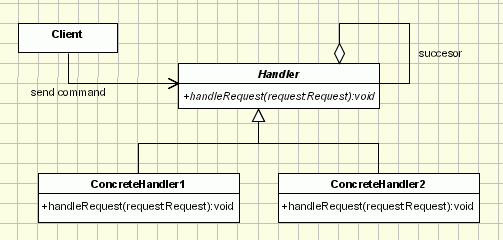
# Week 7 Lab – Implementing Object Orientated Designs

## Introduction

This week we discussed a number of design patterns and looked at their implementations. Although many of the patterns have now been discussed there are still many more that you may wish to look at. This lab will look at one of the patterns which was not covered in the lecture, the pattern is one of Chain of Responsibility. In this Patten allows an object to send a command without knowing what object will deal with the command. The request is sent to an object which may handle the request in a partial or full manor or it may not be able to handle the request. When the request is either not handled or only partly handled then the request is passed on to the next handler object in the chain.



Source: oodesign.com

Scenario.

We will look at modelling a hospital accident and emergency department. A patient attends an A&E department with a Condition. The Patient is assessed by a Triage Nurse who for minor ailments may treat to the patient or refer them to their GP service. If the Triage Nurse cannot treat the patient they are then given to the doctor who can treat a wider range of conditions. However the doctor may be unable to treat the patient and will pass them to a consultant. The consultant then can either treat the patient or in severe cases pass them to either surgery or a medical ward for further treatment.

List of Conditions: ChestPain, Injury, Infection, HeadTrama, GeneralPain, Other

List of Severity: VeryLow, Low, Medium, High, Critical

1. You need to create an Enumeration for both the Condition and Severity.
2. Add the following Patient Class into your project .

public class Patient

{

private String name;

public Condition condition { get; set; }

public Severity severity { get; set; }

public Patient(String name, Condition condition)

{

this.name = name;

this.condition = condition;

}

public override string ToString()

{

return name + " " +condition.ToString() +" "+ severity.ToString();

}

}

1. Creating the initial abstract Hander. The abstract hander contains the methods needed to chain the handlers. In the case of the consultant they will need to be able to send a patient to either a ward or to surgery so we need to make the method virtual.

public abstract class MedicalHandler

{

protected MedicalHandler successor;

public virtual void SetSuccessor(MedicalHandler successor)

{

this.successor = successor;

}

public abstract void HandlePatient(Patient patient);

}

1. Create the classes for Triage, Doctor, Consultant, AdmissionsWard and Sergery. All these classes need to inherit from MedicalHandler and implement the abstract method of the super class. You can leave them without an implementation for now. These methods will be completed in the following steps.
2. Rules For Triage.   
   Triage’s first task is to assess the severity of the patient’s condition. You will need to create a random number and based on the random number set the severity for the patient.

If the severity is very low then the patient is discharged and referred to their GP.

If the severity is low or medium and the condition is either an injury or an infection when the Triage nurse can treat the patient and discharge them.

In all other cases the Patient is referred to the Doctor.   
NOTE: you should always send a message to the console regarding which handler is dealing with the request, the decision made and the patient details.

1. Rules For Doctor.  
   In all cases of critical the doctor refers the patient to the Consultant.  
   If the condition is High and the condition is chest pain or head trauma the patient is referred to the consultant.  
   In all other cases the doctor will treat the patient and discharge them.
2. Overriding the Successor for the Consultant.  
   The consultant can pass the patient to either a ward or surgery. Therefore we need to modify the variable which holds the successor so it is not a list.  
   Additionally the SetSuccessor method will add the successor to the list.
3. Rules For Consultant.  
   If the condition is chest pain the patient is set to the ward

If the condition id head trauma or injury the patient is set to surgery

In all other cases the Consultant will treat the patient and discharge them.

1. AdmissionsWard and Surgery.   
   You should just send a message to the console with the patient details that they are with in a ward or surgery.
2. Add the code of the Main Method creating instances of all the classes. Use the SetSucessor methods to chain the MedicalHanders such that Triage->Doctor->Consultant and Consultant->AdmissionsWard and Consultant->Surgery
3. Add some patients and make sure the code is working as expected.