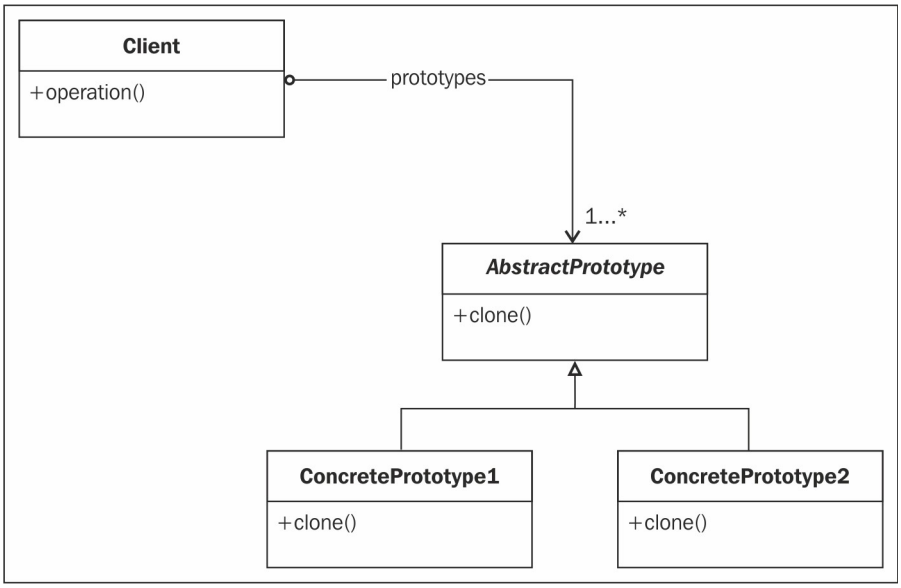
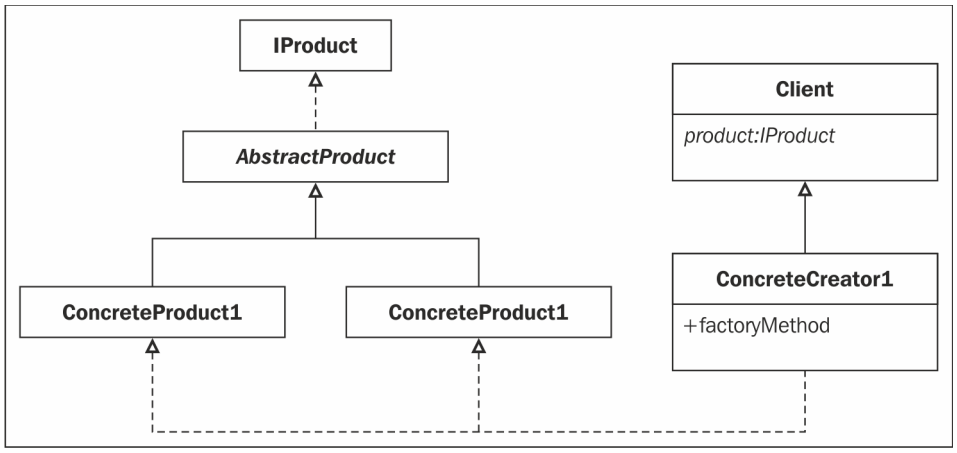
Chapter 1: Creational Patterns

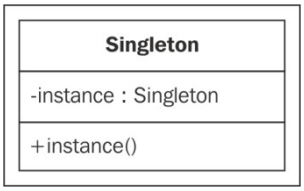


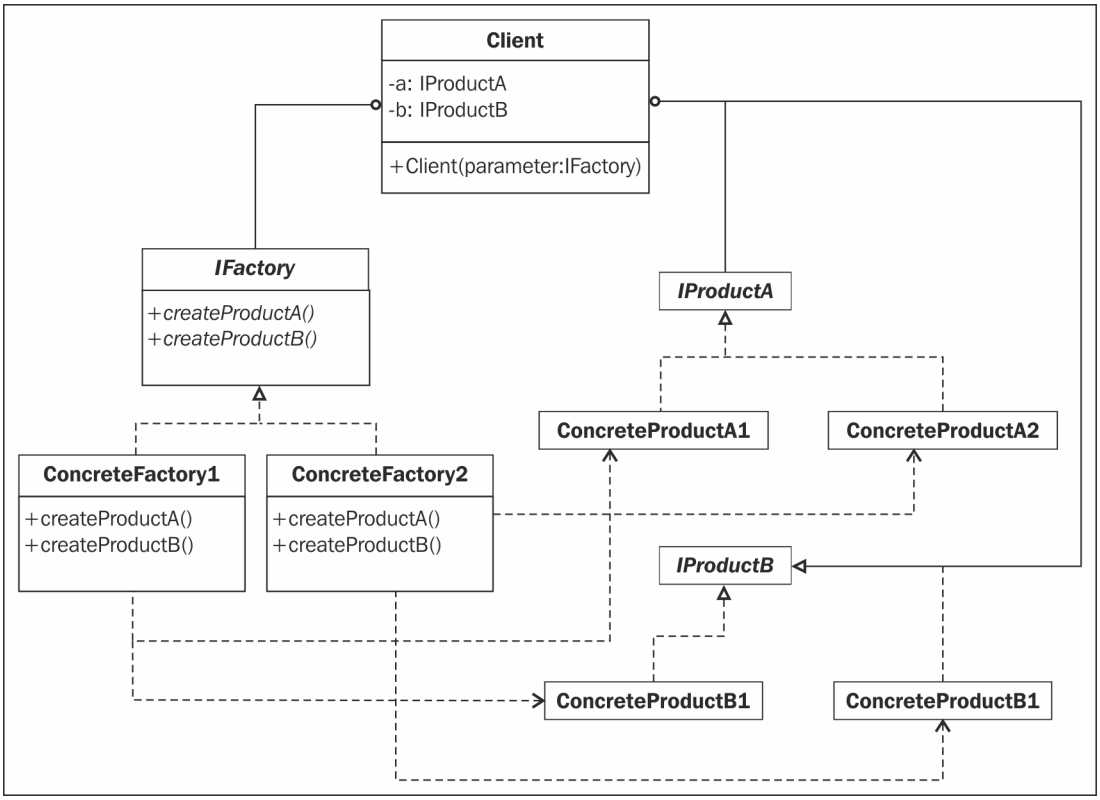

```
// now we use our faceless Manipulator card
   to clone the raidleader
let facelessManipulator = raidLeader.clone()

print("\(facelessManipulator.name,
   facelessManipulator.mana,
   facelessManipulator.attack,
   facelessManipulator.defense)")
```

```
(Optional("Raid Leader"), Optional(3),
      Optional(2), Optional(2))
```

```
AbstractCard

"(Optional("Raid Leader"), Optional(3), Optional(2), Optional(2))\n"
```

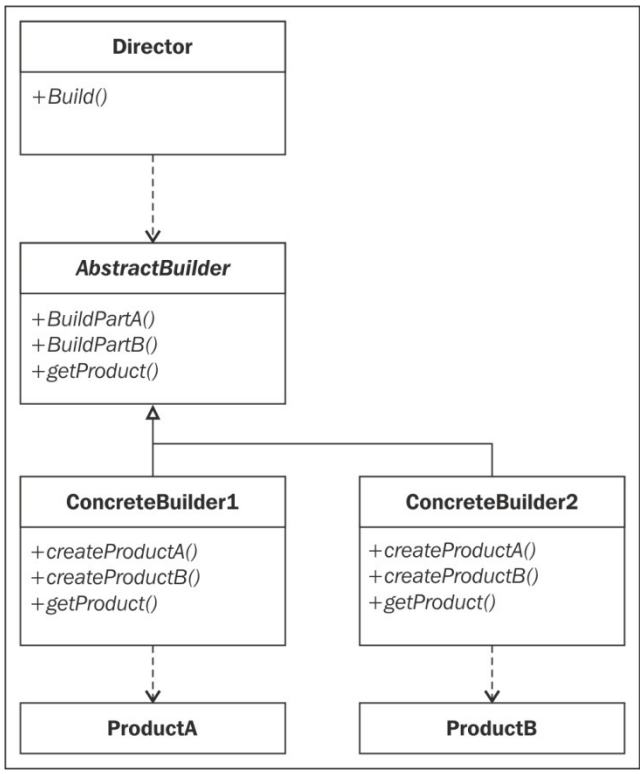

IProduct
AbstractProduct
ConcreteProduct1
ConcreteProduct1
Client
product:IProduct
ConcreteCreator1
+factoryMethod


Singleton
-instance : Singleton
+instance()


Client
-a: IProductA
-b: IProductB
+Client(parameter:IFactory)
IFactory
+createProductA()
+createProductB()
IProductA
ConcreteProductA1
ConcreteProductA2
ConcreteFactory1
+createProductA()
+createProductB()
ConcreteFactory2
+createProductA()
+createProductB()
IProductB
ConcreteProductB1
ConcreteProductB1

```
Watch38mmFactory
MilaneseBand
● r 1,0 g 1,0 b 0,0 a 1,0
"SM"
"Milanese"

GoldDial
"Gold"
"38mm"
```

```
Watch42mmFactory
LinkBracelet
● r 1,0 g 1,0 b 0,0 a 1,0
"ML"
"LinkBracelet"

GoldDial
"Gold"
"42mm"
```



Watch
r 1,0 g 1,0 b 0,0 a 1,0
"SportBand"
"ML"
"42mm"
"Aluminium"

BuilderGoldMilanese38mmWatch
Director

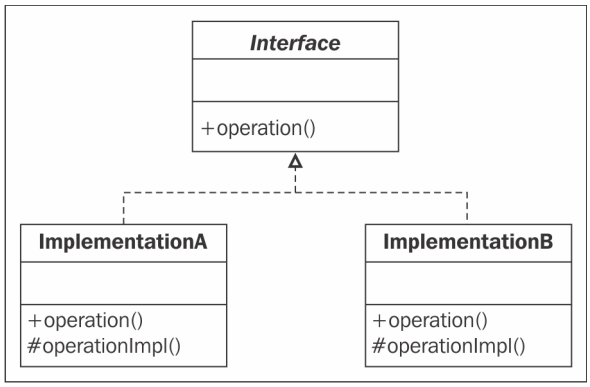
Watch
r 1,0 g 1,0 b 0,0 a 1,0
"Milanese"
"SM"
"38mm"
"Gold"

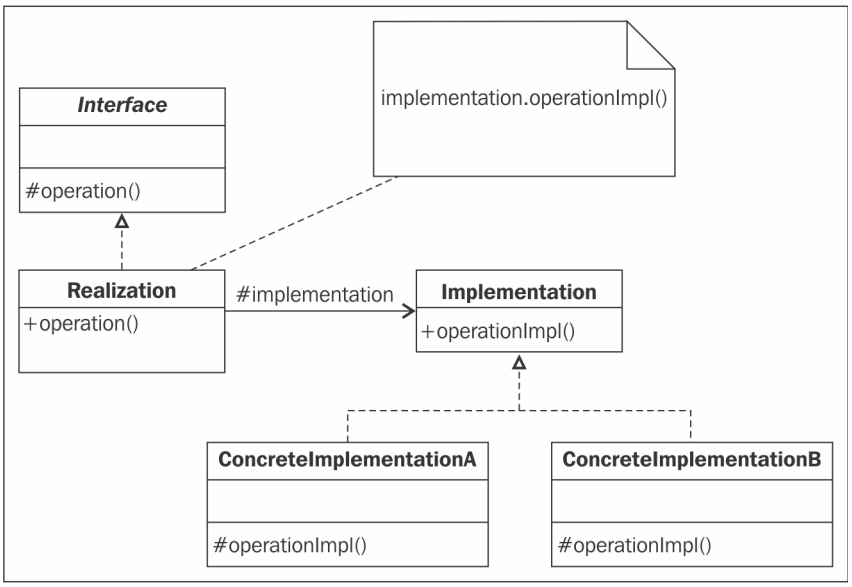
r 1,0 g 1,0 b 0,0 a 1,0
"Milanese"
"ML"
"Gold"
"42mm"

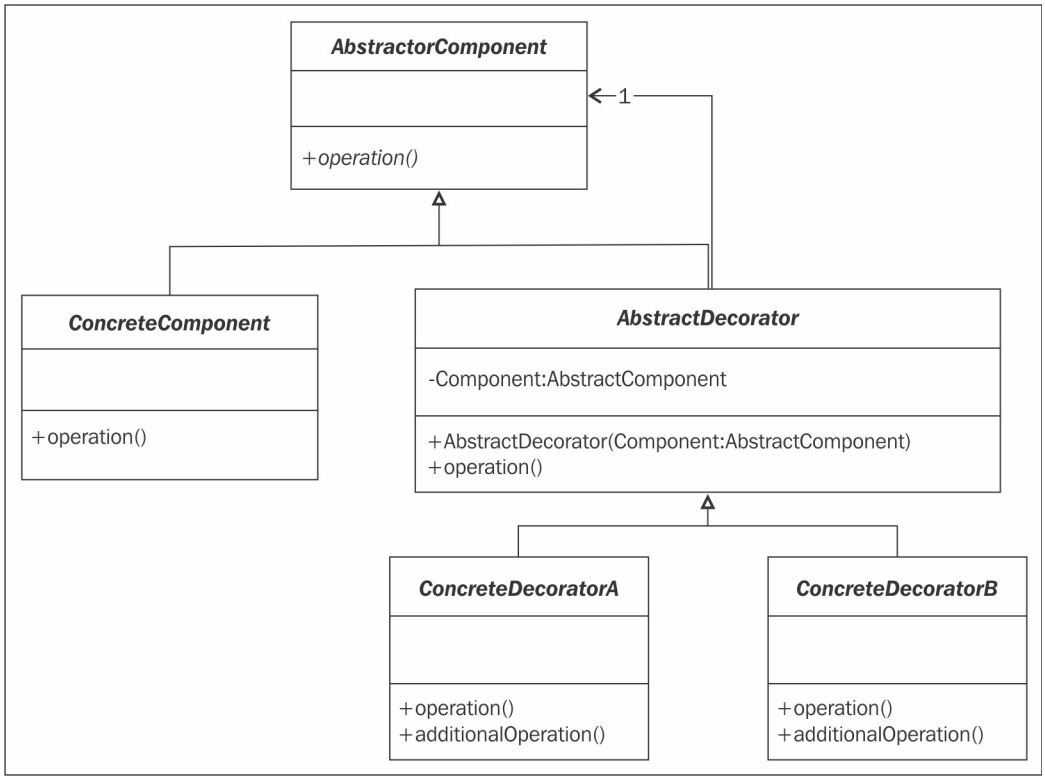
Watch
Watch
Watch

r 1,0 g 1,0 b 0,0 a 1,0
"SportBand"
"SM"
"Aluminium"
"38mm"

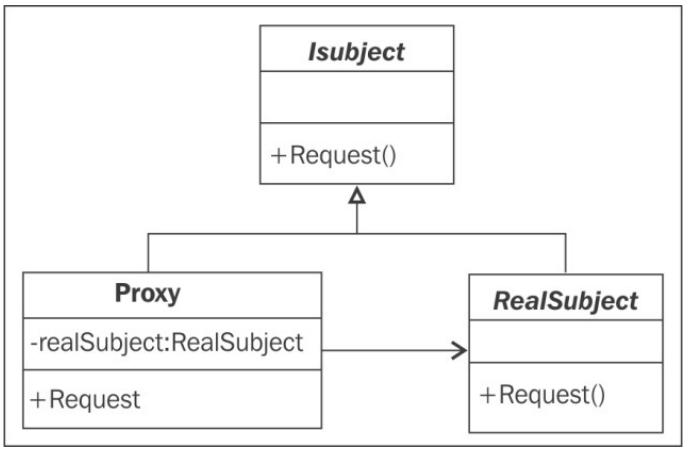
Chapter 2: Structural Patterns – Decorator, Proxy, and Bridge

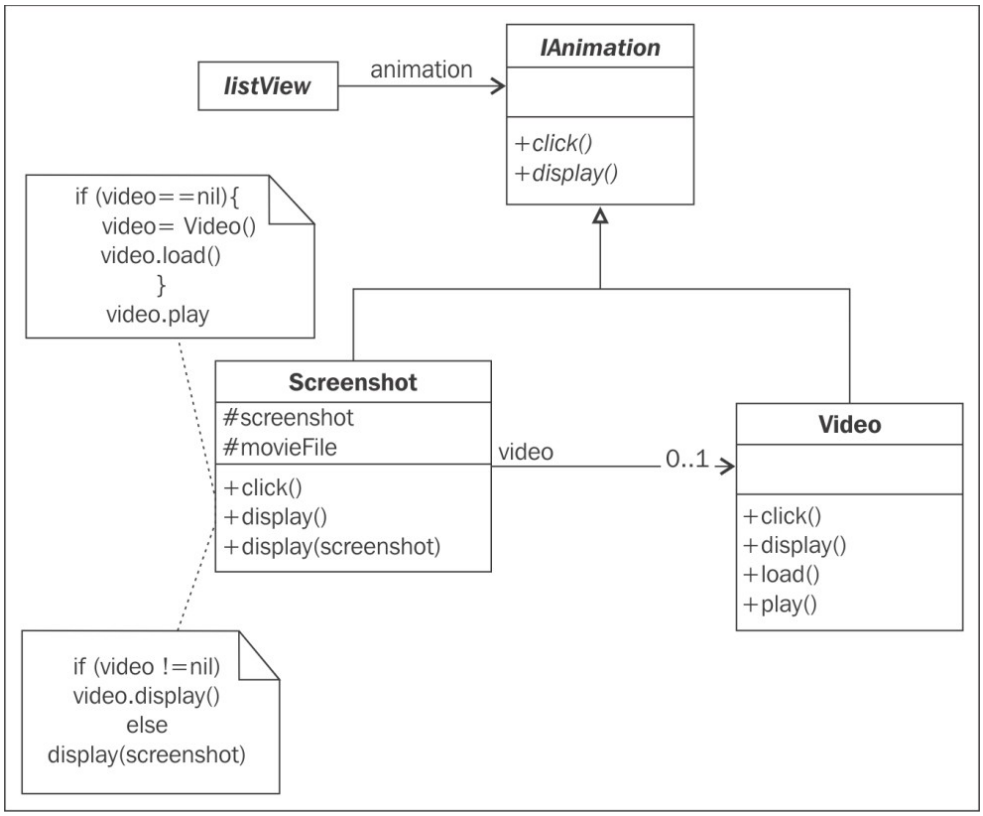




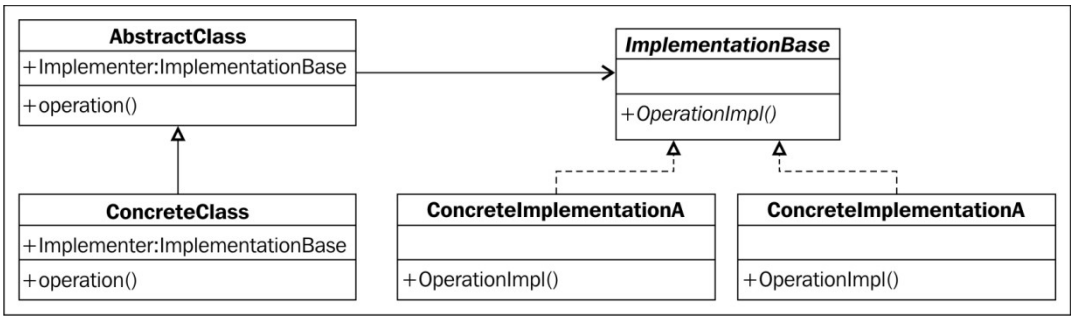



```
"rectangle with Normal Angles\n"
"drawing Shape: Rectangle"

"square with Normal Angles\n"
"drawing Shape: Square"


"Rounded border rectangle\n"
"drawing Shape: Rectangle,Corners are rounded"


"drawing Shape: Square,Corners are rounded"
```





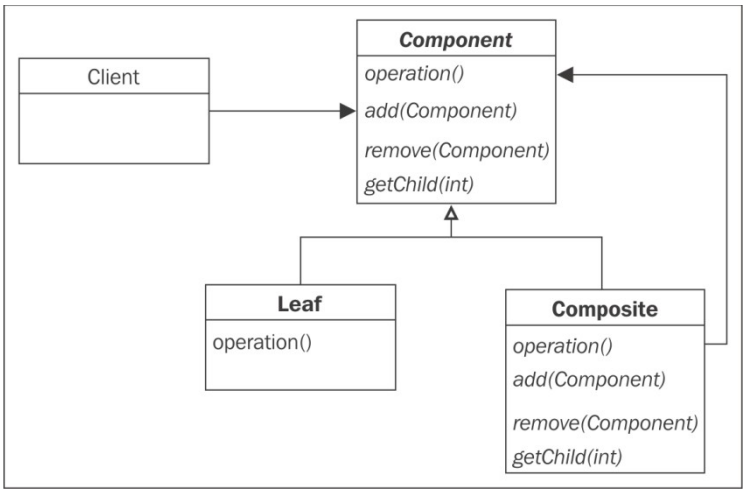
ScreenShot
"Display the screenshot of the video"
"Loading the video"
"Display the video"

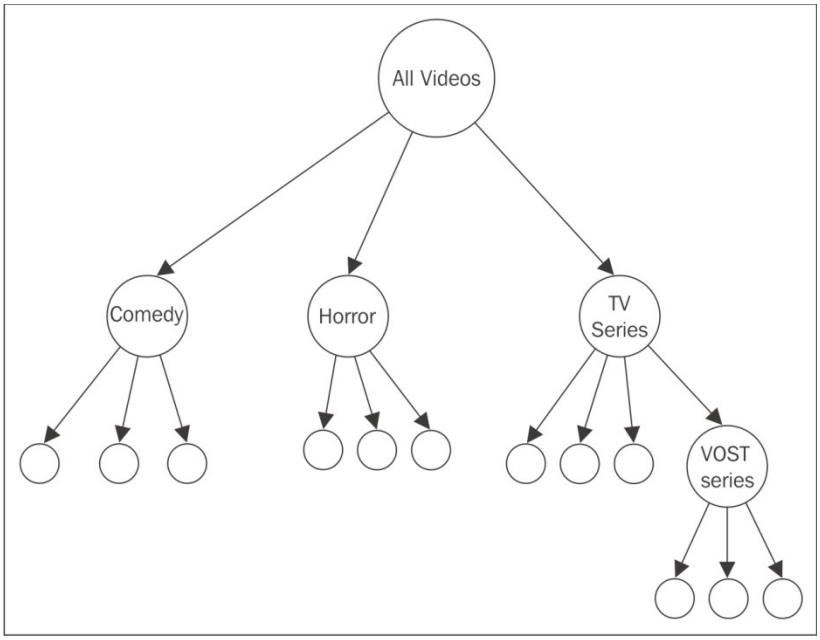


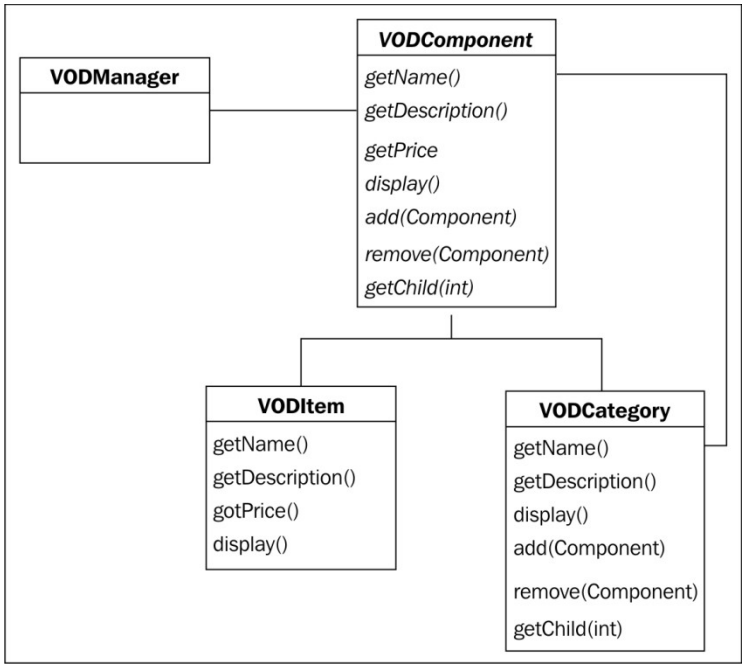
```
class TV: ImplementationBase {
  func run() {
    print("tv turned on");
  }
}

/* Implementation Classes 2 */
class Light: ImplementationBase {
  func run() {
    print("light turned on")
  }
}

let tvRemoteControl = RemoteControl(impl: TV())
tvRemoteControl.turnOn()

let lightRemoteControl = RemoteControl(impl: Light())
lightRemoteControl.turnOn()
```

```
"tv turned on\n"

"light turned on\n"

RemoteControl
RemoteControl

RemoteControl
RemoteControl
```

Chapter 3: Structural Patterns – Composite and Flyweight








```
83   override func remove(vodComponent: VODComponent) {
84     vodComponents.removeAll()
```

```
M                  Void removeAll()
M                  Void removeAll(keepCapacity: Bool)
M               Element removeAtIndex(index: Int)
M  Self.Generator.Element removeFirst()
M                  Void removeFirst(n: Int)
M               Element removeLast()
M                  Void removeRange(subRange: Range<Self.Index>)
```

Remove all elements.

```
63   override func display() -> String {
64     return " \(name!), \(price!),  ----  \(description!)"
65   }
```

(12 times)

```
99   override func display() -> String {
100    var text = " \(name!),  \(description!) \r\n
         ----------------"
101    for e in vodComponents{
102        text += "\r\n\(e.display()) \r\n"
103    }
104    return text
105  }
```

(5 times)

(16 times)

(5 times)

" All VOD, All vod components \r\n ----------\r\n Hor...

All VOD, All vod components

Horror, Horror movies category

Scream, 9.99, ---- Scream movie

Paranormal Activity, 9.99, ---- Paranormal Activity movie

Blair Witch Project, 9.99, ---- Blair Witch movie

TV Series, TV Series category

Game of thrones S1E1, 1.99, ---- Game of thrones Saison 1 episode 1

Deadwood, 1.99, ---- Deadwood Saison 1 episode 1

Breaking Bad, 1.99, ---- Breaking Bad Saison 1 Episode 1

VOSTSeries, VOST TV Series sub category

Doc Martin, 1.99, ---- Doc Martin French serie Saison 1 Episode 1

Camping Paradis, 1.99, ---- Camping Paradis French serie Saison 1 Episode 1

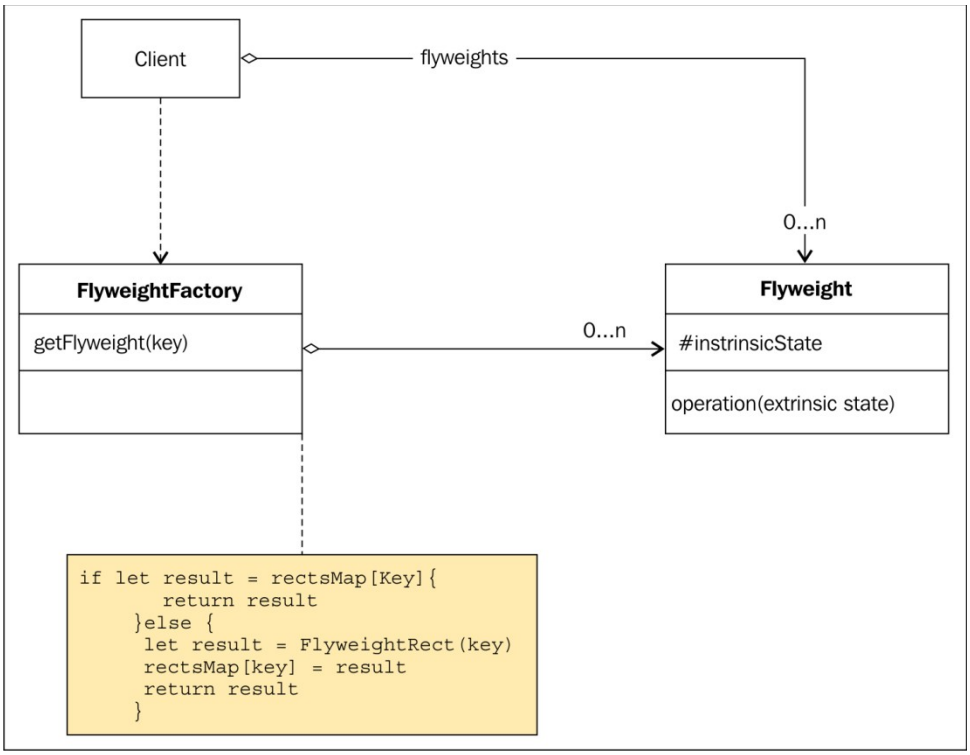
Comedy, Comedy category

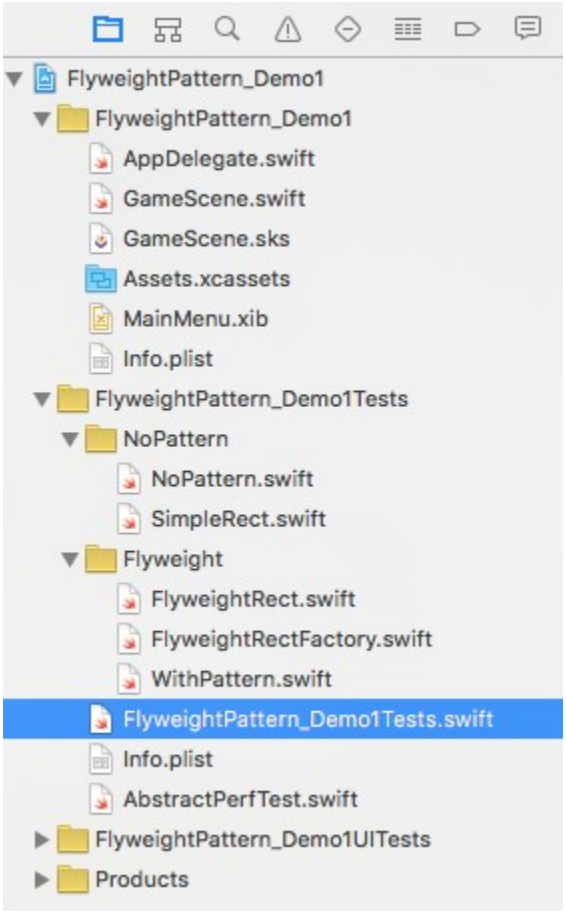
Very Bad Trip, 9.99, ---- Very Bad Trip Movie

Hot Chick, 9.99, ---- Hot Chick Movie

Step Brothers, 9.99, ---- Step Brothers Movie

Bad teacher, 9.99, ---- Bad Teacher Movie
...


Client
flyweights
0...n
FlyweightFactory
getFlyweight(key)
0...n
Flyweight
#instrinsicState
operation(extrinsic state)
if let result = rectsMap[Key]{
return result
}else {
let result = FlyweightRect(key)
rectsMap[key] = result
return result
}

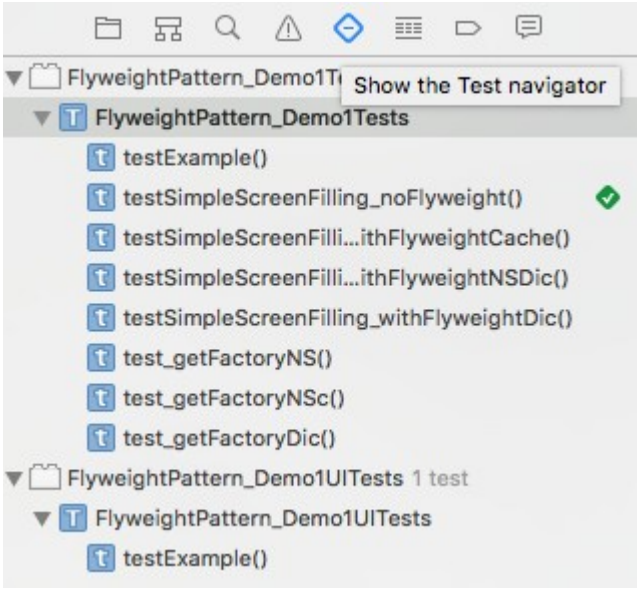

FlyweightPattern_Demo1
FlyweightPattern_Demo1
AppDelegate.swift
GameScene.swift
GameScene.sks
Assets.xcassets
MainMenu.xib
Info.plist
FlyweightPattern_Demo1Tests
NoPattern
NoPattern.swift
SimpleRect.swift
Flyweight
FlyweightRect.swift
FlyweightRectFactory.swift
WithPattern.swift
FlyweightPattern_Demo1Tests.swift
Info.plist
AbstractPerfTest.swift
FlyweightPattern_Demo1UITests
Products

```
32      //TEST without applying the pattern
33      func testSimpleScreenFilling_noFlyweight() {
34          // This is an example of a performance test case.
35          // it is executed 10 times by default to get an average
36
37          self.measureBlock() {
38            let noPattern = NoPattern()
39            noPattern.run()
40        }
41      }
42
```

```
200000 rects generated
200000 rects generated
200000 rects generated
200000 rects generated
200000 rects generated
200000 rects generated
200000 rects generated
200000 rects generated
200000 rects generated
200000 rects generated
```

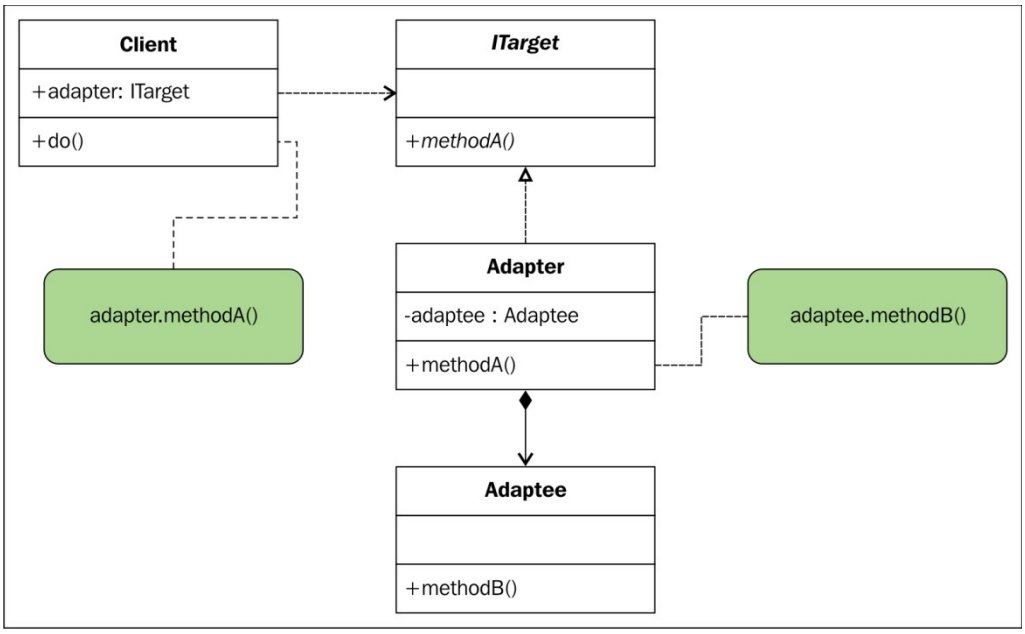
```
self.measureBlock() {
  let noPattern = NoPattern()
  noPattern.run()
}
```

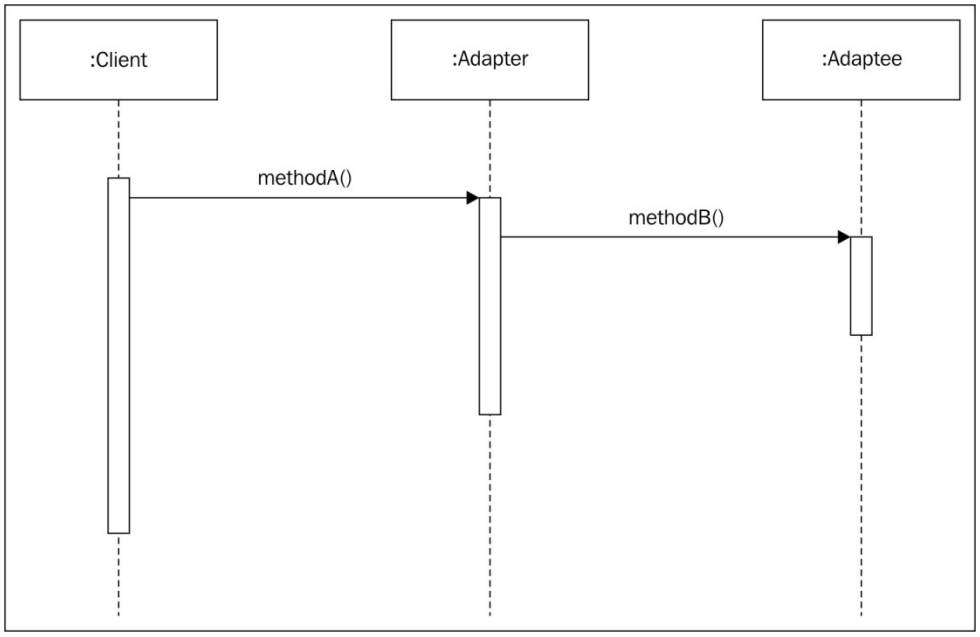
Time: 0.804 sec (5% STDEV) 2

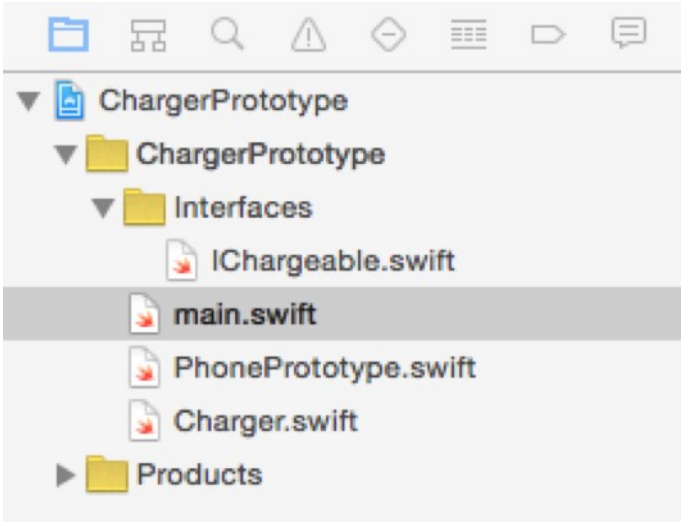


```swift
//Dictionnary
func testSimpleScreenFilling_withFlyweightDic() {
  // This is an example of a performance test case.
  // it is executed 10 times by default to get an average

  self.measureBlock() {
    let withPattern = WithPattern()
    withPattern.run()
  }                                                    Time: 0.247 sec (9% STDEV) 2
}


  //TEST without applying the pattern
  func testSimpleScreenFilling_noFlyweight() {
      // This is an example of a performance test case.
      // it is executed 10 times by default to get an average

      self.measureBlock() {
        let noPattern = NoPattern()
        noPattern.run()
    }                                                  Time: 0.877 sec (6% STDEV) 2
  }
```

Chapter 4: Structural Patterns – Adapter and Facade








```
18      self.phone.charge(volts)
19  }
```

Declaration `func charge(volts: Double)`

Description This function is called to charge a mobile phone

Usage:
charge(5.5)

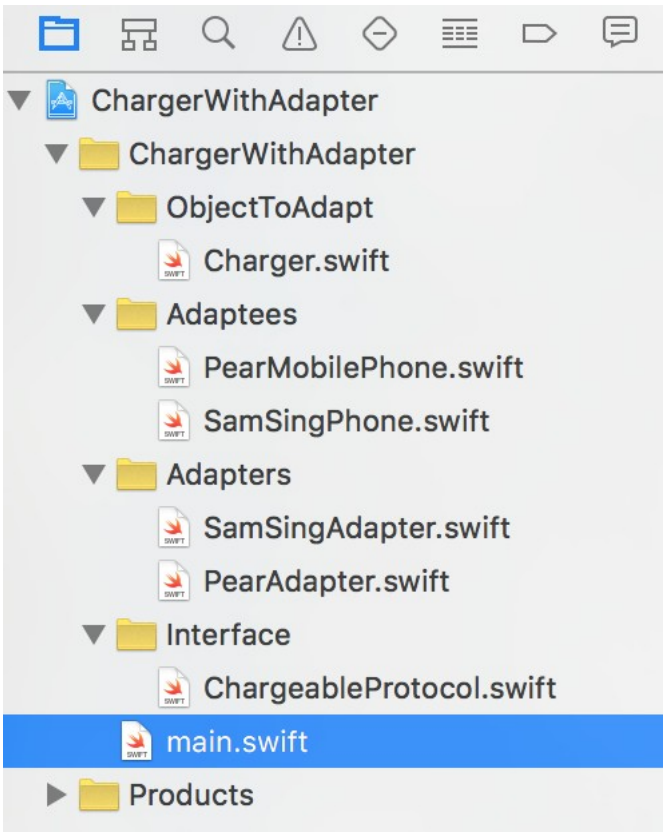
Parameters `volts` voltage needed to charge the battery

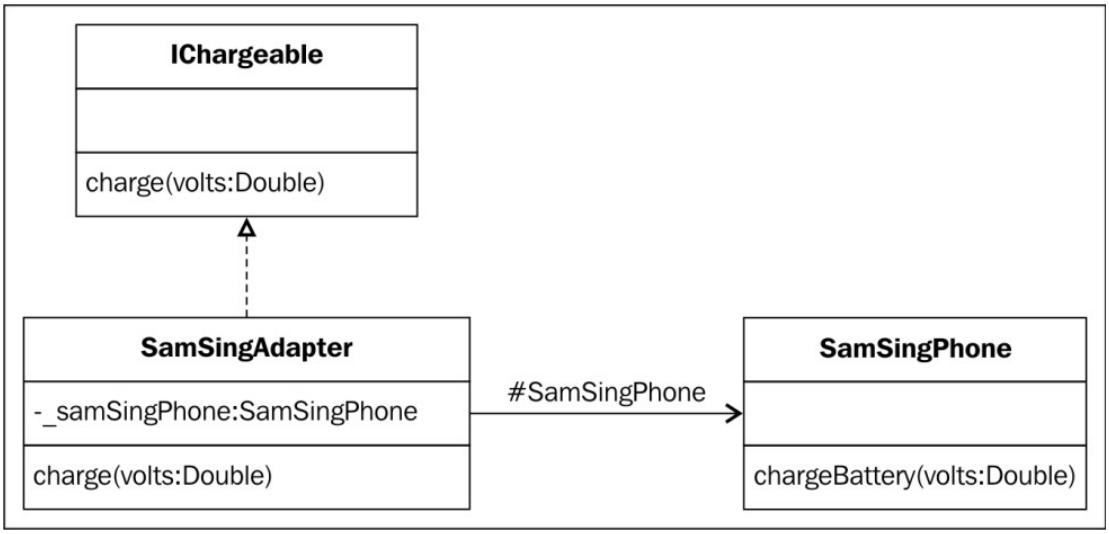
Returns Void

Declared In ChargeableProtocol.swift

```
First Prototype
A mobile is plugged
Charging our PhonePrototype
current voltage 10.0
Program ended with exit code: 0
```

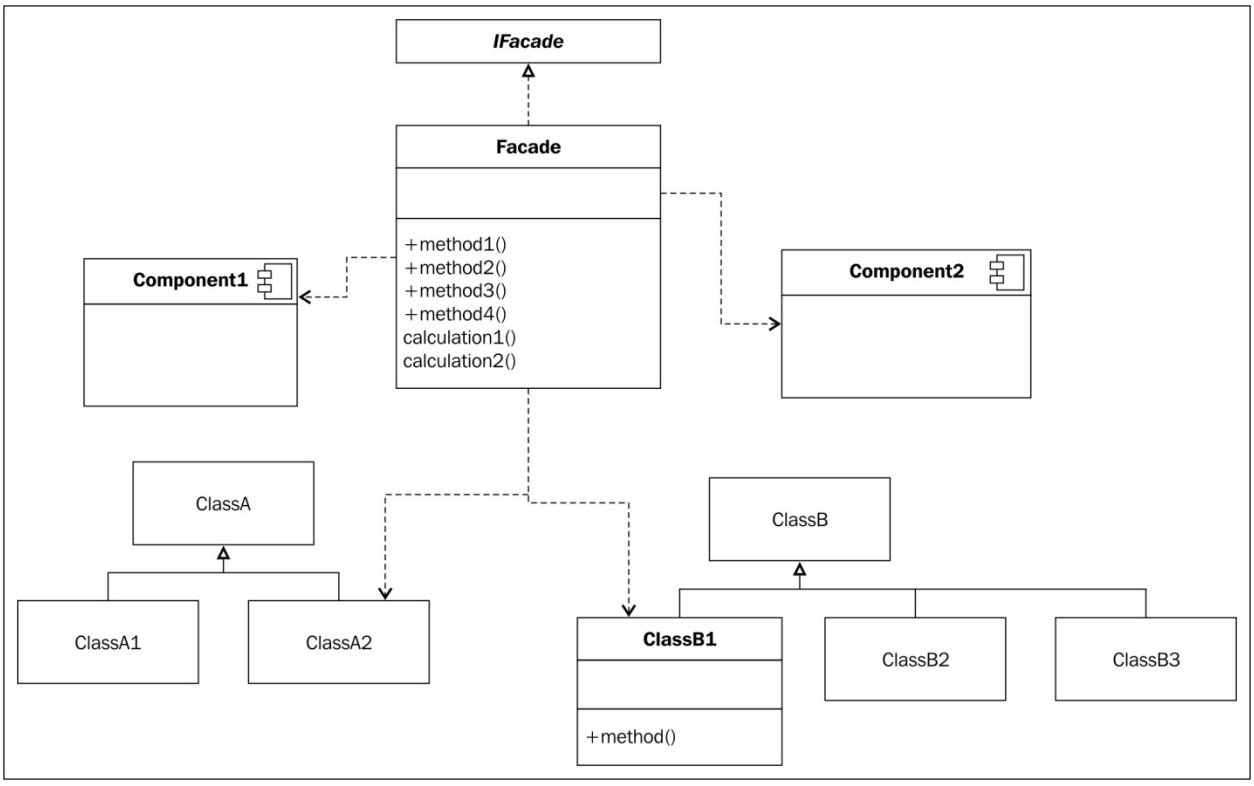
All Output

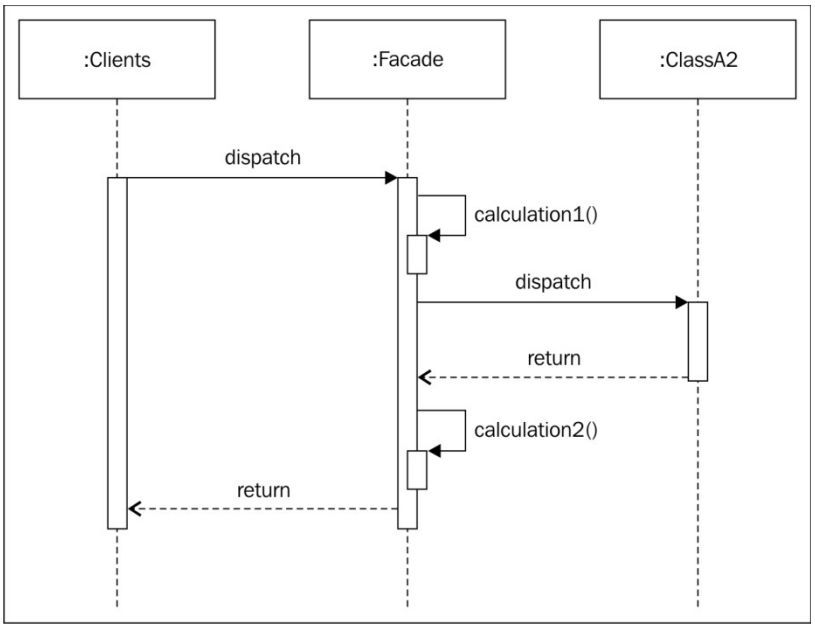

ChargerWithAdapter
ChargerWithAdapter
ObjectToAdapt
Charger.swift
Adaptees
PearMobilePhone.swift
SamSingPhone.swift
Adapters
SamSingAdapter.swift
PearAdapter.swift
Interface
ChargeableProtocol.swift
main.swift
Products

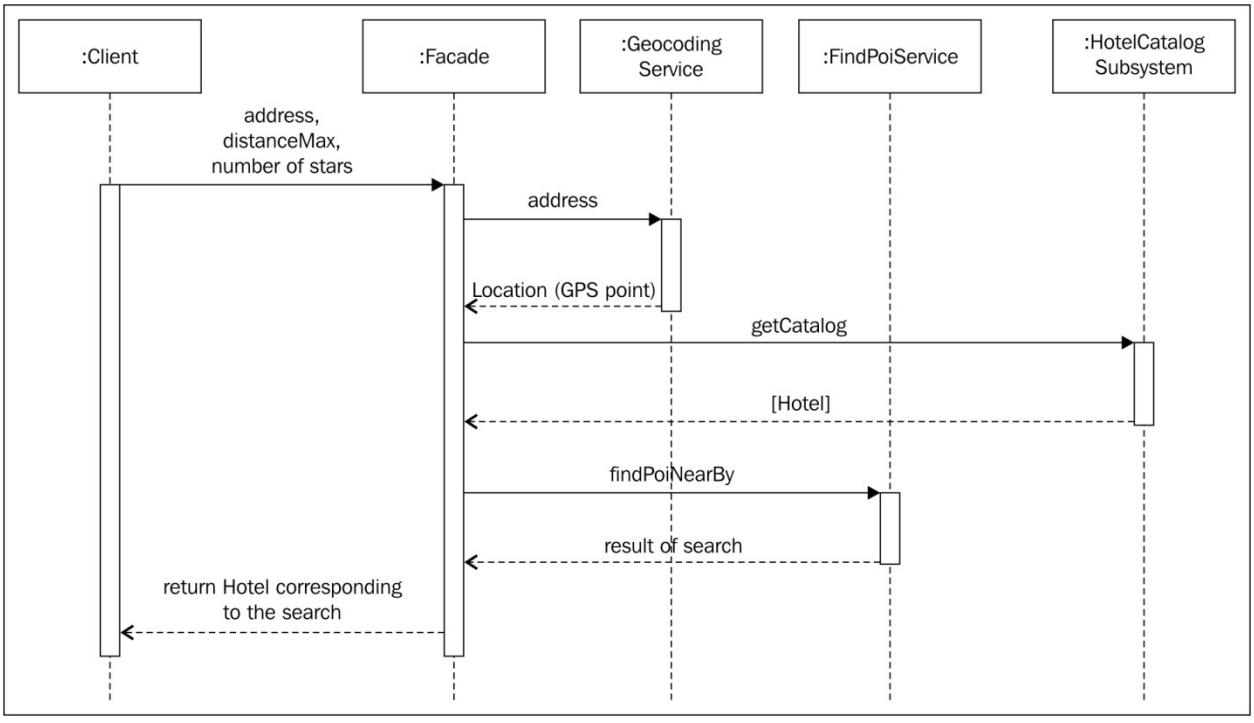

IChargeable
charge(volts:Double)
SamSingAdapter
-_samSingPhone:SamSingPhone
charge(volts:Double)
#SamSingPhone
SamSingPhone
chargeBattery(volts:Double)

```
*** start test program
*** charger ready test program
Will charge a Pear Mobile Phone
A mobile is plugged
Adapter started
Pear mobile phone is charging
Current voltage 5.5
Adapter ended
*** -
Will charge a SamSing Mobile Phone
A mobile is plugged
Adapter started
SamSing mobile phone is charging
Current voltage 10.0
Adapter ended
*** end test program
Program ended with exit code: 0
```

All Output



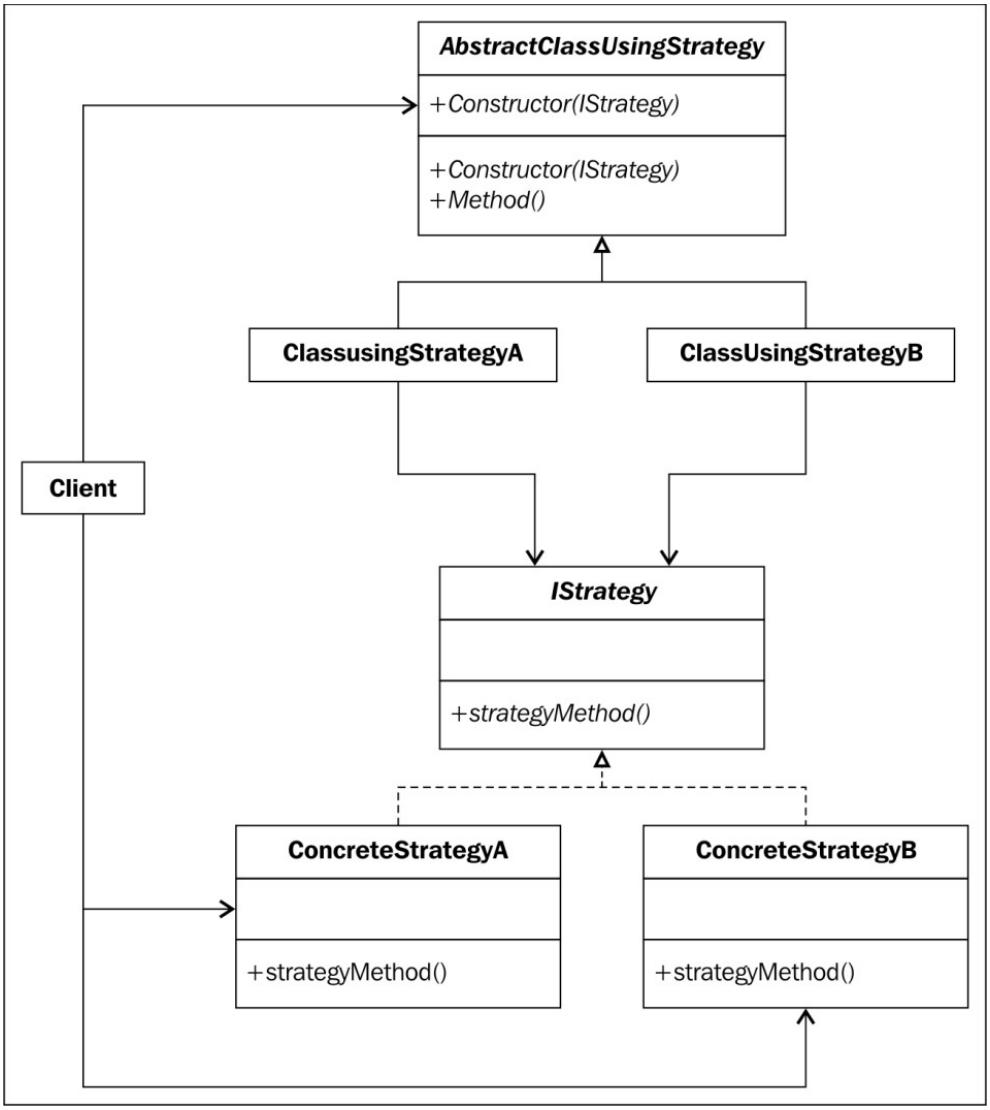

:Clients
:Facade
:ClassA2
dispatch
calculation1()
dispatch
return
calculation2()
return

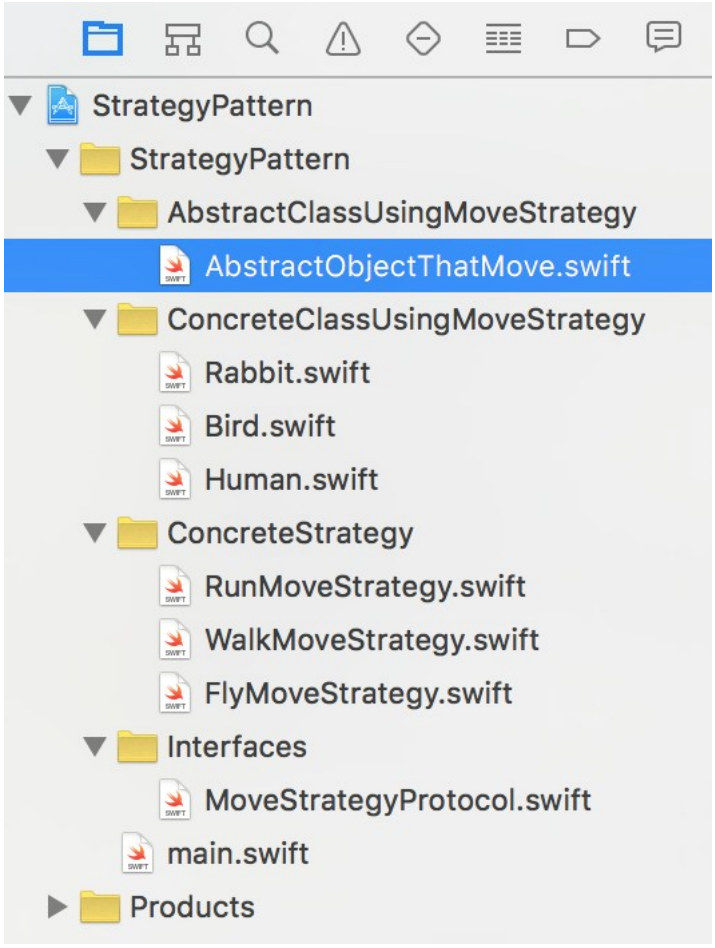

:Client
:Facade
:Geocoding Service
:FindPoiService
:HotelCatalog Subsystem
address, distanceMax, number of stars
address
Location (GPS point)
getCatalog
[Hotel]
findPoiNearBy
result of search
return Hotel corresponding to the search

```
*** RESULTS
Their is Optional(94) results :
Hotel latitude:Optional(-17.0) longitude:Optional(9.0), stars: 4
Hotel latitude:Optional(-44.0) longitude:Optional(-23.0), stars: 4
Hotel latitude:Optional(-1.0) longitude:Optional(-5.0), stars: 4
Hotel latitude:Optional(-22.0) longitude:Optional(8.0), stars: 4
Hotel latitude:Optional(-5.0) longitude:Optional(-25.0), stars: 4
Hotel latitude:Optional(5.0) longitude:Optional(-33.0), stars: 4
Hotel latitude:Optional(-30.0) longitude:Optional(-14.0), stars: 4
Hotel latitude:Optional(-34.0) longitude:Optional(-11.0), stars: 4
Hotel latitude:Optional(-35.0) longitude:Optional(0.0), stars: 4
Hotel latitude:Optional(-11.0) longitude:Optional(-31.0), stars: 4
Hotel latitude:Optional(-9.0) longitude:Optional(-3.0), stars: 4
Hotel latitude:Optional(-5.0) longitude:Optional(19.0), stars: 4
Hotel latitude:Optional(-9.0) longitude:Optional(-41.0), stars: 4
Hotel latitude:Optional(-19.0) longitude:Optional(-2.0), stars: 4
Hotel latitude:Optional(-23.0) longitude:Optional(33.0), stars: 4
Hotel latitude:Optional(-31.0) longitude:Optional(29.0), stars: 4
Hotel latitude:Optional(-22.0) longitude:Optional(-18.0), stars: 4
Hotel latitude:Optional(7.0) longitude:Optional(11.0), stars: 4
Hotel latitude:Optional(-28.0) longitude:Optional(21.0), stars: 4
```

All Output

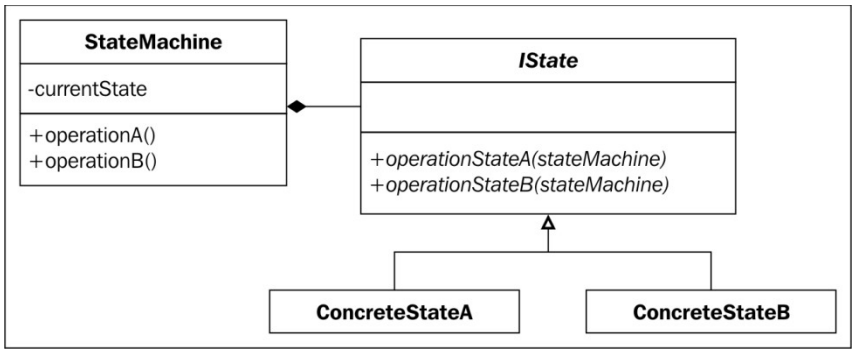
Chapter 5: Behavioral Patterns – Strategy, State, and Template Method

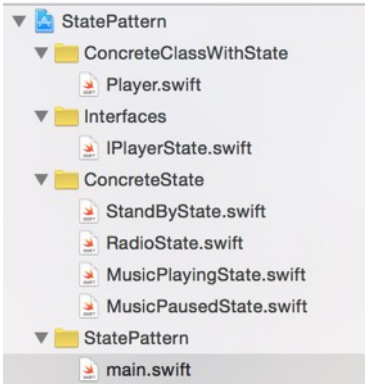





```
- *** working with Human
i'm a human
I am walking
- *** working with Bird
i'm a bird
I am flying
- *** working with Rabbit
i'm a rabbit
I am running
Program ended with exit code: 0
```

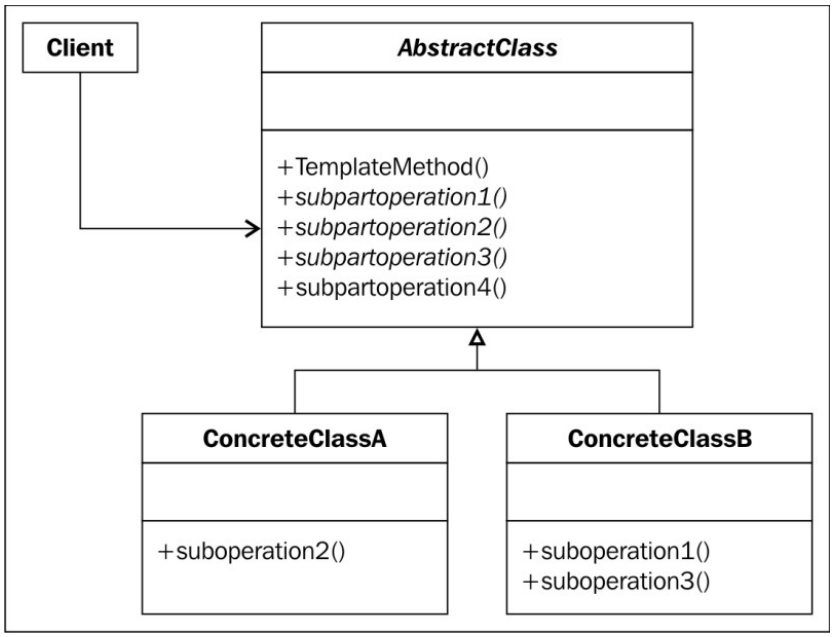
All Output

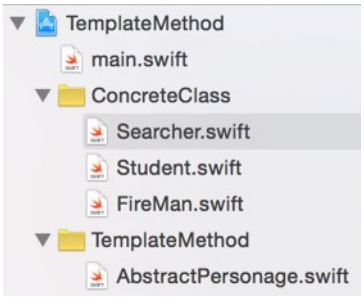





```
STANDBY MODE
cannot launch an action in standby mode
Changing to Radio Mode
RADIO MODE
Choosing next Station & playing it
Changing to MUSIC Mode
MUSIC PLAY MODE
Changing to Pausing Mode
MUSIC PAUSED MODE
Changing to playing Mode
MUSIC PLAY MODE
Changing source to Standby Mode
STANDBY MODE
Program ended with exit code: 0
```

All Output

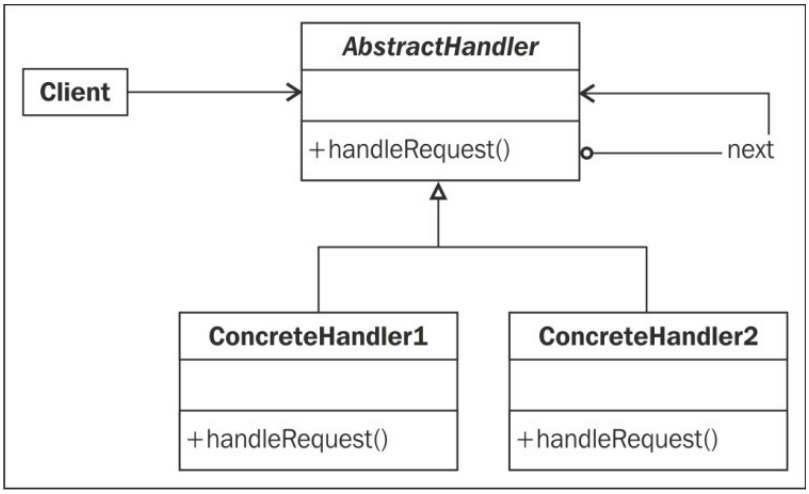


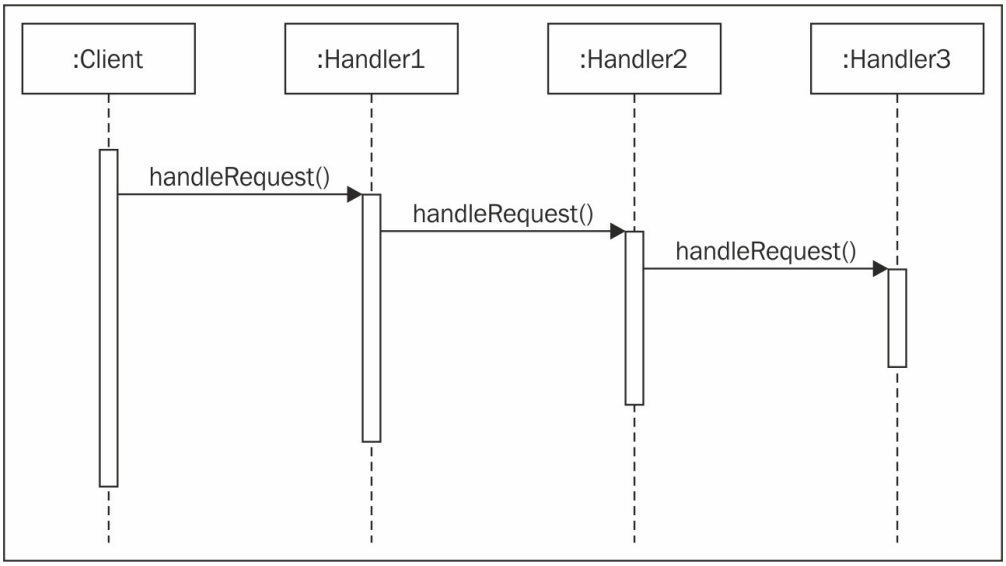



```
PLAYING DAY
Get Up!
Eat Breakfast
do washing up
Go to work
Work
Receive Pay
BackHome
Do personal activities
Eat dinner
do washing up
Sleep
- **** 30 days later:
Name: Simon / fatigue : 100 / happiness 1300 / Hungry 250 / knowledge 100 / money: 0 /
Name: Natasha / fatigue : 100 / happiness 1150 / Hungry 250 / knowledge 100 / money: 3000 /
Name: Edward / fatigue : 100 / happiness 250 / Hungry 100 / knowledge 100 / money: 2490 /
```

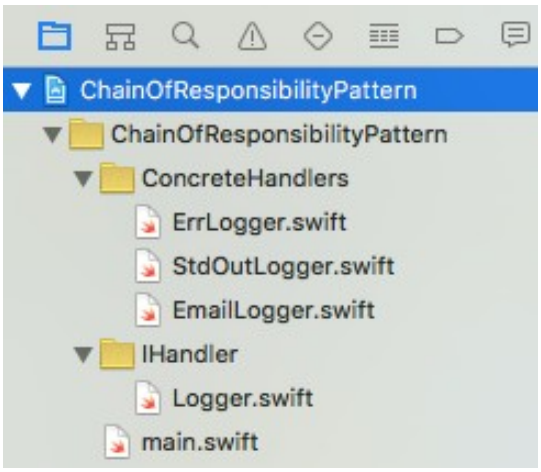
All Output

Chapter 6: Behavioral Patterns – Chain of Responsibility and Command

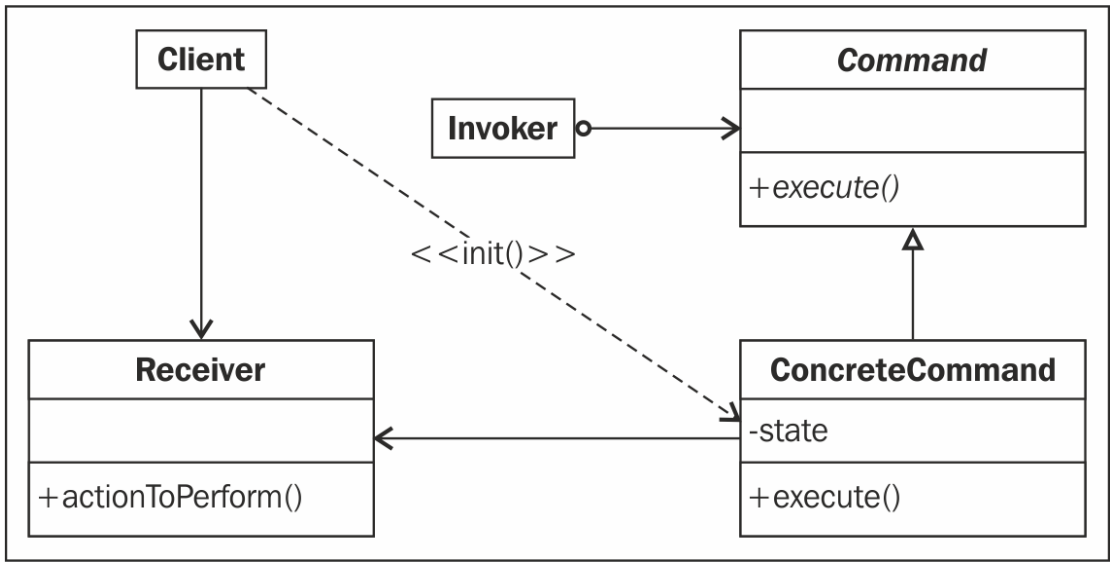


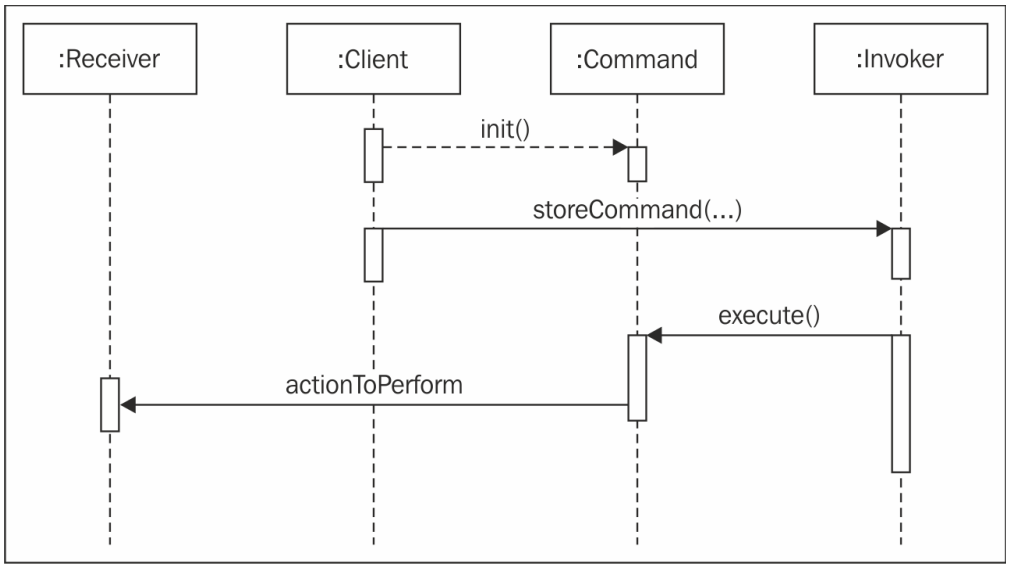


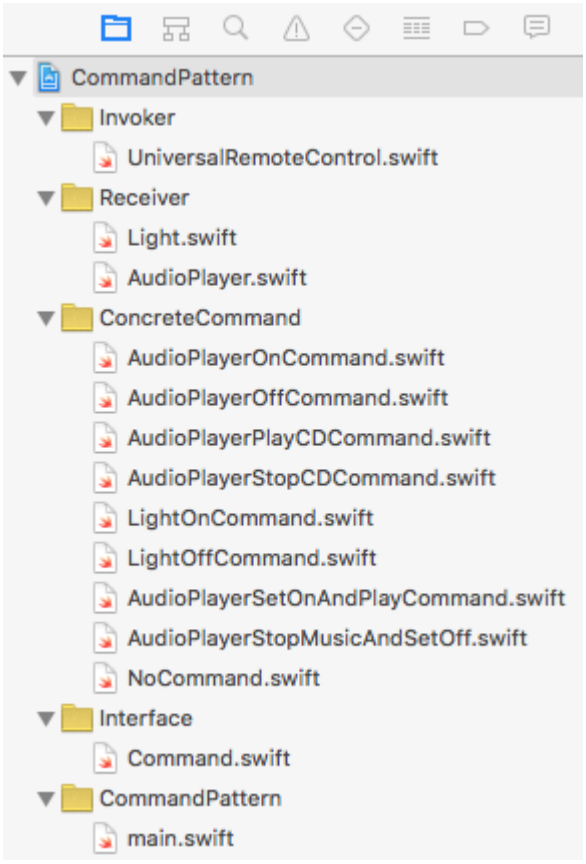

```
Building the Chain
fatal error: Must be overriden: file /Users/Admin/
Dropbox/Documents/Ebooks reviews/Ecriture Swift Design
Patterns/chapitres/Chapter6/code/CORPattern/
ChainOfResponsibilityPattern/
ChainOfResponsibilityPattern/Logger.swift, line 34
Program ended with exit code: 9
```

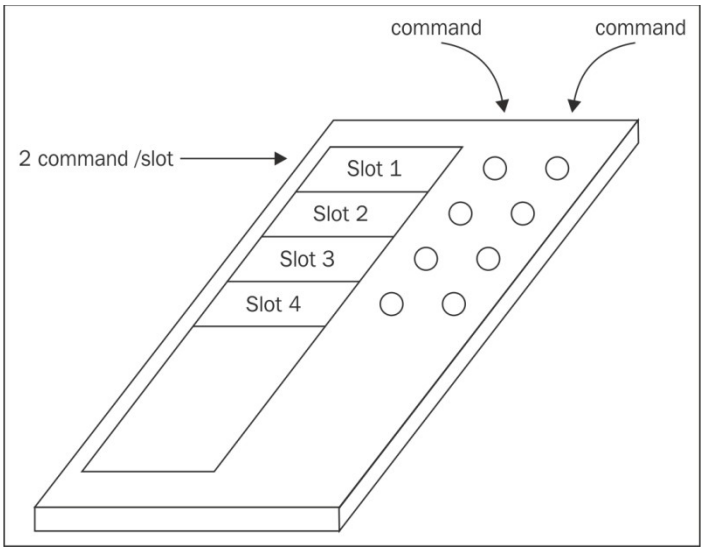



```
Building the Chain
- *** stdOutLogger:
Sending to StdOutLogger: Entering the func Y()
- StdOutLogger && EmailLogger:
Sending to StdOutLogger: Step 1 Completed
Sending by Email: Step 1 Completed
- all three loggers:
Sending to StdOutLogger: An error occured
Sending by Email: An error occured
Sending to ErrorLogger: An error occured
Program ended with exit code: 0
```



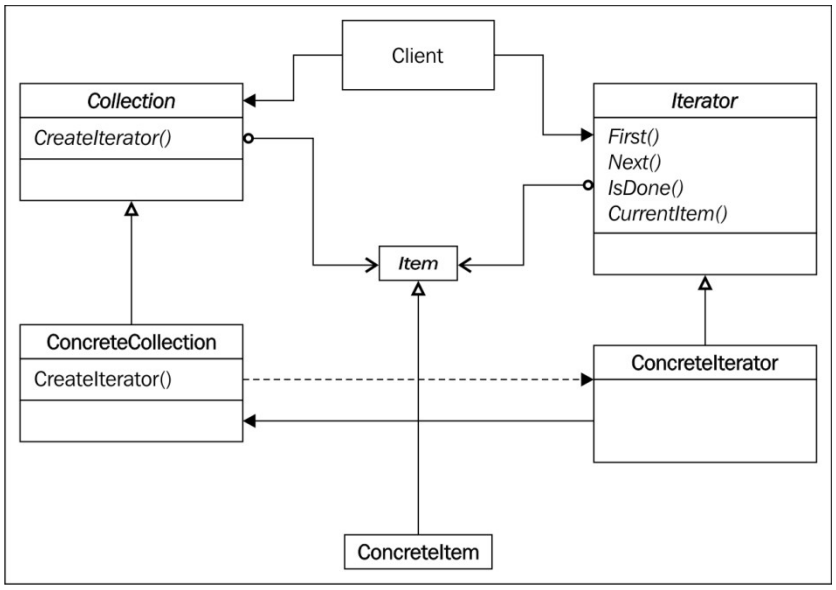

:Receiver
:Client
:Command
:Invoker
init()
storeCommand(...)
execute()
actionToPerform


CommandPattern
Invoker
UniversalRemoteControl.swift
Receiver
Light.swift
AudioPlayer.swift
ConcreteCommand
AudioPlayerOnCommand.swift
AudioPlayerOffCommand.swift
AudioPlayerPlayCDCommand.swift
AudioPlayerStopCDCommand.swift
LightOnCommand.swift
LightOffCommand.swift
AudioPlayerSetOnAndPlayCommand.swift
AudioPlayerStopMusicAndSetOff.swift
NoCommand.swift
Interface
Command.swift
CommandPattern
main.swift



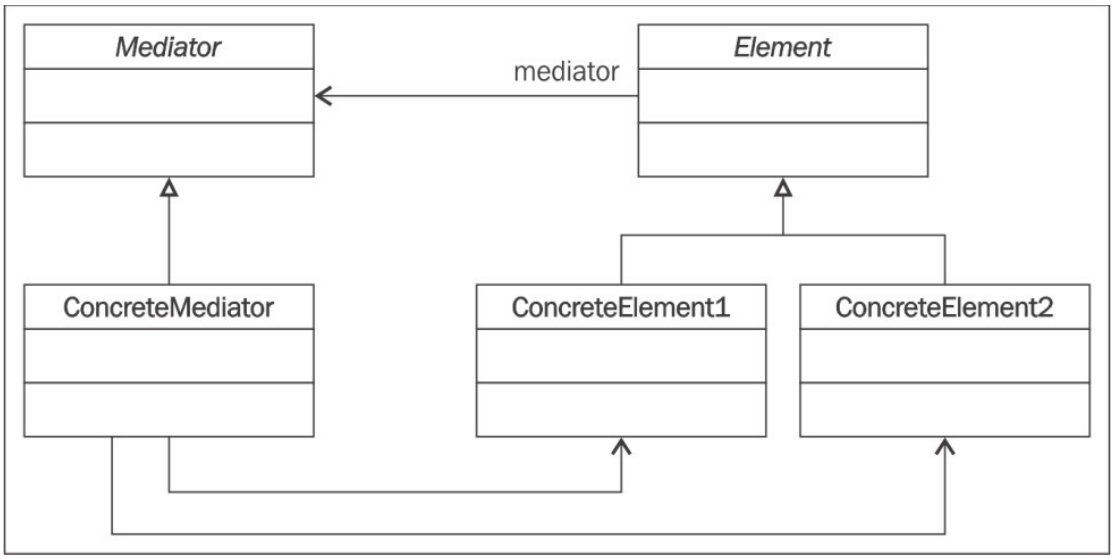

```
Light is On
Light is Off
Light is On
Light is Off
Audio Player is On
Audio Player is Off
Audio Player is On
AudioPlayer is playing
AudioPlayer has stopped to play music
Audio Player is Off
```

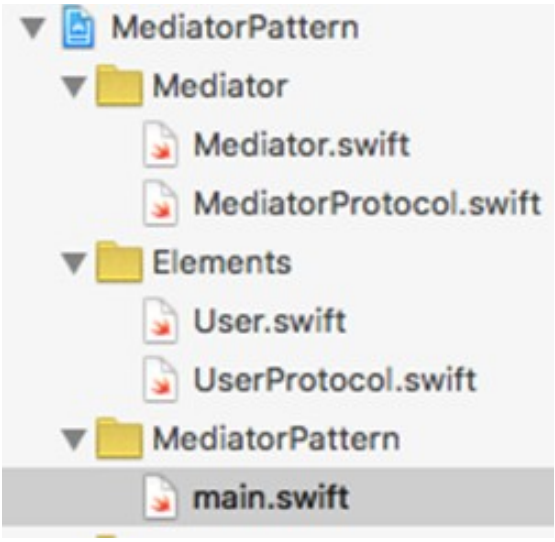
Chapter 7: Behavioral Patterns – Iterator, Mediator, and Observer



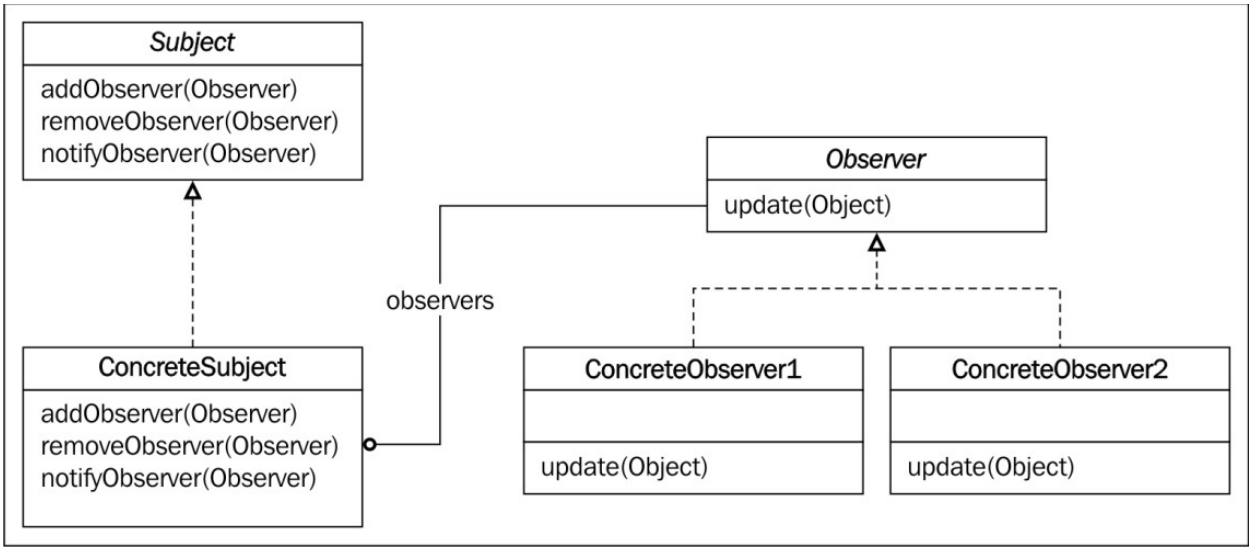

```
45 for player in players {                    Type 'OurCollection<Player>' does not conform to protocol 'SequenceType'
46   print("analysing \(player.name)")
47 }
```

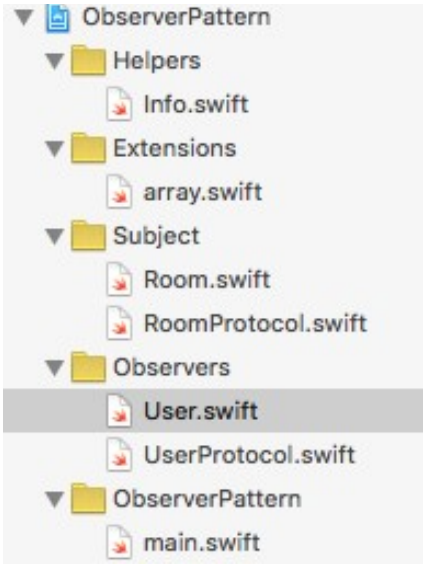
```
Name: Helmi
Name: Raphael
Name: Adrien
Name: Alain
Program ended with exit code: 0
```





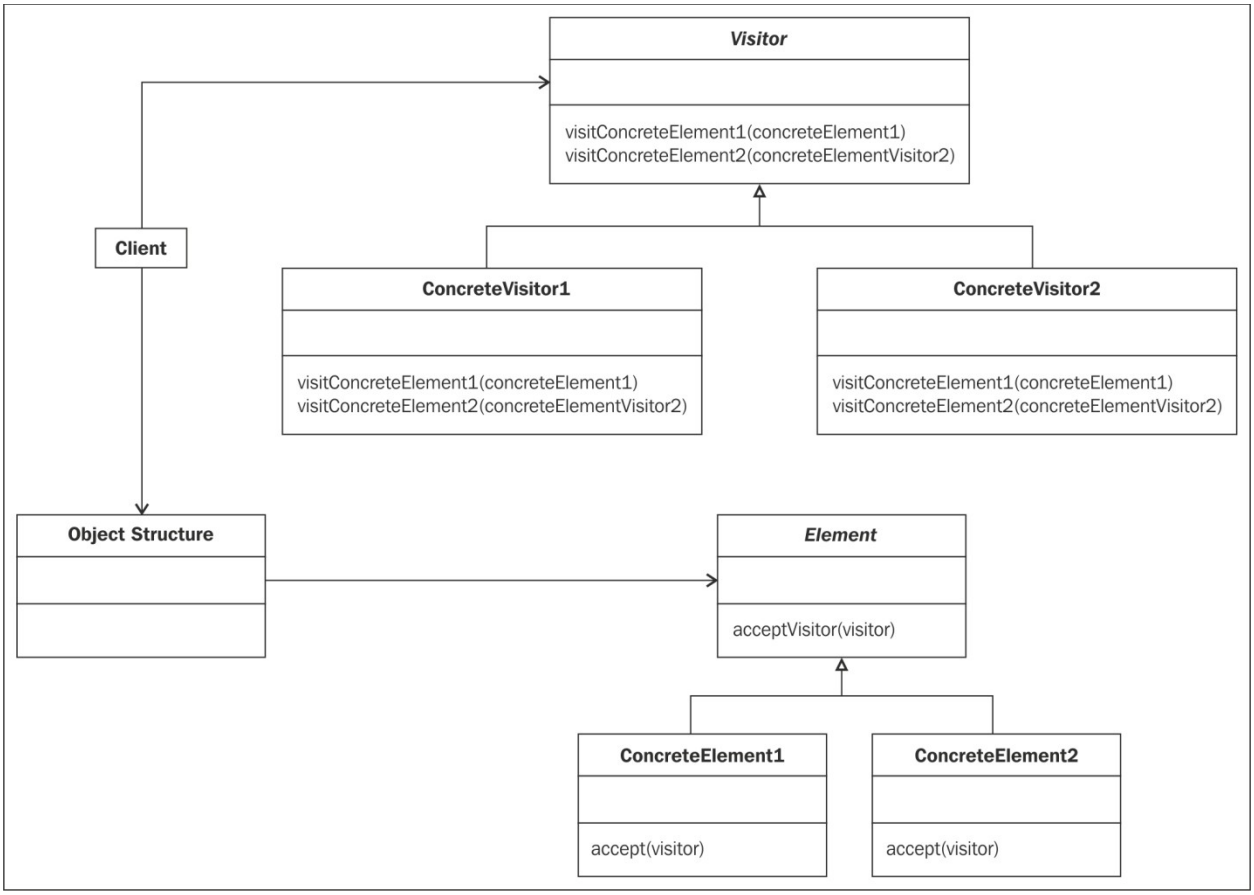

```
Helmi received message1 from Julien
Adrien received message1 from Julien
Raphael received message 2 from Helmi
Program ended with exit code: 0
```





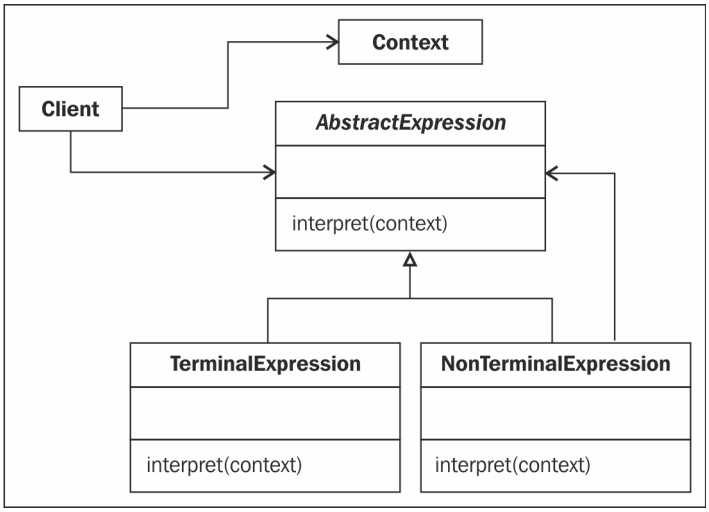

```
Julien notified that Julien have status Join on 2015-09-13 19:56:54 +0000
Julien notified that Alain have status Join on 2015-09-13 19:56:54 +0000
Alain notified that Alain have status Join on 2015-09-13 19:56:54 +0000
Julien notified that Helmi have status Join on 2015-09-13 19:56:54 +0000
Alain notified that Helmi have status Join on 2015-09-13 19:56:54 +0000
Helmi notified that Helmi have status Join on 2015-09-13 19:56:54 +0000
Julien notified that Raphael have status Join on 2015-09-13 19:56:54 +0000
Alain notified that Raphael have status Join on 2015-09-13 19:56:54 +0000
Helmi notified that Raphael have status Join on 2015-09-13 19:56:54 +0000
Raphael notified that Raphael have status Join on 2015-09-13 19:56:54 +0000
Julien notified that Alain have status Leave on 2015-09-13 19:56:54 +0000
Helmi notified that Alain have status Leave on 2015-09-13 19:56:54 +0000
Raphael notified that Alain have status Leave on 2015-09-13 19:56:54 +0000
Julien notified that Helmi have status Leave on 2015-09-13 19:56:54 +0000
Raphael notified that Helmi have status Leave on 2015-09-13 19:56:54 +0000
Raphael notified that Julien have status Leave on 2015-09-13 19:56:54 +0000
Raphael notified that Alain have status Join on 2015-09-13 19:56:54 +0000
Alain notified that Alain have status Join on 2015-09-13 19:56:54 +0000
Program ended with exit code: 0
```

Chapter 8: Behavioral Patterns – Visitor, Interpreter, and Memento




```
let prices = cars.map { (car: Car) -> Double in
  let visitor = PriceVisitor()
  car.accept(visitor)
  return visitor.price
}
```

```
[23 200, 15 300, 17 100]
(3 times)

(3 times)
```



```
14
MCCMXXVIII is not a correct roman number
1928
Program ended with exit code: 0
```

All Output