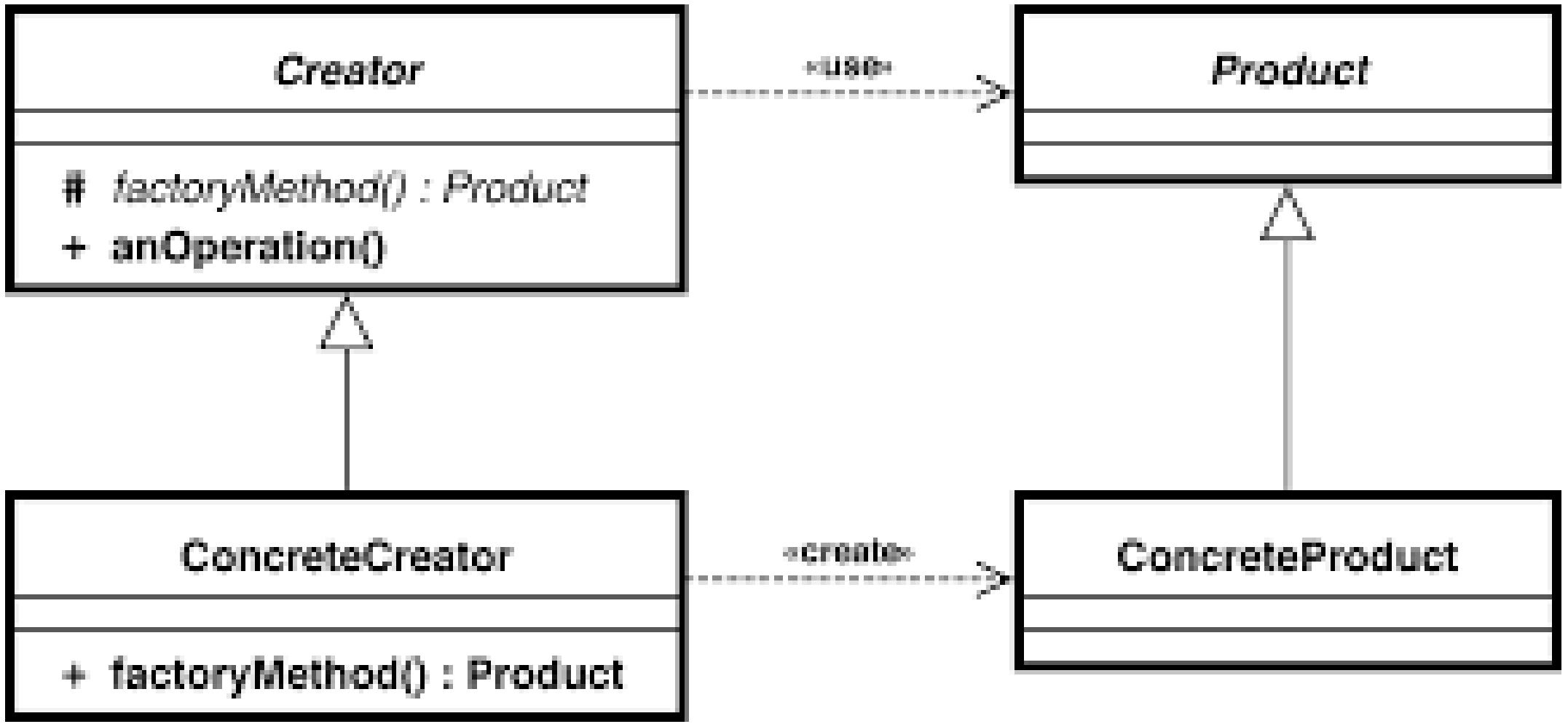
BRIDGE


Abstraction
+operation()
#imp
Implementor
+operationImp()
imp->operationImp();
RefinedAbstraction
ConcreteImplementor
+operationImp()

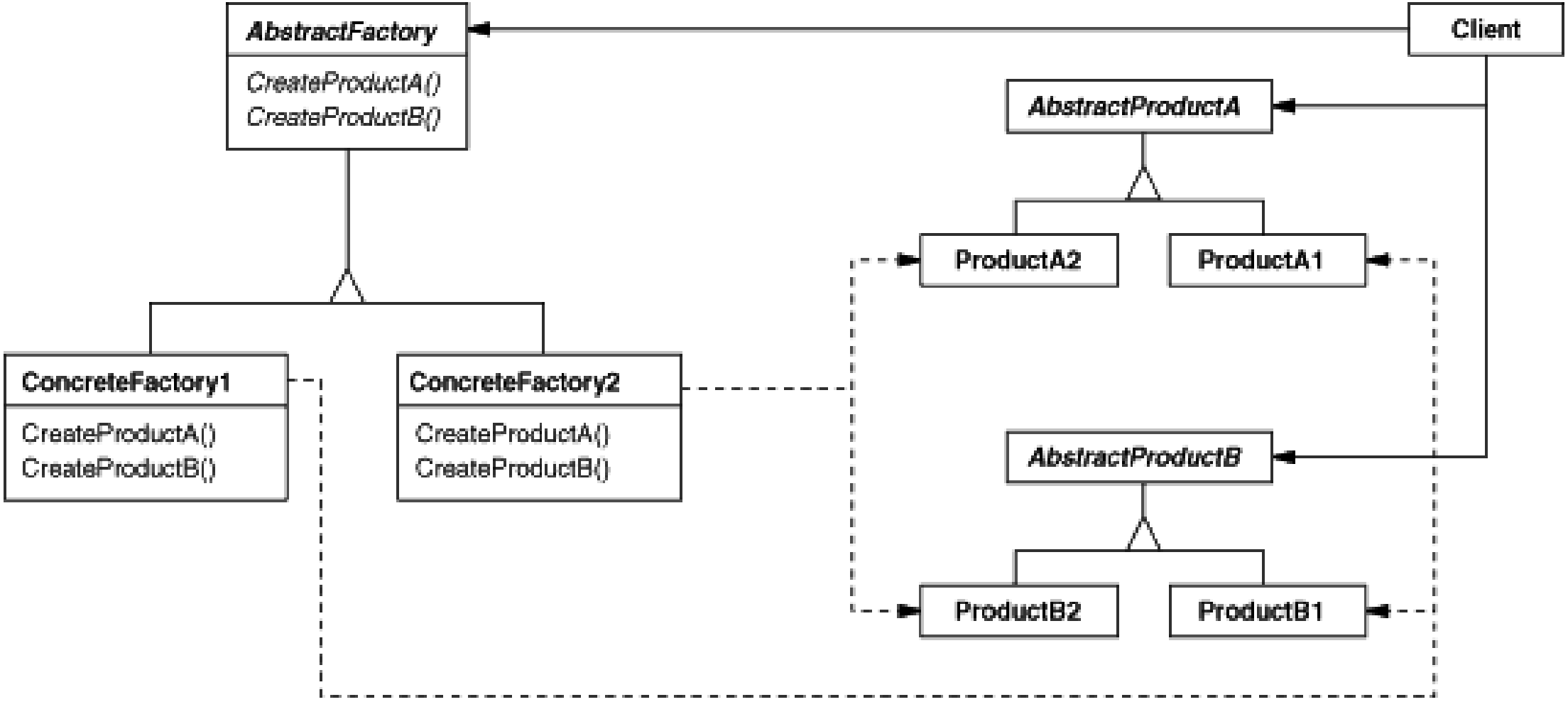
FACADE


Client
Facade
Class
Class
Class
Class
Class
Class
Class
Class

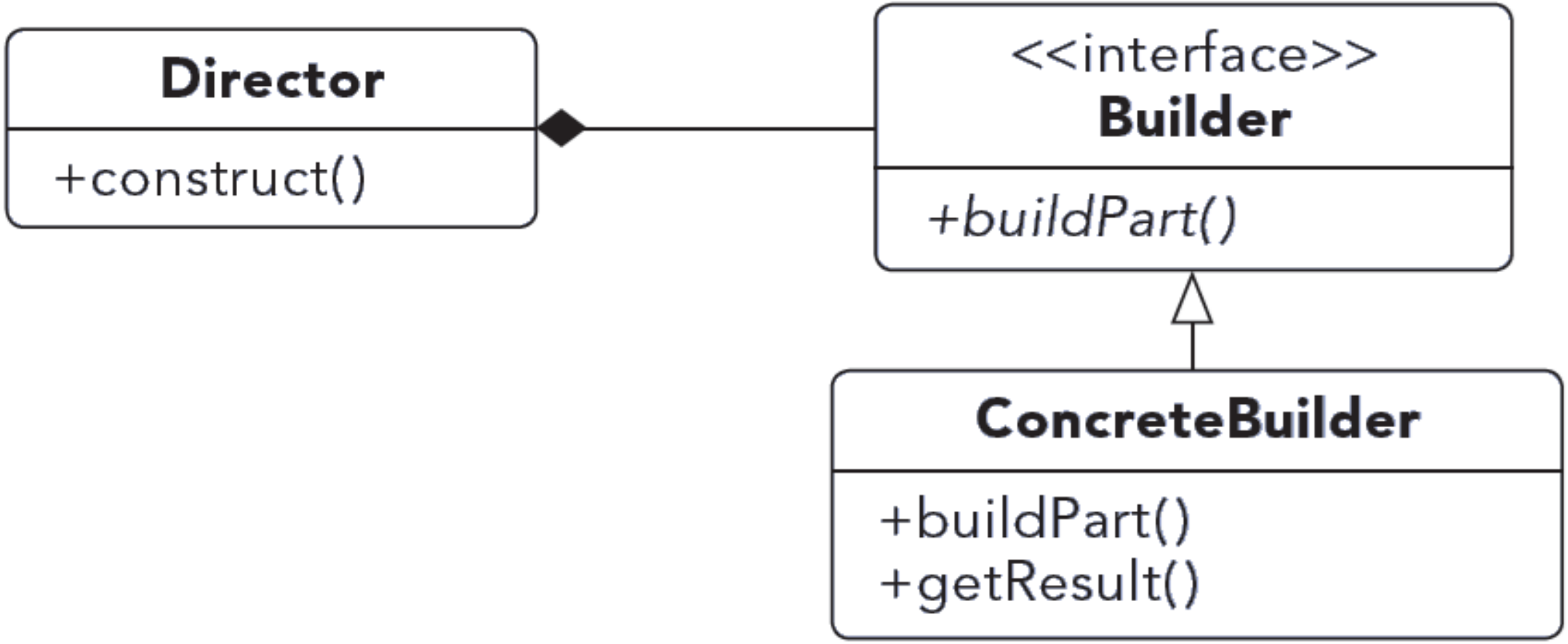
FACTORYMETHOD



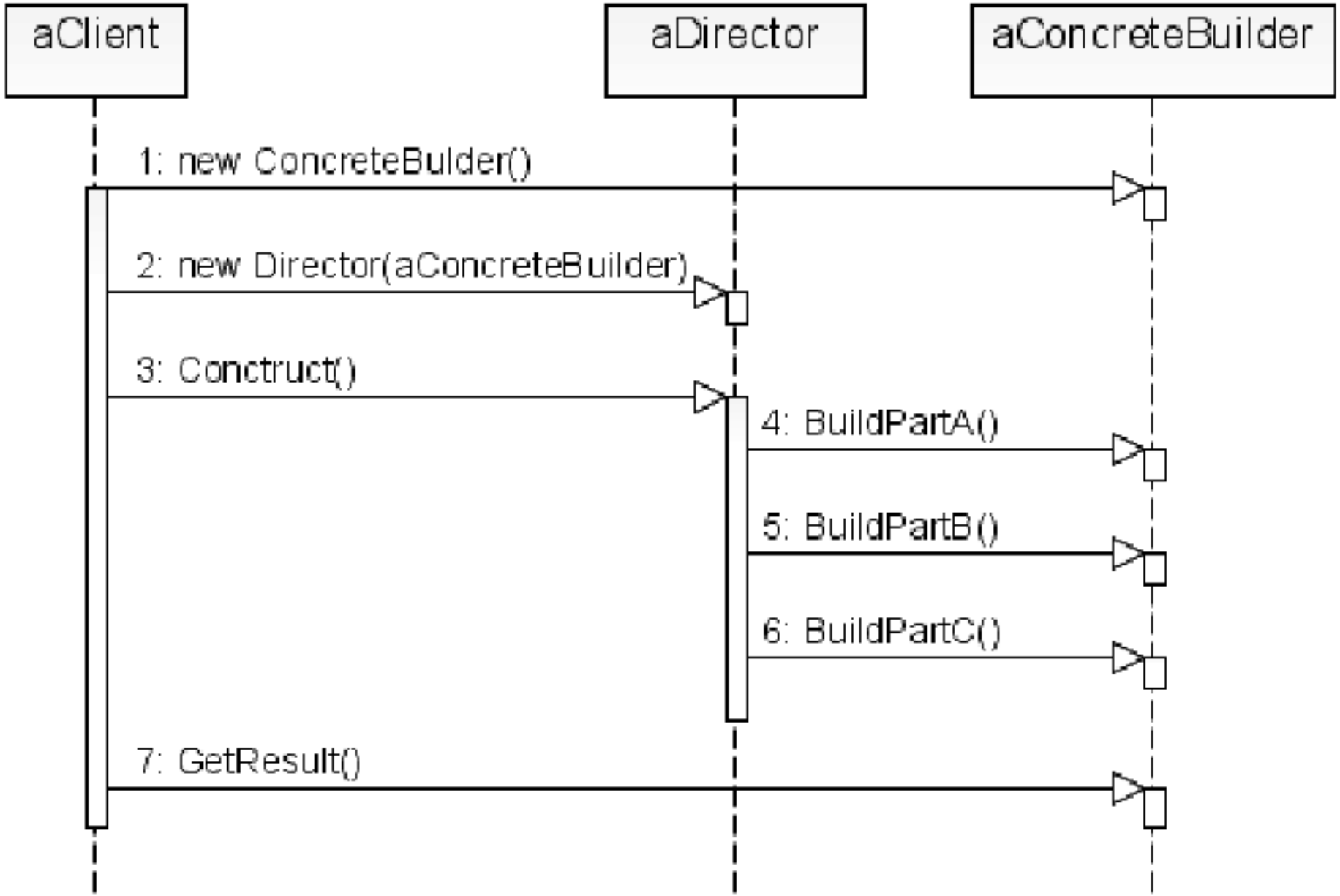
ABSTRACT FACTORY



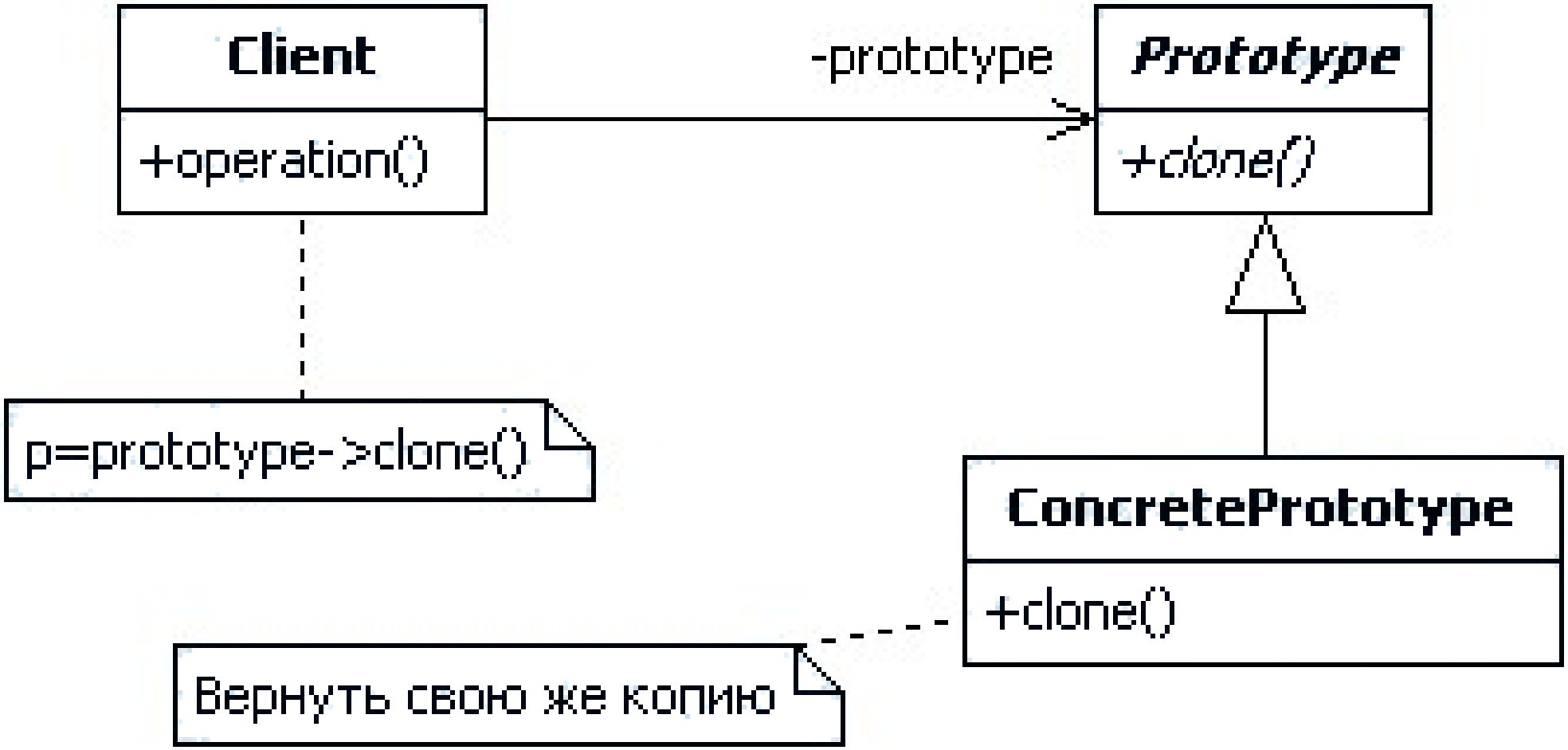
BUILDER

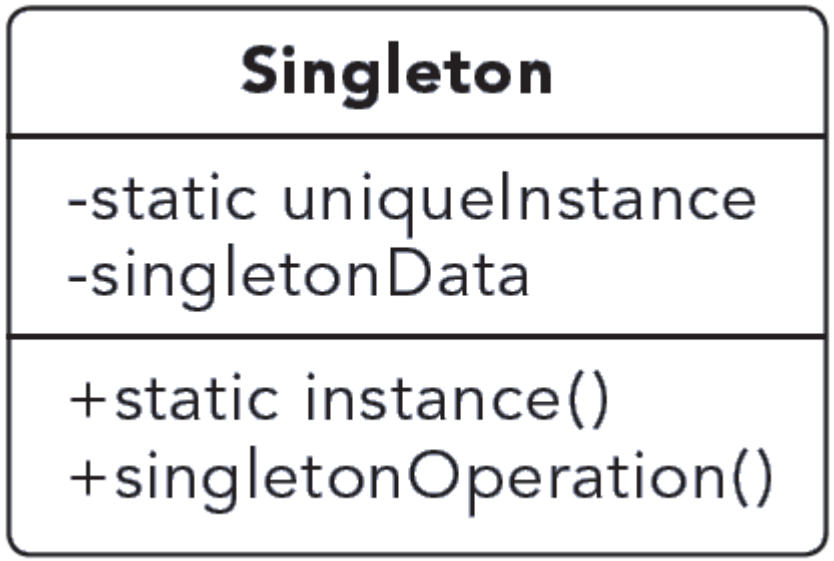

Director
+construct()
<<interface>>
Builder
+buildPart()
ConcreteBuilder
+buildPart()
+getResult()

BUILDER

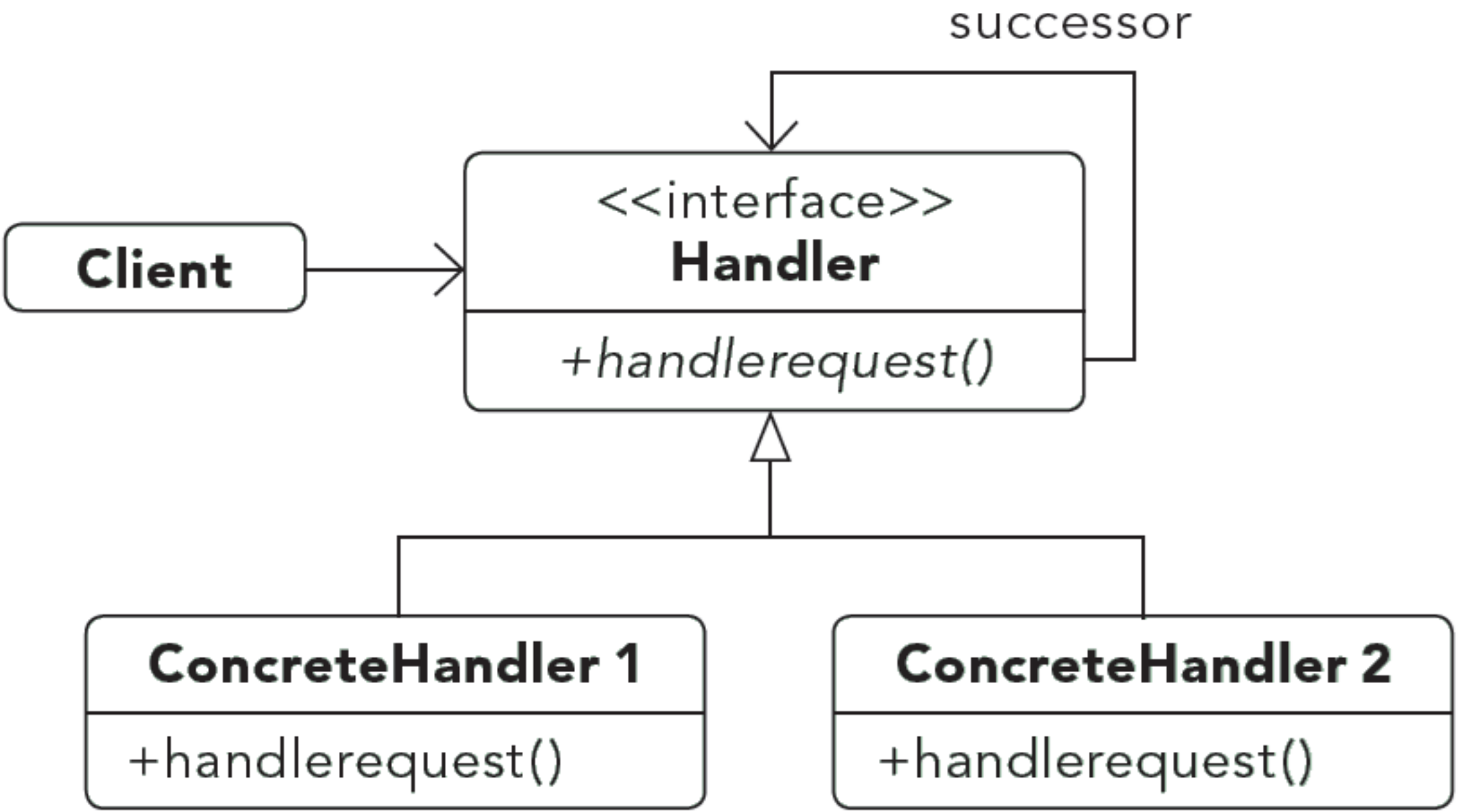

aClient
aDirector
aConcreteBuilder
1: new ConcreteBulder()
2: new Director(aConcreteBuilder)
3: Conctruct()
4: BuildPartA()
5: BuildPartB()
6: BuildPartC()
7: GetResult()

PROTOTYPE

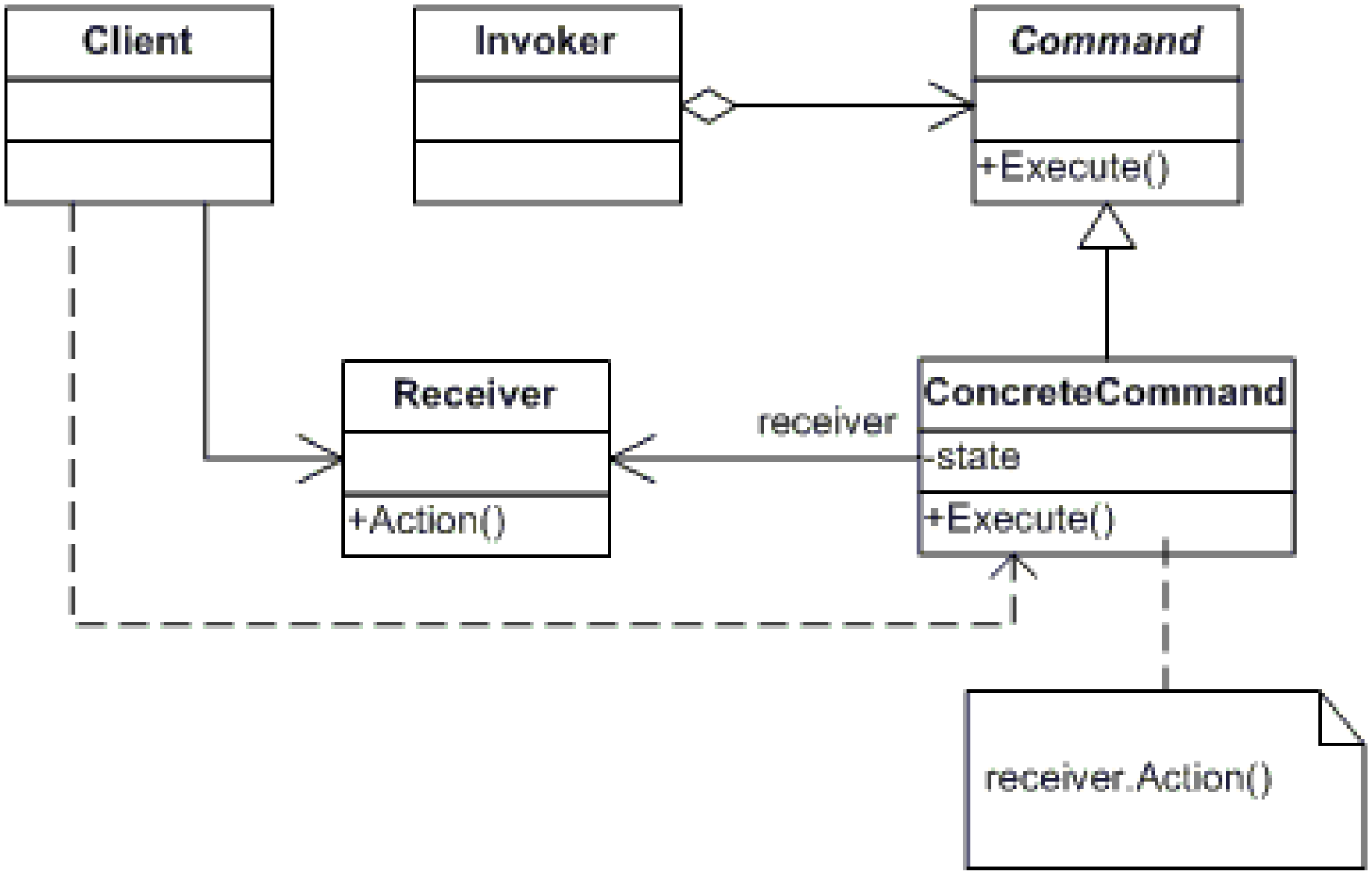

Client
+operation()
-prototype
Prototype
+clone()
p=prototype->clone()
ConcretePrototype
+clone()
Вернуть свою же копию


Singleton
-static uniqueInstance
-singletonData
+static instance()
+singletonOperation()

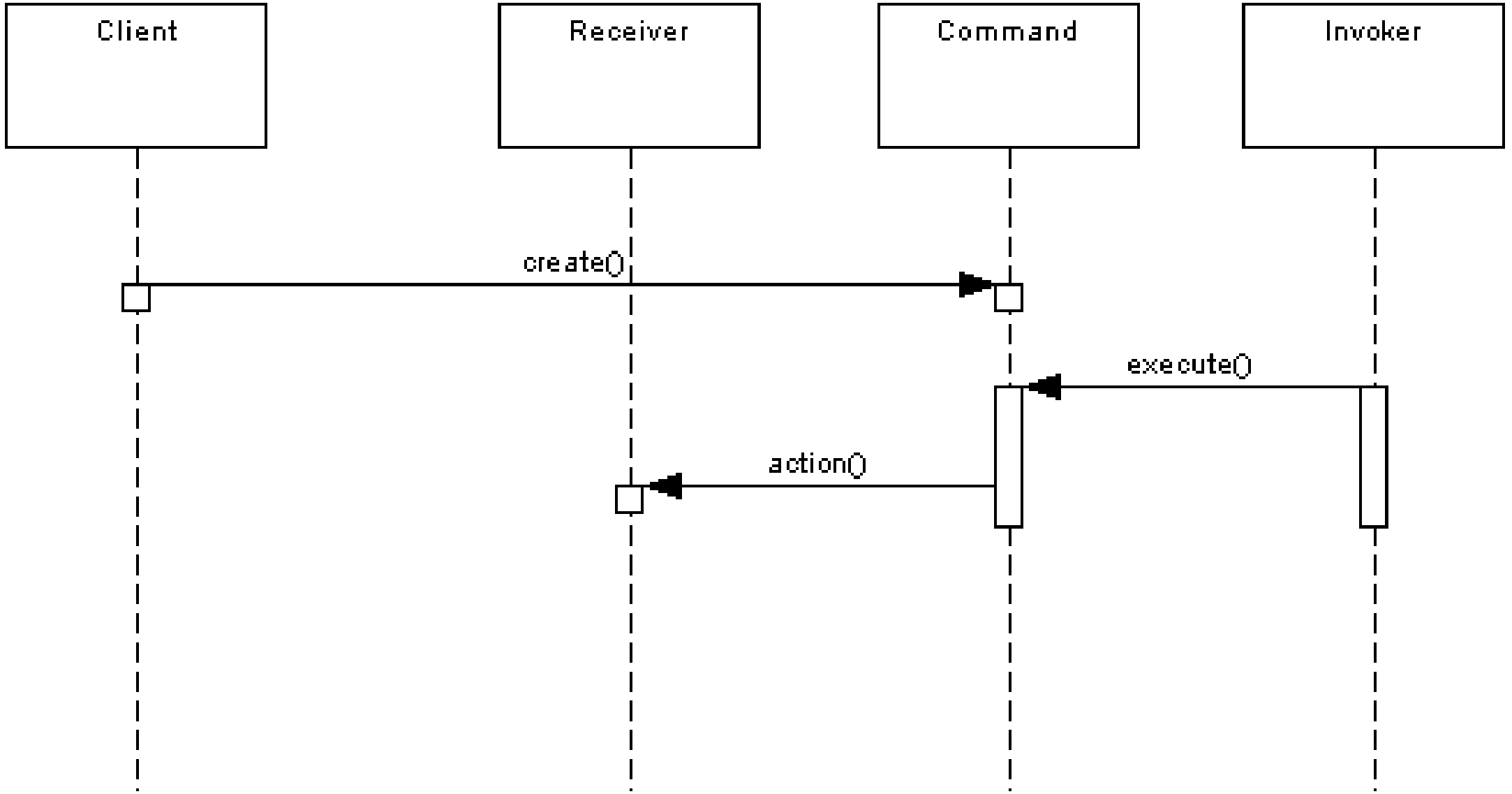
CHAIN OF RESPONSIBILITY



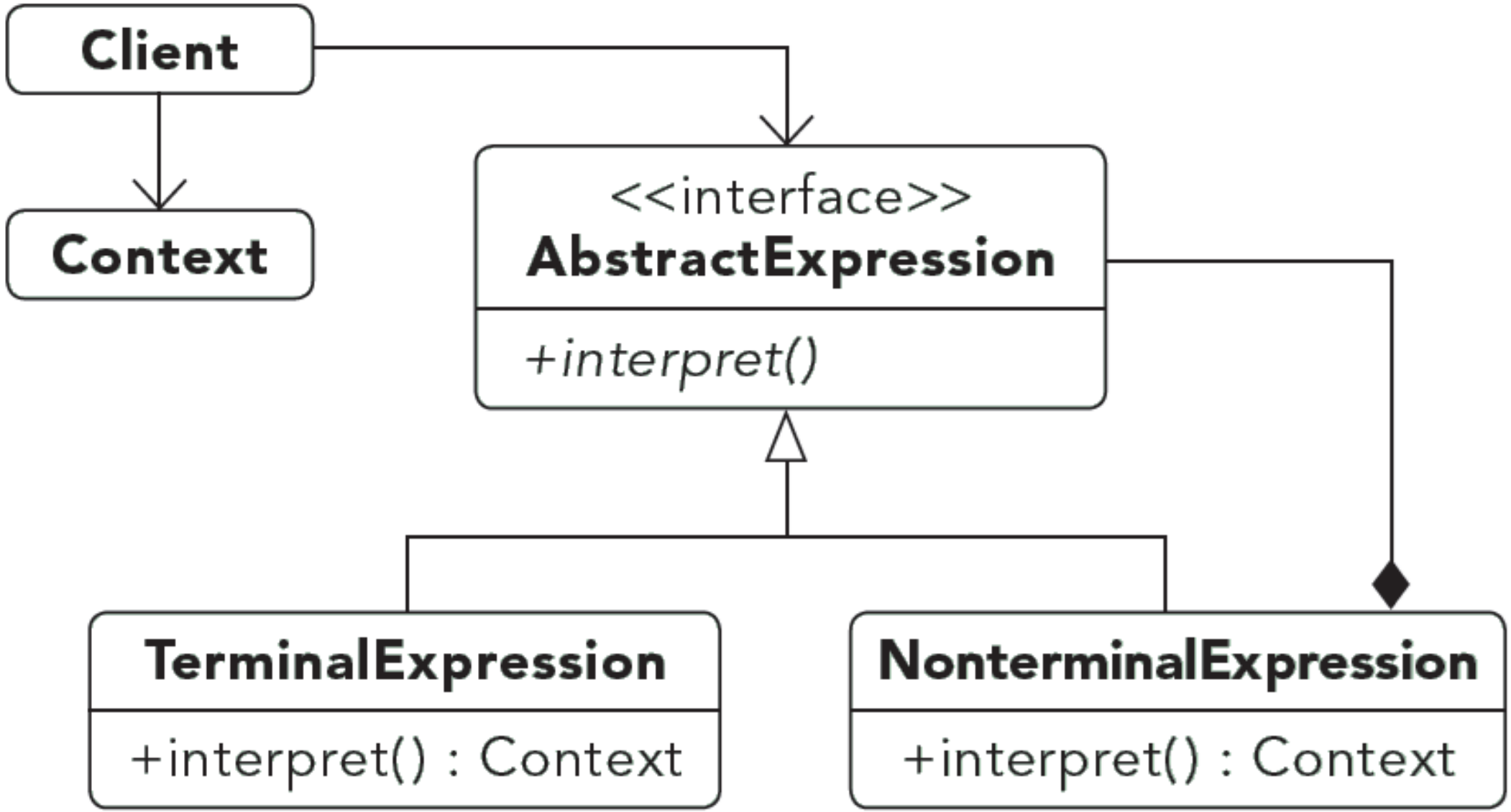
COMMAND


Client
Invoker
Command
+Execute()
Receiver
+Action()
receiver
ConcreteCommand
-state
+Execute()
receiver.Action()

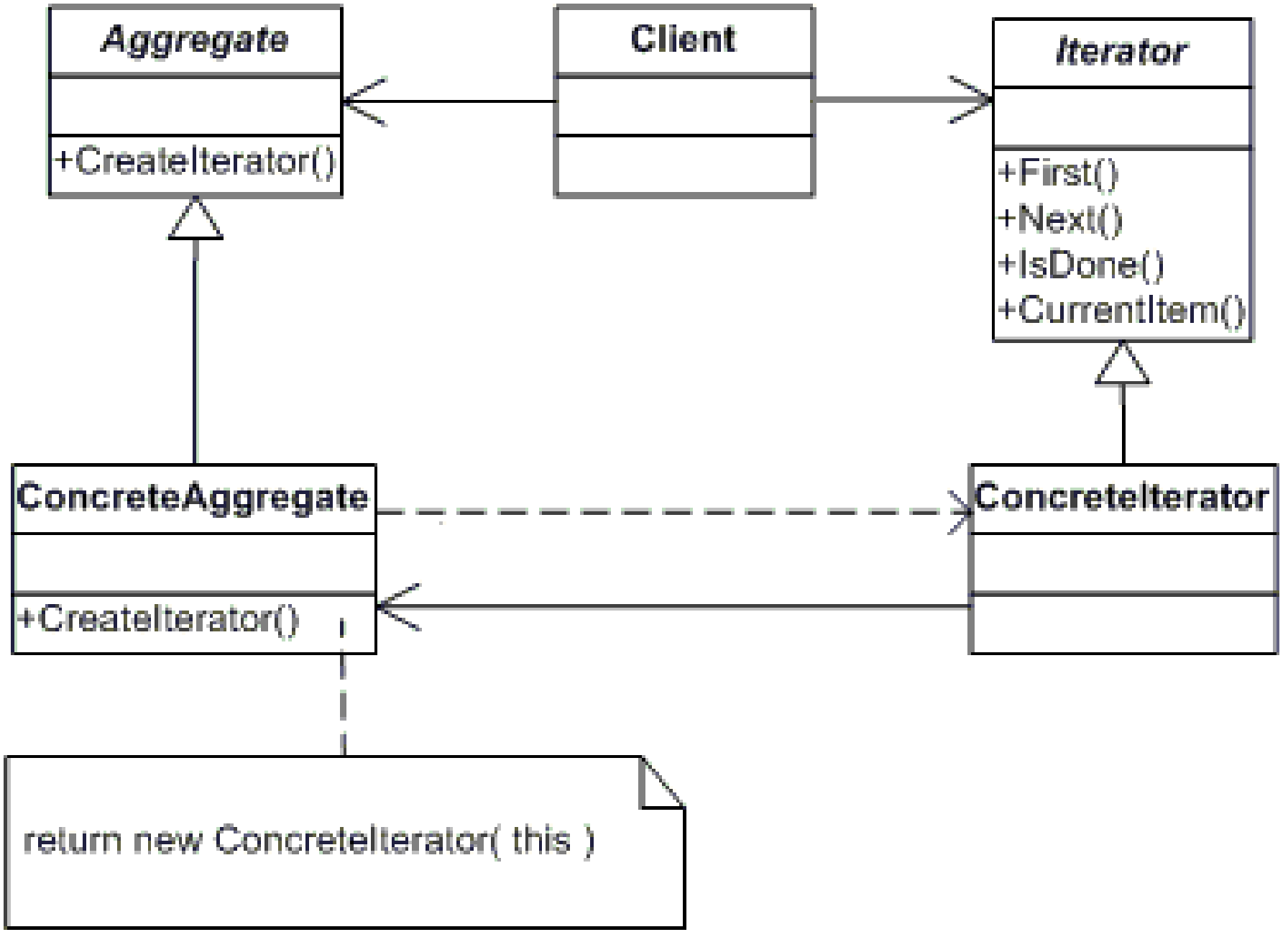
COMMAND



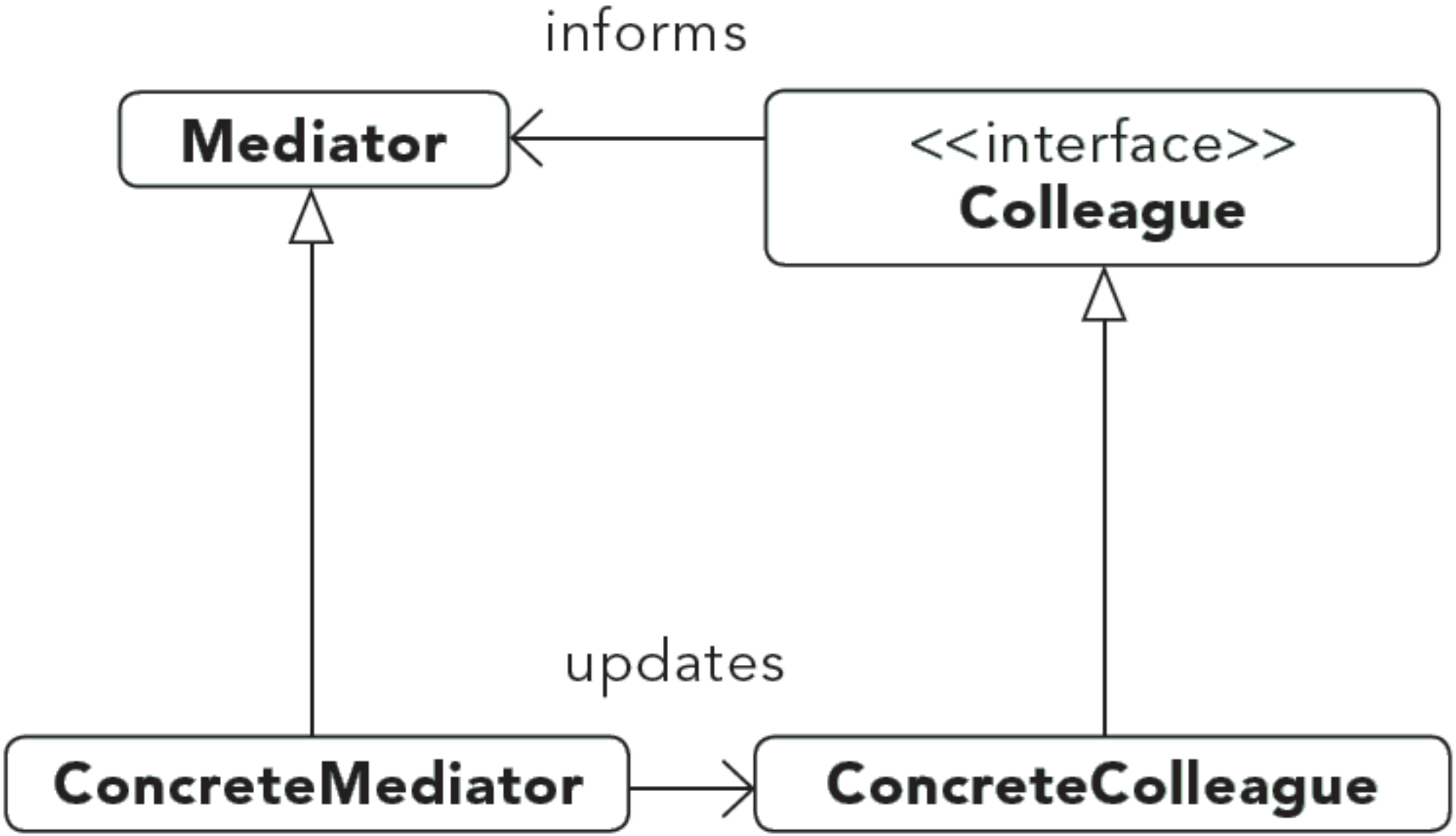
INTERPRETER



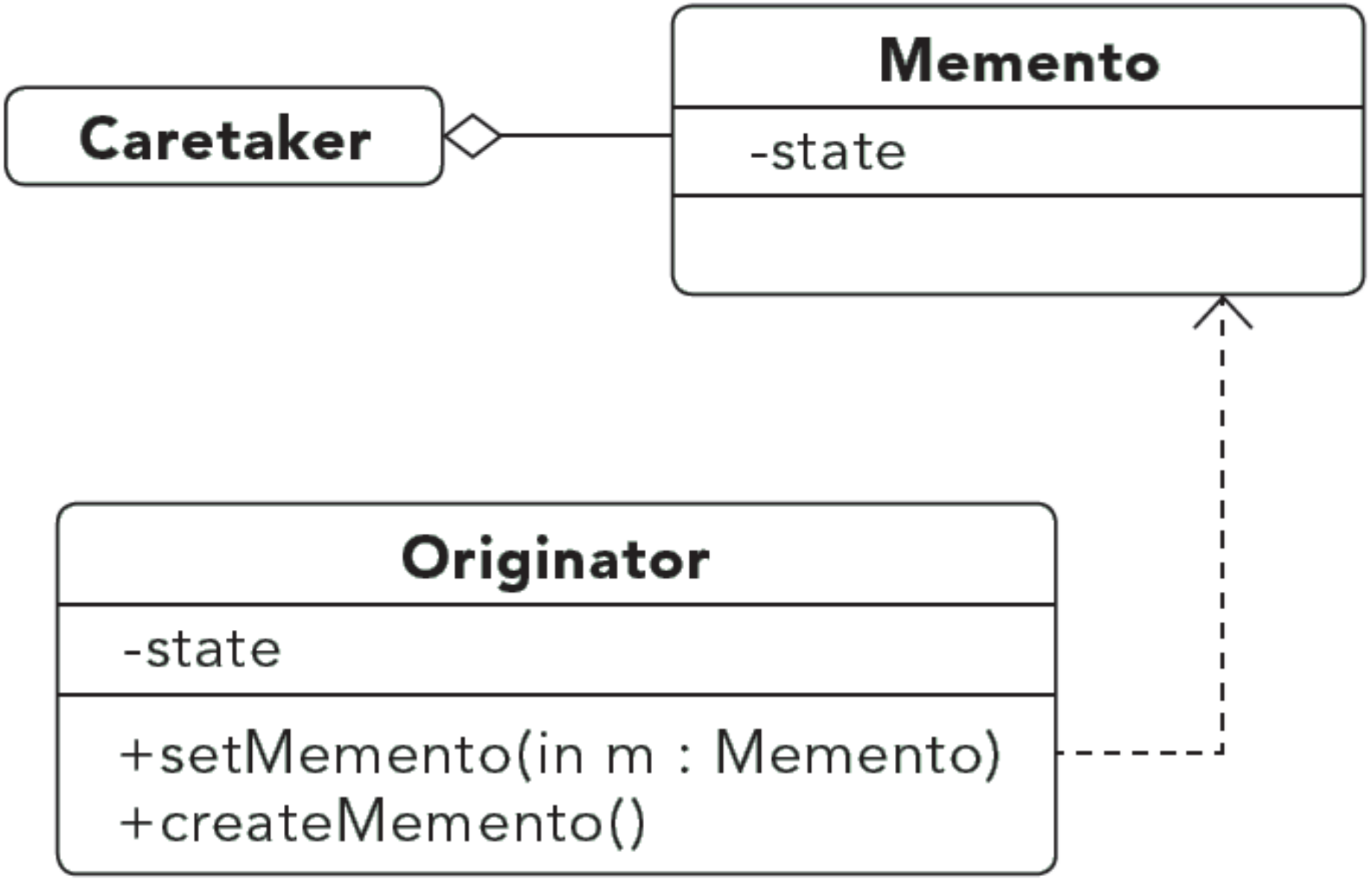
ITERATOR


Aggregate
+CreateIterator()
Client
Iterator
+First()
+Next()
+IsDone()
+CurrentItem()
ConcreteAggregate
+CreateIterator()
ConcreteIterator
return new ConcreteIterator(this)

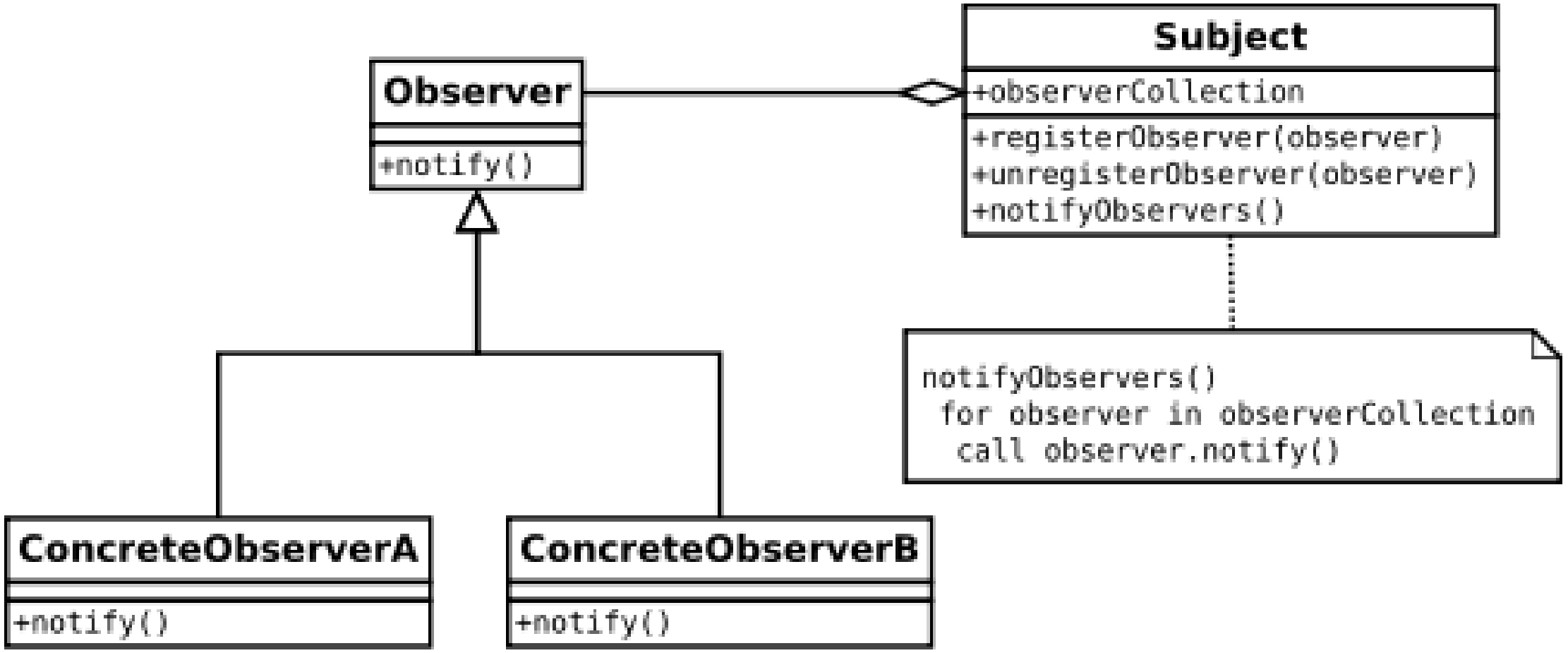
MEDIATOR


informs
Mediator
<<interface>>
Colleague
updates
ConcreteMediator
ConcreteColleague

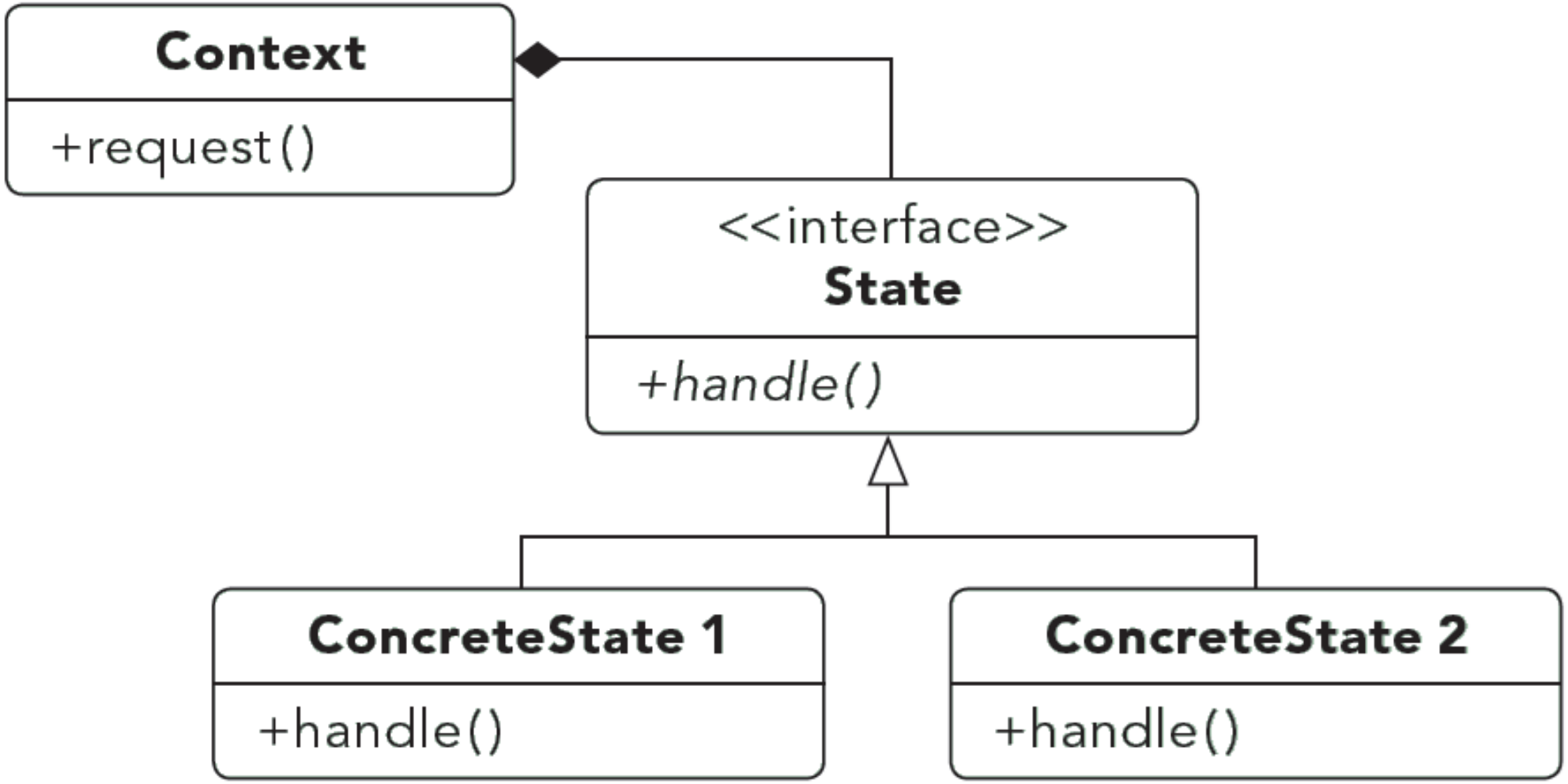
MEMENTO


Caretaker
Memento
-state
Originator
-state
+setMemento(in m : Memento)
+createMemento()

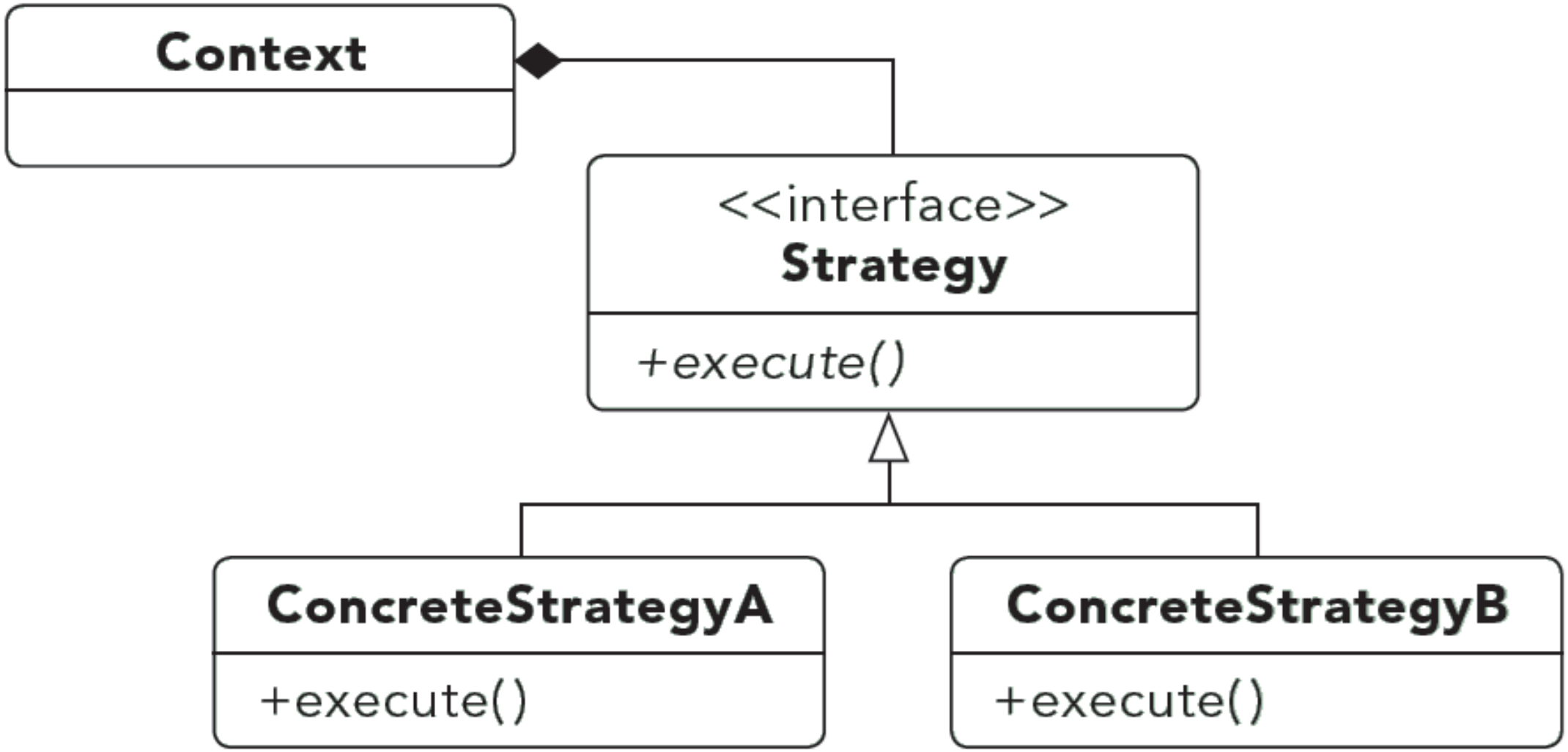
OBSERVER


Subject
+observerCollection
+registerObserver(observer)
+unregisterObserver(observer)
+notifyObservers()
Observer
+notify()
notifyObservers()
for observer in observerCollection
call observer.notify()
ConcreteObserverA
+notify()
ConcreteObserverB
+notify()

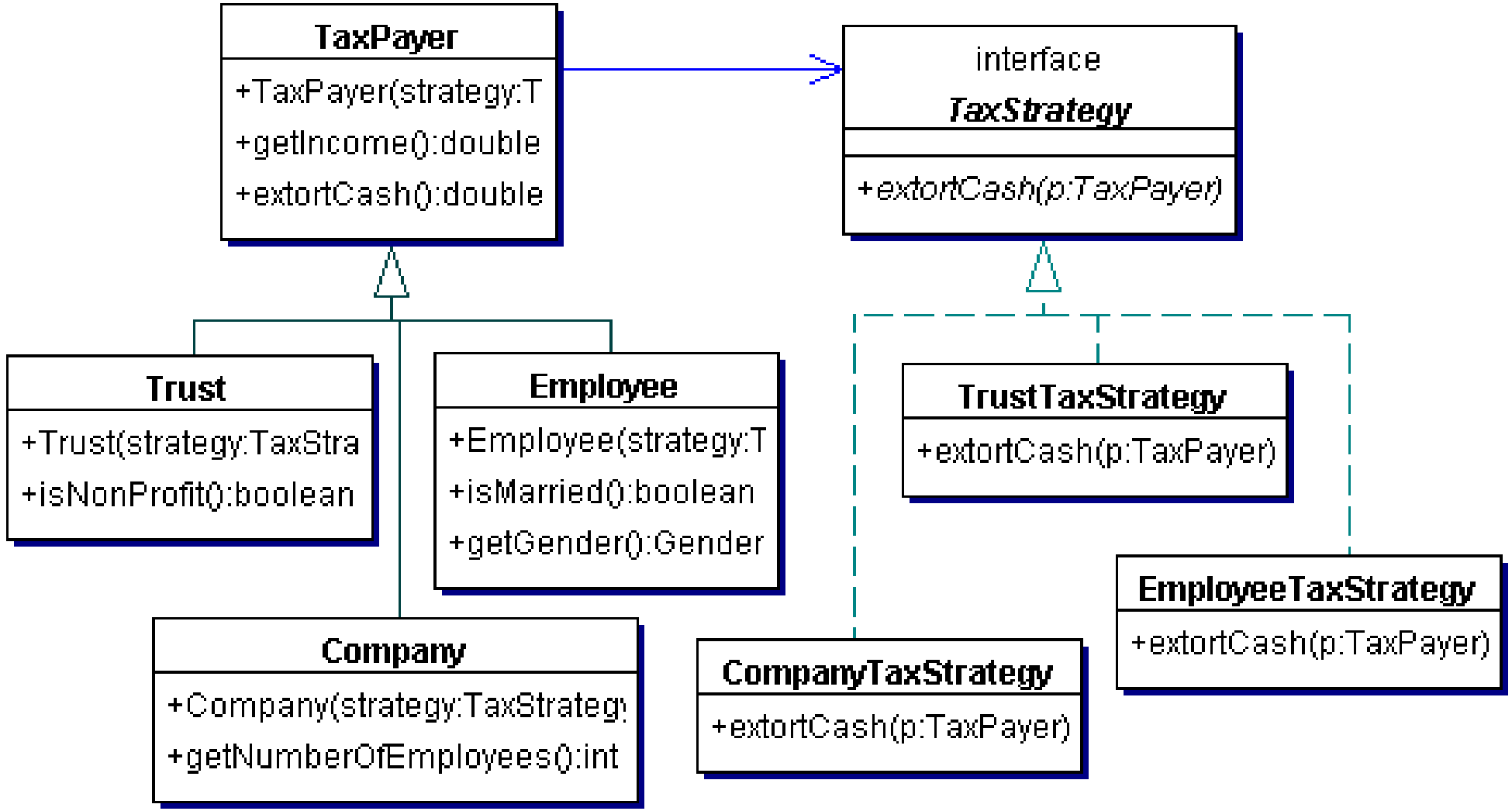
STATE


Context
+request()
<<interface>>
State
+handle()
ConcreteState 1
+handle()
ConcreteState 2
+handle()

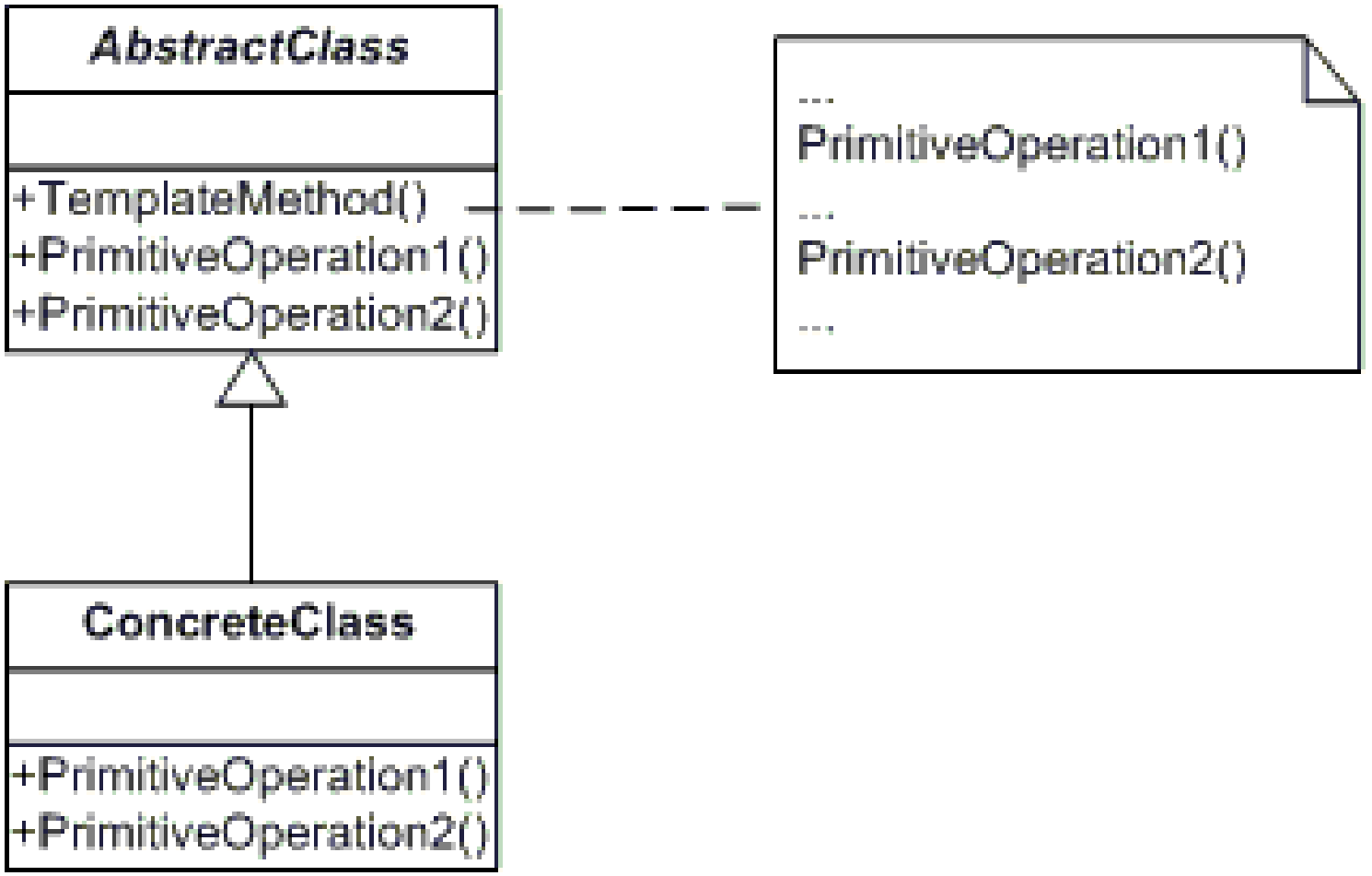
STRATEGY


Context
<<interface>>
Strategy
+execute()
ConcreteStrategyA
+execute()
ConcreteStrategyB
+execute()

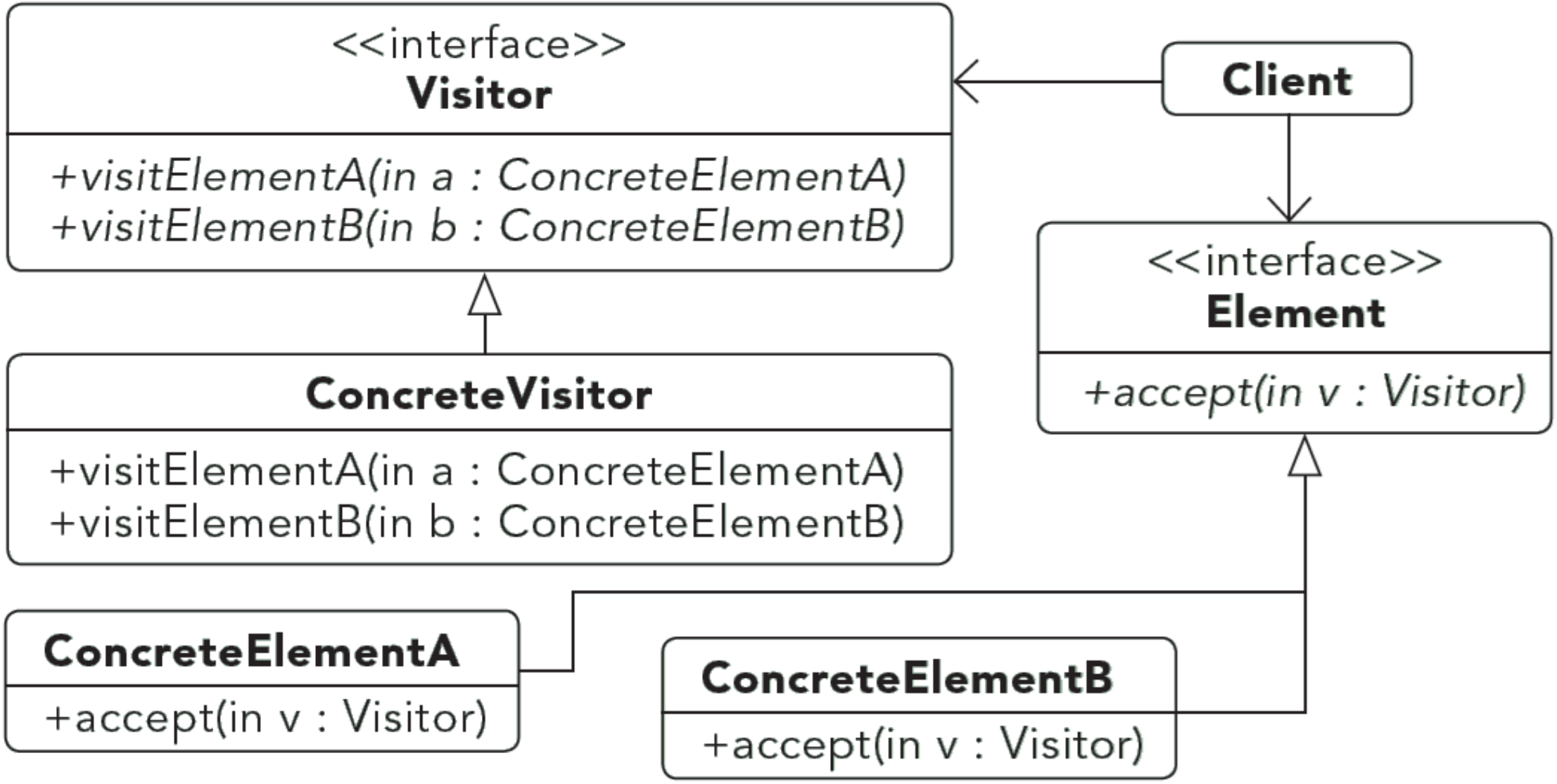
Example

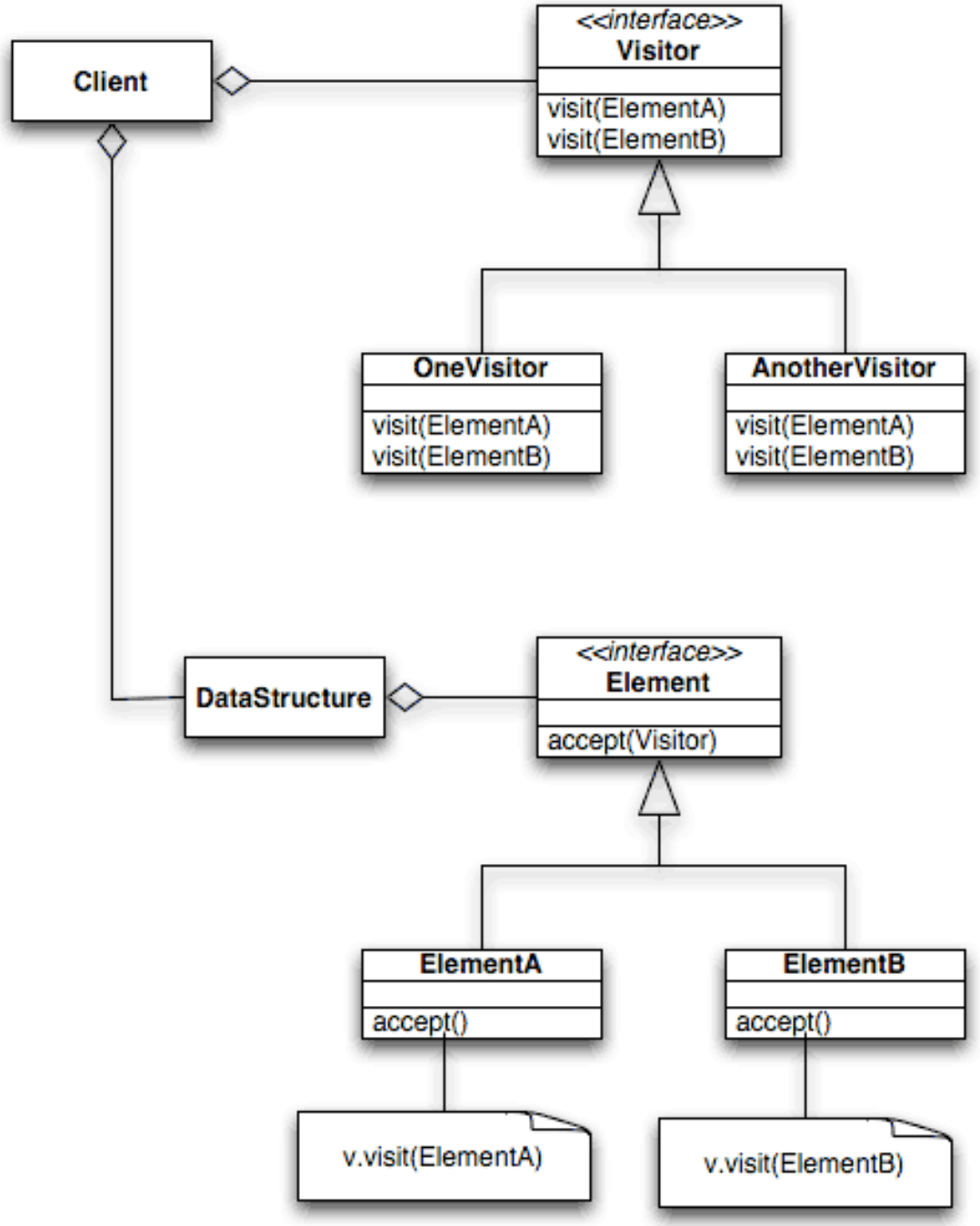

TaxPayer
+TaxPayer(strategy:T
+getIncome():double
+extortCash():double
interface
TaxStrategy
+extortCash(p:TaxPayer)
Trust
+Trust(strategy:TaxStra
+isNonProfit():boolean
Employee
+Employee(strategy:T
+isMarried():boolean
+getGender():Gender
TrustTaxStrategy
+extortCash(p:TaxPayer)
EmployeeTaxStrategy
+extortCash(p:TaxPayer)
Company
+Company(strategy:TaxStrateg
+getNumberOfEmployees():int
CompanyTaxStrategy
+extortCash(p:TaxPayer)

TEMPLATE METHOD



VISITOR


<<interface>>
Visitor
+visitElementA(in a : ConcreteElementA)
+visitElementB(in b : ConcreteElementB)
Client
<<interface>>
Element
+accept(in v : Visitor)
ConcreteVisitor
+visitElementA(in a : ConcreteElementA)
+visitElementB(in b : ConcreteElementB)
ConcreteElementA
+accept(in v : Visitor)
ConcreteElementB
+accept(in v : Visitor)

VISITOR

Example