

# 7 :

		GoF	Abstract Factory
pattern			
Together			
Together			business rule

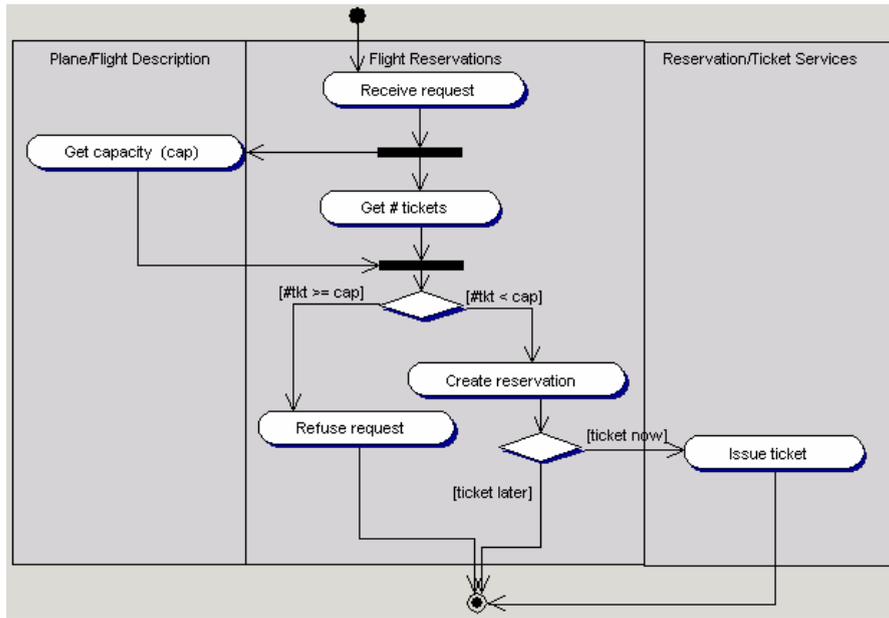
1. Business rule
2. GoF refactoring
3. code
4. Tip & Tricks

GoF “Gang of Four” Design Pattern, Element of Reusable  
 Object-Oriented Software Eric Gamma, Richard Helm, Ralph Johnson John Vlissides

## 1.

Flight.makeReservation() sequence  
 activity

” 가  
 “



Flight.makeReservation()

```
public void makeReservation(String name, int tKind) throws ReservationException {
    int t = this.numberOfTickets();
    int c = lnkFlightDescription.getCapacity();

    // Make a reservation only if the number of tickets sold is less than the plane capacity
    if (t < c) {
        Reservation r = new Reservation(nam,tKind);
        lnkReservation.add(r);
    }
    else throw new ReservationException();
}
```

Step : Flight.makeReservation()

Together

code sense

InkFlightDescription

code sense

FlightDescription

Editor " \* \* "

가

```
public class Flight {
    public void makeReservation(String name, int tKind) throws Re
    int t = this.numberOfTickets();
    int c = lnkFlightDescription.
```

clone()	Object
equals(Object)	boolean
finalize()	void
getCapacity()	int
getClass()	Class
hashCode()	int
notify()	void
notifyAll()	void
setCapacity(int)	void
toString()	String
wait()	void

Reservation stub . Reservation name tKind .  
 Name tKind .  
 , tKind 가 1 , Coach ticket . tKind 가 2 , FirstClass  
 ticket .

Step : Reservation

Reservation

```
public Reservation(String name, int tKind) {
    this.name = name;

    // We will refactor this next piece of code
    lnkTicket = null;
    if (tKind == 1)
        lnkTicket = new Coach() ;
    else if (tKind == 2)
        lnkTicket = new FirstClass();
}
```

## 2. Refactoring with a GoF pattern

airline 가 가 . ticket  
 가 Reservation .  
 ticket creation factory ticket  
 , Ticket TicketMaster . GoF collection  
 Abstract Factory

Step : GoF collection Abstract Factory, Concrete Factory and Abstract Product  
 Abstract Factory .

- IcreateTickets for Abstract Factory
- Ticket for Abstract Product
- TicketMaster for Concrete Factory



Choose Pattern                      Next

Next >

product                      text field 가

select box                      Select element

product                      Model

AirlinePD                      Ticket class                      , OK

Finish

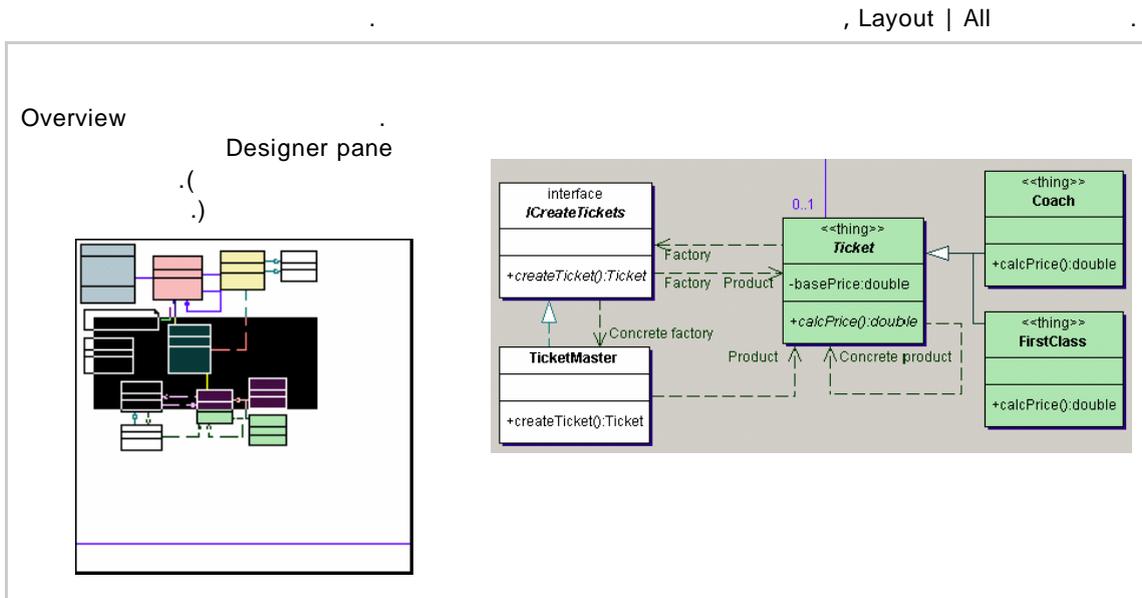
### 3. Putting the finishing touches on diagrams and code

가 process

TicketMaster

ICreateTickets                      dependencies                      가

Step :                      AirlinePD



Reservation

Ticket                      TicketMaseter

Step : IcreateTickets operation    int                      가                      , TicketMaster

1. ICreateTickets                      operation    in-place editor  
    createTicket(tKind:int):Ticket
2. IcreateTickets                      TicketMaster    createTicket(tKind:int)Ticket operation    Ctrl+drag  
    가                      TicketMaster.createTicket()

Step : Reservation                      Ticket    TicketMaster

Editor pane

가

Ticket

return

Ticket t (InITicket )

```
public Ticket createTicket(int tKind) {
    Ticket t = null;
    if (tKind == 1)
        t = new Coach();
    else if (tKind == 2)
        t = new FirstClass();
    return t;
}
```

1 2

final ( )

( ,

).

Step : Reservation	TicketMaster	Ticket
--------------------	--------------	--------

Reservation 가

```
public Reservation(String name, int tKind) {
    this.name = name;

    TicketMaster tm = new TicketMaster();
    lnkTicket = tm.createTicket(tKind);
}
```

Step : Code	
<p>Make build Project</p> <p>builder tab . Compiler output</p> <p>messages</p> <p>error 가 , error message Editor pane</p> <p>highligh</p>	<p>+ Make Project</p> <p>+++ Rebuild Project</p>

## 4. Tips and Tricks

- GoF Pattern . Composite pattern  
reservation .
- 가 .
- Ctrl+drag operation .  
Together operation .
- Editor Designer .. , method statement  
method method source .  
pane method

(<http://www.nextech.co.kr/>)

(<http://www.togethercommunity.co.kr>)

558-

4678