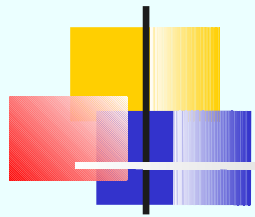




Công nghệ phần mềm

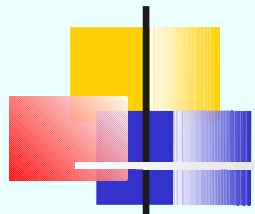
Pha cài đặt

Giảng viên: TS. Nguyễn Mạnh Hùng
Học viện Công nghệ Bưu chính Viễn thông (PTIT)



Nội dung tham khảo từ

Stephen R. Schach. *Object-Oriented and Classical Software Engineering*. Eighth Edition, WCB/McGraw-Hill, 2010



Pha cài đặt (1)

Mục đích:

- Cài đặt thành chương trình
- Kiểm thử chương trình



Chuẩn bị kiểm thử (1)

Thực hiện:

- Viết test case cho mỗi phương thức và mỗi lớp trước khi cài đặt chúng
- Test case dưới dạng hộp đen (black-box test):
 - Chỉ rõ đầu vào
 - Đầu ra mong muốn



Chuẩn bị kiểm thử (2)

Về nguyên tắc:

- Phải test hết tất cả các trường hợp có thể có của các kiểu dữ liệu
- Tuy nhiên nếu làm vậy số trường hợp phải test là quá lớn
- → dùng kĩ thuật test biên



Chuẩn bị kiểm thử (3)

Kĩ thuật test biên:

- Nếu một tham số đầu vào có một giới hạn biên **x**, thì phải test ít nhất 4 trường hợp:
 - 1: giá trị đầu vào bất kì **cách xa x**
 - 2: giá trị đầu vào **ngay trên x**
 - 3: giá trị đầu vào **ngay dưới x**
 - 4: giá trị đầu vào **đúng bằng x**

Ví dụ:

- Nếu phép chia cho số nguyên có điều kiện số bị chia $\neq 0$ thì phải test khi số bị chia: -1, 0, 1, và một số >100



Chuẩn bị kiểm thử (4)

Kĩ thuật test biên (tt):

- Nếu một tham số đầu vào có 2 giới hạn biên x_1 và x_2 thì phải test ít nhất 7 trường hợp:
 - 1,2: giá trị đầu vào đúng bằng x_1 , ngay trên x_1
 - 3,4: giá trị đầu vào ngay dưới x_2 , đúng bằng x_2
 - 5: giá trị đầu vào đúng bằng $(x_1+x_2)/2$
 - 6: giá trị đầu vào nhỏ hơn x_1
 - 7: giá trị đầu vào lớn hơn x_2



Chuẩn bị kiểm thử (5)

Kĩ thuật test chức năng thao tác CSDL:

- Nếu chức năng thêm một đối tượng vào CSDL thì phải test ít nhất 3 trường hợp:
 - 1: thêm một đối tượng **chưa có** trong CSDL
 - 2: thêm một đối tượng **đã có** trong CSDL
 - 3: thêm **liên tục 2 lần** một đối tượng **chưa có** trong CSDL



Chuẩn bị kiểm thử (6)

Kĩ thuật test chức năng thao tác CSDL (tt):

- Nếu chức năng sửa một đối tượng trong CSDL thì phải test ít nhất 3 trường hợp:
 - 1: sửa một đối tượng **chưa có** trong CSDL
 - 2: sửa một đối tượng **đã có** trong CSDL
 - 3: sửa **liên tục 2 lần** một thuộc tính của đối tượng **đã có** trong CSDL



Chuẩn bị kiểm thử (7)

Kĩ thuật test chức năng thao tác CSDL (tt):

- Nếu chức năng xóa một đối tượng trong CSDL thì phải test ít nhất 3 trường hợp:
 - 1: xóa một đối tượng **chưa có** trong CSDL
 - 2: xóa một đối tượng **đã có** trong CSDL
 - 3: xóa **liên tục 2 lần** một đối tượng **đã có** trong CSDL



Chuẩn bị kiểm thử (8)

Kĩ thuật test chức năng thao tác CSDL (tt):

- Nếu chức năng tìm kiếm một số đối tượng trong CSDL thì phải test ít nhất 2 trường hợp:
 - 1: tìm kiếm một đối tượng **chưa có** trong CSDL
 - 2: tìm kiếm một đối tượng **đã có** trong CSDL



Chuẩn bị kiểm thử (9)

Nếu CSDL có 3 bảng: **KhachHang** lưu thông tin khách hàng, bảng **SanPham** lưu thông tin sản phẩm, bảng **HoaDon** lưu thông tin một khách hàng mỗi lần mua một số sản phẩm. Khi đó, chức năng thêm một hóa đơn vào trong CSDL thì phải test ít nhất 2 trường hợp:

- 1: thêm một hóa đơn **chưa có** trong CSDL
- 2: thêm một hóa đơn **đã có** trong CSDL
- 3: thêm một hóa đơn mà **khách hàng chưa có** trong CSDL
- 4: thêm một hóa đơn mà **sản phẩm chưa có** trong CSDL
- 5: thêm một hóa đơn mà **cả khách hàng và sản phẩm chưa có** trong CSDL



Chuẩn bị kiểm thử (10)

Ví dụ test case cho chức năng thêm một quyển sách vào CSDL:

- B1: Dữ liệu hiện thời:

id	name	author	release year
0	Java core	Steve Jobs	2011
1	Software Engineering	Scatches	2003



Chuẩn bị kiểm thử (11)

Ví dụ test case cho chức năng thêm một quyển sách vào CSDL (tt):

- B2: Các thao tác, và kết quả mong đợi (có thể dùng requirement diagram/test case trong VP):

Các bước thao tác

1. NV chọn chức năng thêm sách

2. Nhân viên nhập:

Id=2,

Tên = Data mining

Tác giả = Jennings

Năm xb = 2012

Và click nút submit 1 lần

Kết quả mong đợi

Giao diện thêm sách hiện ra, gồm các ô nhập: id, tên, tác giả, năm xb và nút submit

Thông báo thêm thành công và CSDL sẽ có như sau:

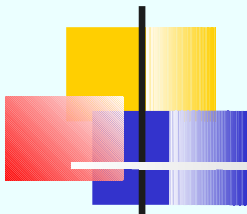


Chuẩn bị kiểm thử (12)

Ví dụ test case cho chức năng thêm một quyển sách vào CSDL (tt):

- B3: kết quả mong đợi trong CSDL:

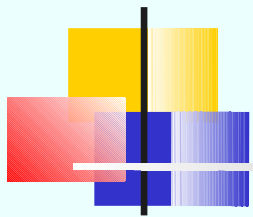
id	name	author	release year
0	Java core	Steve Jobs	2011
1	Software Engineering	Scatches	2003
2	Data mining	Jennings	2012



Bài tập

Viết test case cho các chức năng sau:

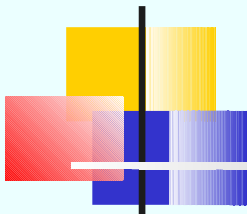
- Sửa thông tin phòng
- Xóa thông tin phòng
- Đặt phòng tại quầy
- Checkin
- Trả phòng và thanh toán



Bài tập về nhà

Với mỗi modul cá nhân:

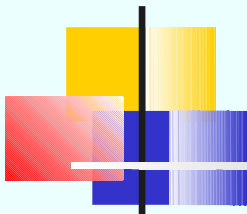
- Viết các test case cho từng chức năng của modul



Cài đặt (1)

Thực hiện theo thứ tự:

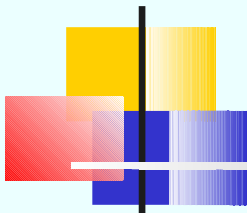
- Cài đặt các lớp thực thể
- Cài đặt các lớp giao diện
- Cài đặt các lớp điều khiển



Cài đặt (2)

Chú thích code:

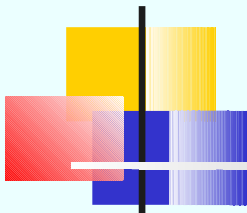
- Chú thích code là cần thiết và quan trọng cho pha bảo trì
- Nên chú thích code đầu mỗi lớp và đầu mỗi phương thức



Cài đặt (3)

Chú thích code cần có các thông tin tối thiểu:

- Tên của phần code, mô tả ngắn gọn chức năng và hoạt động của đoạn code
- Tên người code và ngày code
- Ngày được duyệt và tên người duyệt
- Tên các tham số, theo abc, giải thích ngắn gọn ý nghĩa từng tham số
- Tên các file truy cập/thay đổi bởi đoạn code
- Đầu vào – đầu ra
- Khả năng xử lý ngoại lệ
- Danh sách các thay đổi, cập nhật từ lần code đầu



Cài đặt (4)

Đặt tên biến:

- Tên biến nên đặt theo cách gợi nhớ, trừ các biến chạy
- Bắt đầu bằng chữ thường
- Thống nhất với nhau

Ví dụ:

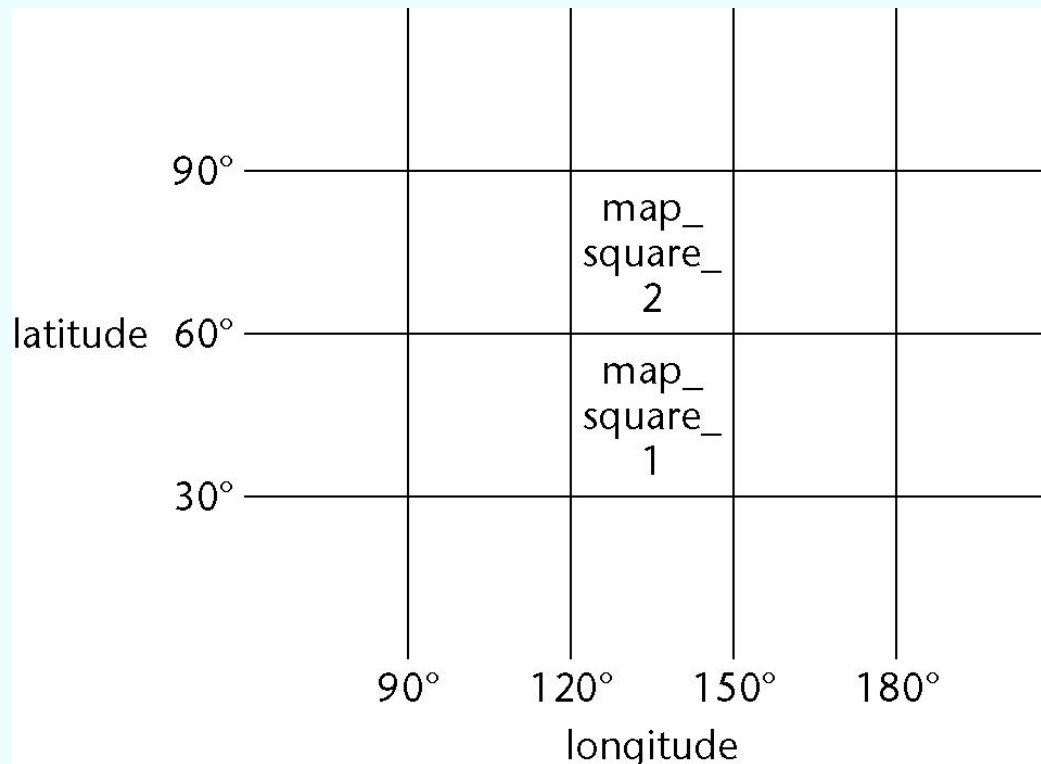
- Có thể sử dụng: frequencyAverage, frequencyTotal
- Có thể sử dụng: averageFrequency, totalFrequency
- Nhưng **không thể** đặt tên cùng nhau:
FrequencyAverage, totalFrequency

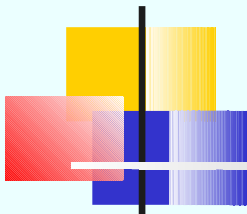


Cài đặt (5)

Xét ví dụ với câu lệnh if:

- Xác định xem một điểm có tọa độ nằm trong vùng map_square1 hoặc map_square2 hoặc không





Cài đặt (6)

Cách 1:

- Trình bày không chấp nhận được

```
if (latitude > 30 && longitude > 120) {if (latitude <= 60 && longitude <= 150)  
mapSquareNo = 1; else if (latitude <= 90 && longitude <= 150) mapSquareNo = 2  
else print "Not on the map";} else print "Not on the map";
```



Cài đặt (7)

Cách 2:

- Khuôn dạng được, nhưng cấu trúc không được

```
if (latitude > 30 && longitude > 120)
{
    if (latitude <= 60 && longitude <= 150)
        mapSquareNo = 1;
    else
        if (latitude <= 90 && longitude <= 150)
            mapSquareNo = 2;
        else
            print "Not on the map";
}
else
    print "Not on the map";
```

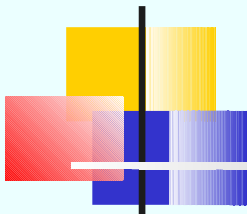



Cài đặt (8)

Cách 3:

- Chấp nhận được

```
if (longitude > 120 && longitude <= 150 && latitude > 30 && latitude <= 60)
    mapSquareNo = 1;
else
    if (longitude > 120 && longitude <= 150 && latitude > 60 && latitude <= 90)
        mapSquareNo = 2;
    else
        print "Not on the map";
```



Cài đặt (9)

Nguyên tắc dùng lệnh if:

- Nếu có lệnh if-if liên nhau dạng

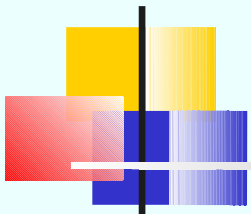
If (điều kiện 1)

If (điều kiện 2) làm x

→ thì nên chuyển thành:

If (điều kiện 1) && (điều kiện 2) làm x

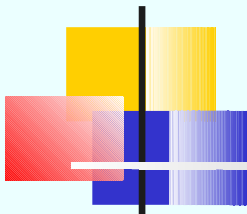
- Nếu câu lệnh if lồng nhau có độ sâu 3 tầng trở lên thì nên xem xét lại dùng cách khác để điều khiển



Cài đặt (10)

Nguyên tắc phân chia và code modul/ method:

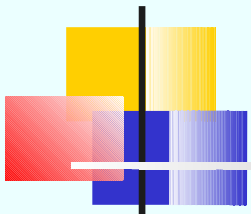
- Một method (modul) chỉ nên có tối đa 30-50 câu lệnh
- Nếu có nhiều hơn thì nên tách thành các modul con



Tích hợp (1)

Các kỹ thuật tích hợp:

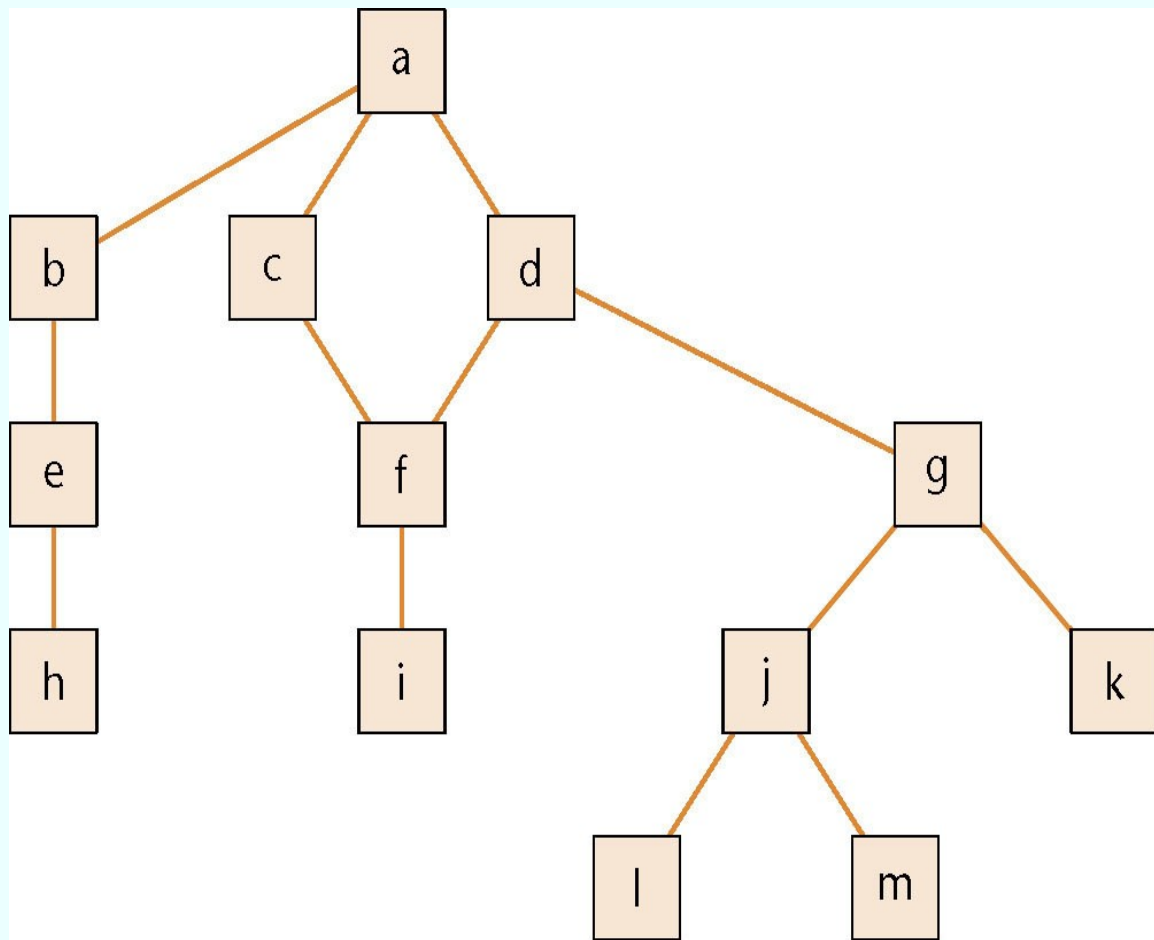
- Top-down: Tích hợp từ trên xuống
- Bottom-up: Tích hợp từ dưới lên
- Sandwich: Tích hợp theo cả hai chiều trên xuống và dưới lên

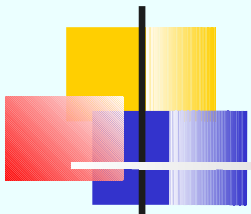


Tích hợp (2)

Top-down:

- Thứ tự tích hợp là:
a,b,c,d,e,f,g,h,i,j
,k,l,m

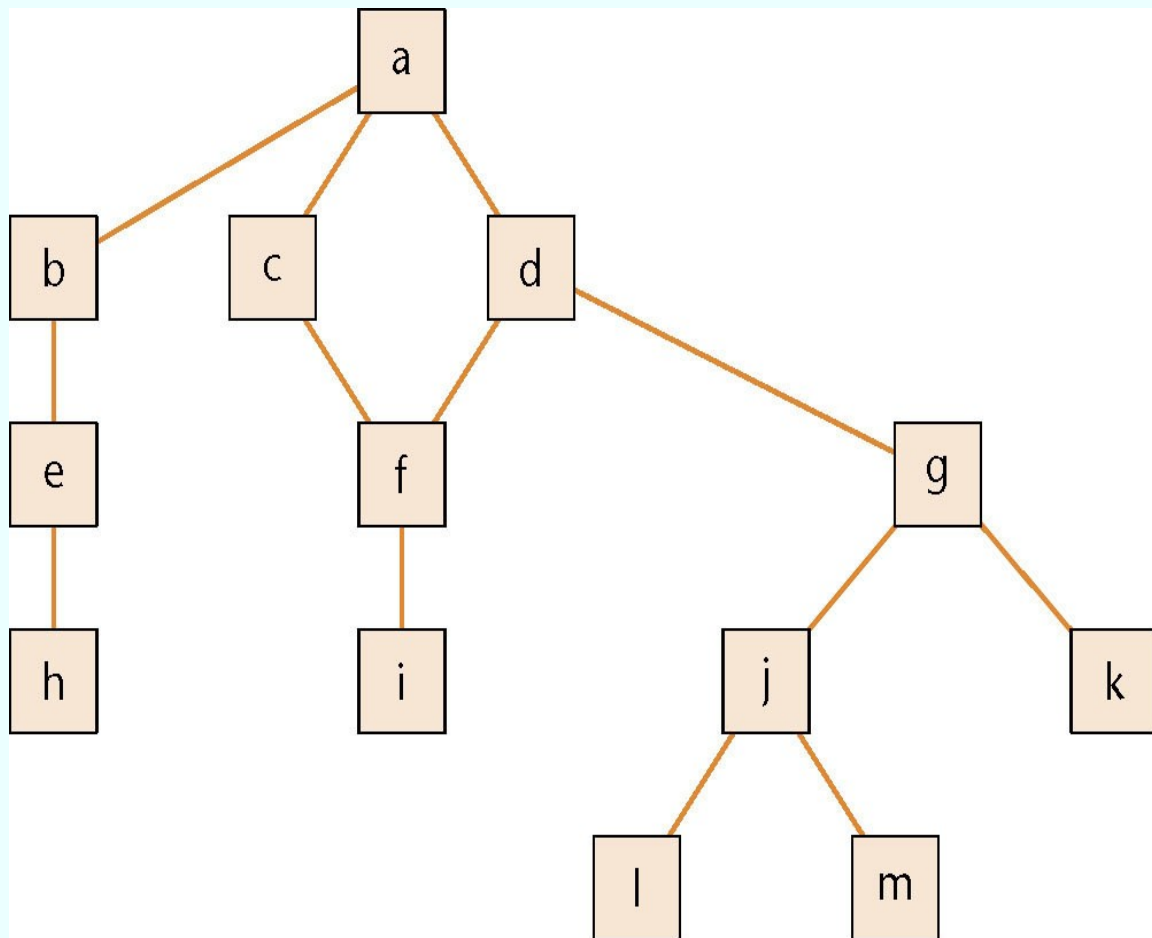


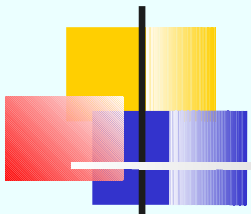


Tích hợp (3)

Top-down (tt):

- Để test modul a, phải coi các modul b,c,d là các hằng số (stubs)

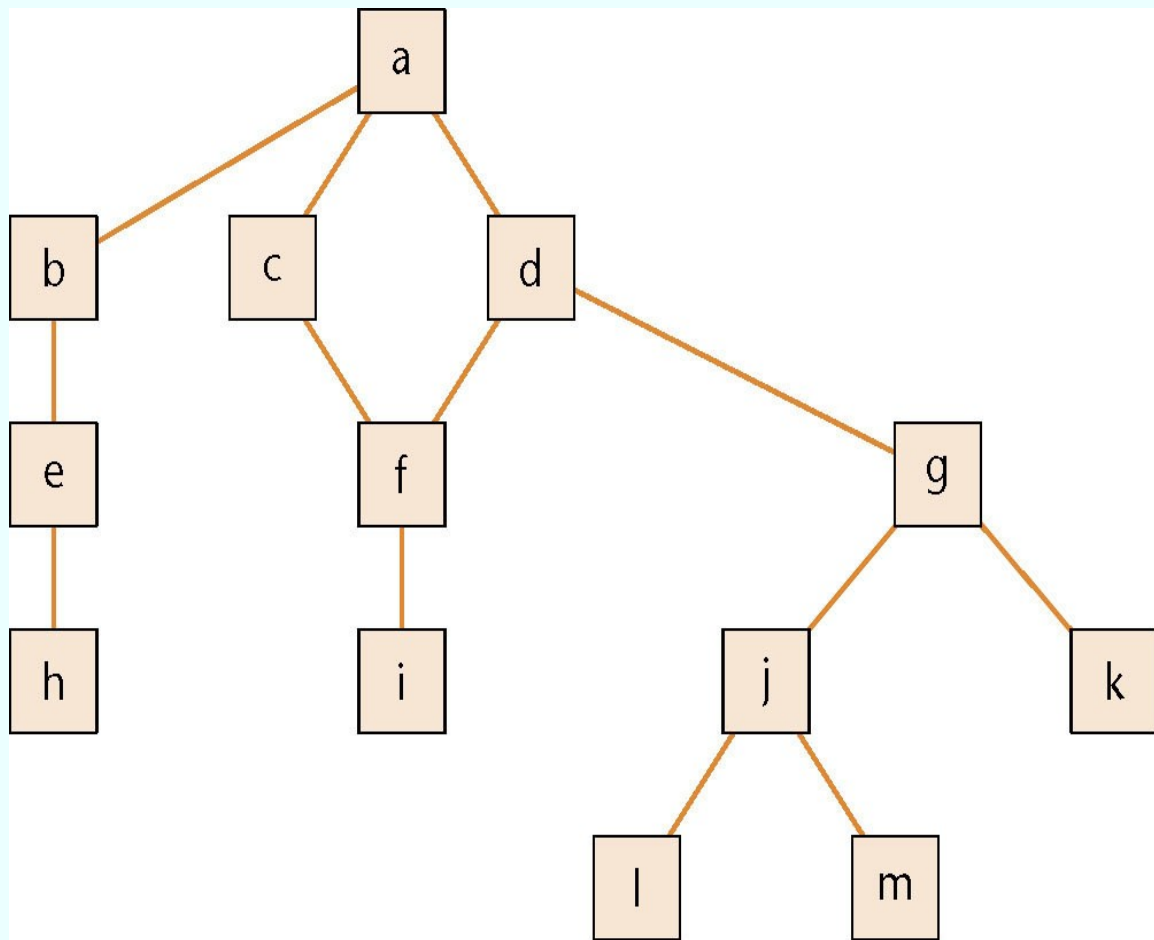


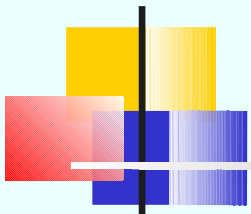


Tích hợp (4)

Bottom-up:

- Thứ tự tích hợp là:
l,m,h,i,j,k,e,f,g,b
,c,d,a

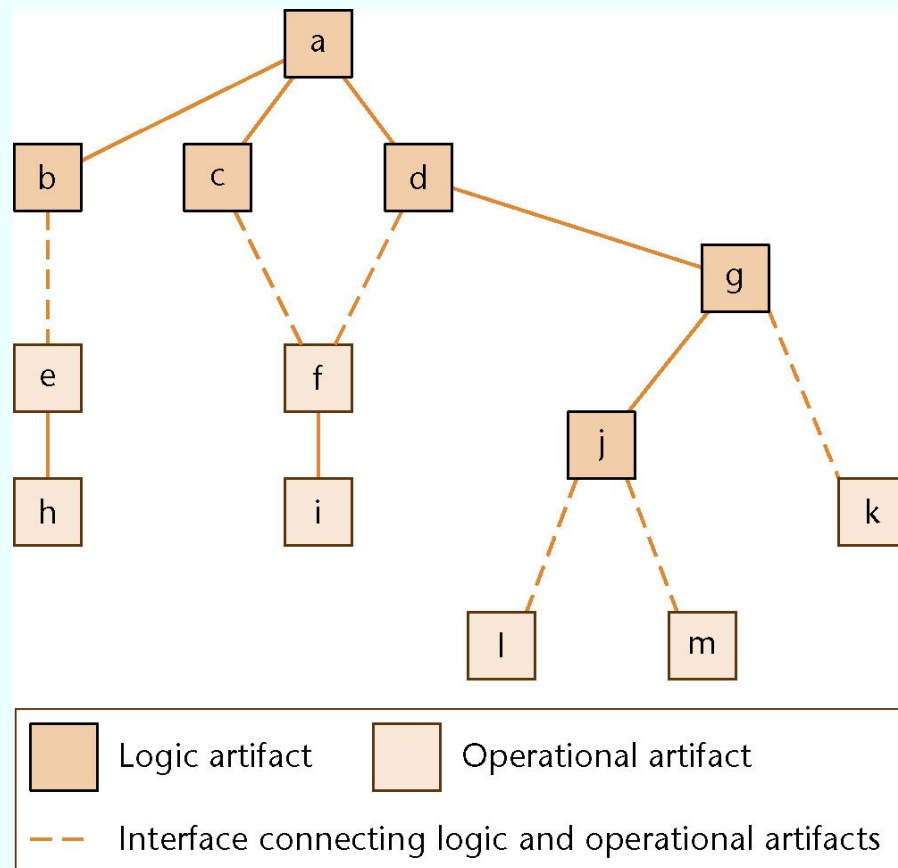


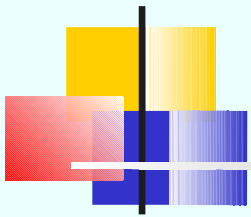


Tích hợp (5)

Sandwich:

- Các modul logic thì tích hợp top-down
- Các modul thực hiện trực tiếp thì tích hợp bottom-up

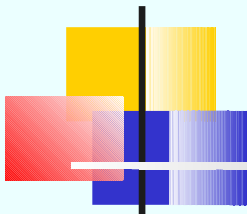




Kiểm thử (1)

Với mỗi modul/ method:

- Chạy các test case đã viết trong phần đầu, lưu kết quả chạy thành nhật kí chạy test case



Kiểm thử (2)

Ví dụ với modul thêm một sách:

- Chạy các test case đã viết trong phần đầu, lưu kết quả chạy thành nhật kí chạy test case

Các test case	Kết quả
1. thêm một sách chưa có id trong CSDL	passed
2. Thêm một sách đã có id trong CSDL	error
3. Thêm liên tục 2 lần một sách chưa có id trong csdl	error

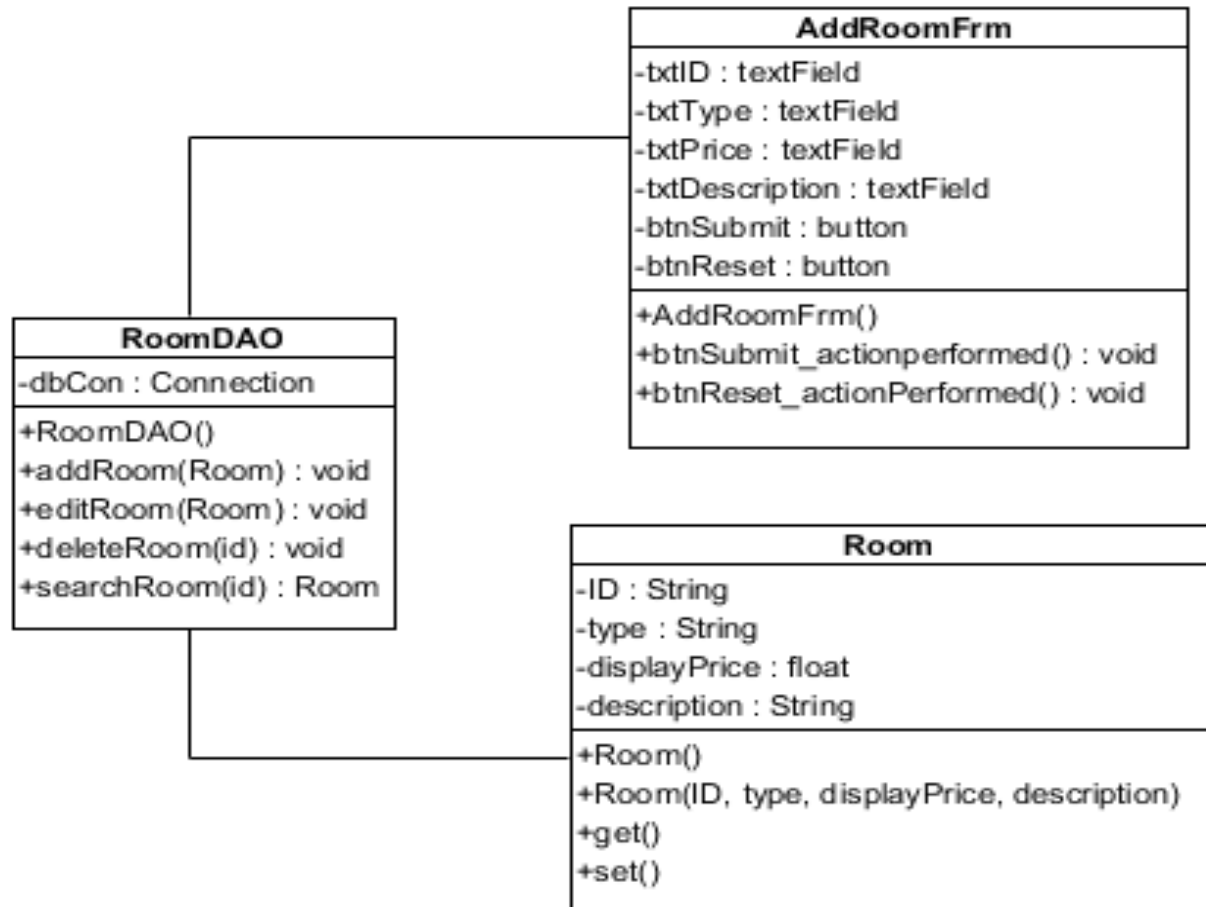


Ví dụ Cài đặt theo thiết kế MVC thuần

*Tham khảo chi tiết tại:
<http://coderandcode.blogspot.com/2014/02/variations-of-mvc-model.html>*

Sơ đồ lớp pha thiết kế






Với chức năng thêm phòng:





Các bảng CSDL liên quan

Bảng chứa thông tin phòng:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI
 id	VARCHAR(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 type	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 displayPrice	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 description	VARCHAR(250)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



Lớp Room (1)

```
package mvcPure;
```

```
public class Room {  
    private String id;  
    private String name;  
    private String type;  
    private float displayPrice;  
    private String description;
```

```
    public Room() {  
        super();  
    }
```

```
    public Room(String id, String name, String type,  
        float displayPrice, String description) {  
        super();  
        this.id = id;  
        this.name = name;  
        this.type = type;  
        this.displayPrice = displayPrice;  
        this.description = description;  
    }
```



Lớp Room (2)

```
public String getId() {
    return id;
}
public void setId(String id) {
    this.id = id;
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
public String getType() {
    return type;
}
public void setType(String type) {
    this.type = type;
}
public float getDisplayPrice() {
    return displayPrice;
}
public void setDisplayPrice(float displayPrice) {
    this.displayPrice = displayPrice;
}
public String getDescription() {
    return description;
}
public void setDescription(String description) {
    this.description = description;
}
```



Lớp RoomDAO (1)

```
package mvcPure;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class RoomDAO {
    private Connection con;

    public RoomDAO(){
        String dbUrl = "jdbc:mysql://localhost:3306/hotel";
        String dbClass = "com.mysql.jdbc.Driver";

        try {
            Class.forName(dbClass);
            con = DriverManager.getConnection (dbUrl,
                                                "root", "12345678");
        }catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```




Lớp RoomDAO (2)

```
public void addRoom(Room room){
    String sql = "INSERT INTO tblRoom(id, name, type, displayPrice,
        description) VALUES(?,?,?,?,?)";
    try{
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, room.getId());
        ps.setString(2, room.getName());
        ps.setString(3, room.getType());
        ps.setFloat(4, room.getDisplayPrice());
        ps.setString(5, room.getDescription());

        ps.executeUpdate();
    }catch(Exception e){
        e.printStackTrace();
    }
}
```



Lớp AddRoomFrm (1)

```
package mvcPure;
```

```
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.WindowAdapter;  
import java.awt.event.WindowEvent;
```

```
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JTextField;
```

```
public class AddRoomFrm extends JFrame implements ActionListener{  
    private JTextField txtID;  
    private JTextField txtName;  
    private JTextField txtType;  
    private JTextField txtDisplayPrice;  
    private JTextField txtDescription;  
    private JButton btnSubmit;  
    private JButton btnReset;
```



Lớp AddRoomFrm (2)

```
public AddRoomFrm(){
    super("Room management pure-MVC");
    txtID = new JTextField(15);
    txtName = new JTextField(15);
    txtType = new JTextField(15);
    txtDisplayPrice = new JTextField(15);
    txtDescription = new JTextField(15);
    btnSubmit = new JButton("Submit");
    btnReset = new JButton("Reset");

    JPanel content = new JPanel();
    content.setLayout(new GridLayout(6,2));
    content.add(new JLabel("ID:")); content.add(txtID);
    content.add(new JLabel("Name:")); content.add(txtName);
    content.add(new JLabel("Type:")); content.add(txtType);
    content.add(new JLabel("Display price:")); content.add(txtDisplayPrice);
    content.add(new JLabel("Description:")); content.add(txtDescription);
    content.add(btnReset); content.add(btnSubmit);
    btnSubmit.addActionListener(this);
    btnReset.addActionListener(this);
    this.setContentPane(content);
    this.pack();

    this.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e){
            System.exit(0);
        }
    });
}
```



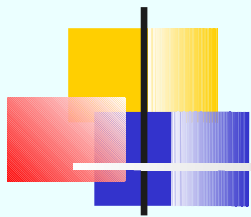
Lớp AddRoomFrm (3)

```
public void actionPerformed(ActionEvent e) {
    JButton btn = (JButton) e.getSource();
    if(btn.equals(btnSubmit)){
        btnSubmit_actionPerformed();
    }else if(btn.equals(btnReset)){
        btnReset_actionPerformed();
    }
}

public void btnSubmit_actionPerformed() {
    Room room = new Room();
    room.setId(txtID.getText());
    room.setName(txtName.getText());
    room.setType(txtType.getText());
    room.setDisplayPrice(Float.parseFloat(txtDisplayPrice.getText()));
    room.setDescription(txtDescription.getText());

    RoomDAO rd = new RoomDAO();
    rd.addRoom(room);
    JOptionPane.showMessageDialog(this, "Add room successfullly!");
}

public void btnReset_actionPerformed() {
    txtID.setText(""); txtName.setText("");
    txtType.setText(""); txtDisplayPrice.setText("");
    txtDescription.setText("");
}
}
```



Lớp Test

```
package mvcPure;

public class Test {

    public static void main(String[] args) {
        AddRoomFrm arf = new AddRoomFrm();
        arf.setVisible(true);
    }
}
```

Lưu ý trước khi chạy, phải:

- Cài đặt CSDL và bật server MySQL
- Add driver của Jdbc mysql vào library của project

Ví dụ

Cài đặt theo thiết kế MVC bean



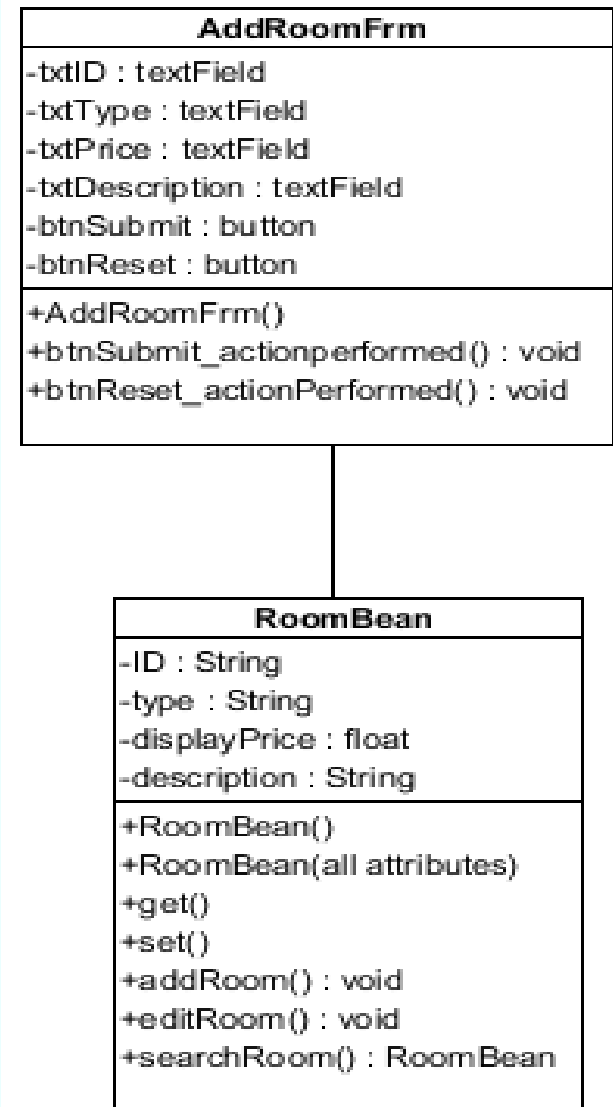
Tham khảo chi tiết tại:

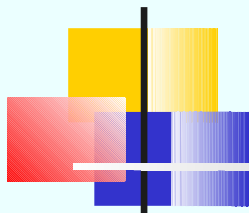
<http://coderandcode.blogspot.com/2014/02/variations-of-mvc-model.html>



Sơ đồ lớp pha thiết kế

Với chức năng thêm phòng:





Lớp RoomBean (1)

```
package mvcBean;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class RoomBean {
    private String id;
    private String name;
    private String type;
    private float displayPrice;
    private String description;

    public RoomBean() {
        super();
    }
    public RoomBean(String id, String name, String type, float
        DisplayPrice, String description) {
        super();
        this.id = id;
        this.name = name;
        this.type = type;
        this.displayPrice = displayPrice;
        this.description = description;
    }
}
```




Lớp RoomBean (2)

```
public String getId() {  
    return id;  
}  
public void setId(String id) {  
    this.id = id;  
}  
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public String getType() {  
    return type;  
}  
public void setType(String type) {  
    this.type = type;  
}  
public float getDisplayPrice() {  
    return displayPrice;  
}  
public void setDisplayPrice(float displayPrice) {  
    this.displayPrice = displayPrice;  
}  
public String getDescription() {  
    return description;  
}  
public void setDescription(String description) {  
    this.description = description;  
}
```



Lớp RoomBean (3)

```
public void addRoom(){
    String dbUrl = "jdbc:mysql://localhost:3306/hotel";
    String dbClass = "com.mysql.jdbc.Driver";
    String sql = "INSERT INTO tblRoom(id, name, type, displayPrice,
        description) VALUES(?,?,?,?,?)";
    try{
        Class.forName(dbClass);
        Connection con = DriverManager.getConnection (
            dbUrl, "root", "12345678");
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, getId());
        ps.setString(2, getName());
        ps.setString(3, getType());
        ps.setFloat(4, getDisplayPrice());
        ps.setString(5, getDescription());

        ps.executeUpdate();
        con.close();
    }catch(Exception e){
        e.printStackTrace();
    }
}
```



Lớp AddRoomFrm (1)

```
package mvcBean;
```

```
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.WindowAdapter;  
import java.awt.event.WindowEvent;
```

```
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JTextField;
```

```
public class AddRoomFrm extends JFrame implements ActionListener{  
    private JTextField txtID;  
    private JTextField txtName;  
    private JTextField txtType;  
    private JTextField txtDisplayPrice;  
    private JTextField txtDescription;  
    private JButton btnSubmit;  
    private JButton btnReset;
```



Lớp AddRoomFrm (2)

```
public AddRoomFrm(){
    super("Room management pure-MVC");
    txtID = new JTextField(15);
    txtName = new JTextField(15);
    txtType = new JTextField(15);
    txtDisplayPrice = new JTextField(15);
    txtDescription = new JTextField(15);
    btnSubmit = new JButton("Submit");
    btnReset = new JButton("Reset");

    JPanel content = new JPanel();
    content.setLayout(new GridLayout(6,2));
    content.add(new JLabel("ID:")); content.add(txtID);
    content.add(new JLabel("Name:")); content.add(txtName);
    content.add(new JLabel("Type:")); content.add(txtType);
    content.add(new JLabel("Display price:")); content.add(txtDisplayPrice);
    content.add(new JLabel("Description:")); content.add(txtDescription);
    content.add(btnReset); content.add(btnSubmit);
    btnSubmit.addActionListener(this);
    btnReset.addActionListener(this);
    this.setContentPane(content);
    this.pack();

    this.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e){
            System.exit(0);
        }
    });
}
```



Lớp AddRoomFrm (3)

```
public void actionPerformed(ActionEvent e) {
    JButton btn = (JButton) e.getSource();
    if(btn.equals(btnSubmit)){
        btnSubmit_actionPerformed();
    }else if(btn.equals(btnReset)){
        btnReset_actionPerformed();
    }
}

public void btnSubmit_actionPerformed() {
    RoomBean room = new RoomBean();
    room.setId(txtID.getText());
    room.setName(txtName.getText());
    room.setType(txtType.getText());
    room.setDisplayPrice(Float.parseFloat(txtDisplayPrice.getText()));
    room.setDescription(txtDescription.getText());

    room.addRoom();
    JOptionPane.showMessageDialog(this, "Add room successfullly!");
}

public void btnReset_actionPerformed() {
    txtID.setText(""); txtName.setText("");
    txtType.setText(""); txtDisplayPrice.setText("");
    txtDescription.setText("");
}
}
```



Lớp Test

```
package mvcBean;  
  
public class Test {  
  
    public static void main(String[] args) {  
        AddRoomFrm arf = new AddRoomFrm();  
        arf.setVisible(true);  
    }  
}
```

Lưu ý trước khi chạy, phải:

- Cài đặt CSDL và bật server MySQL
- Add driver của Jdbc mysql vào library của project

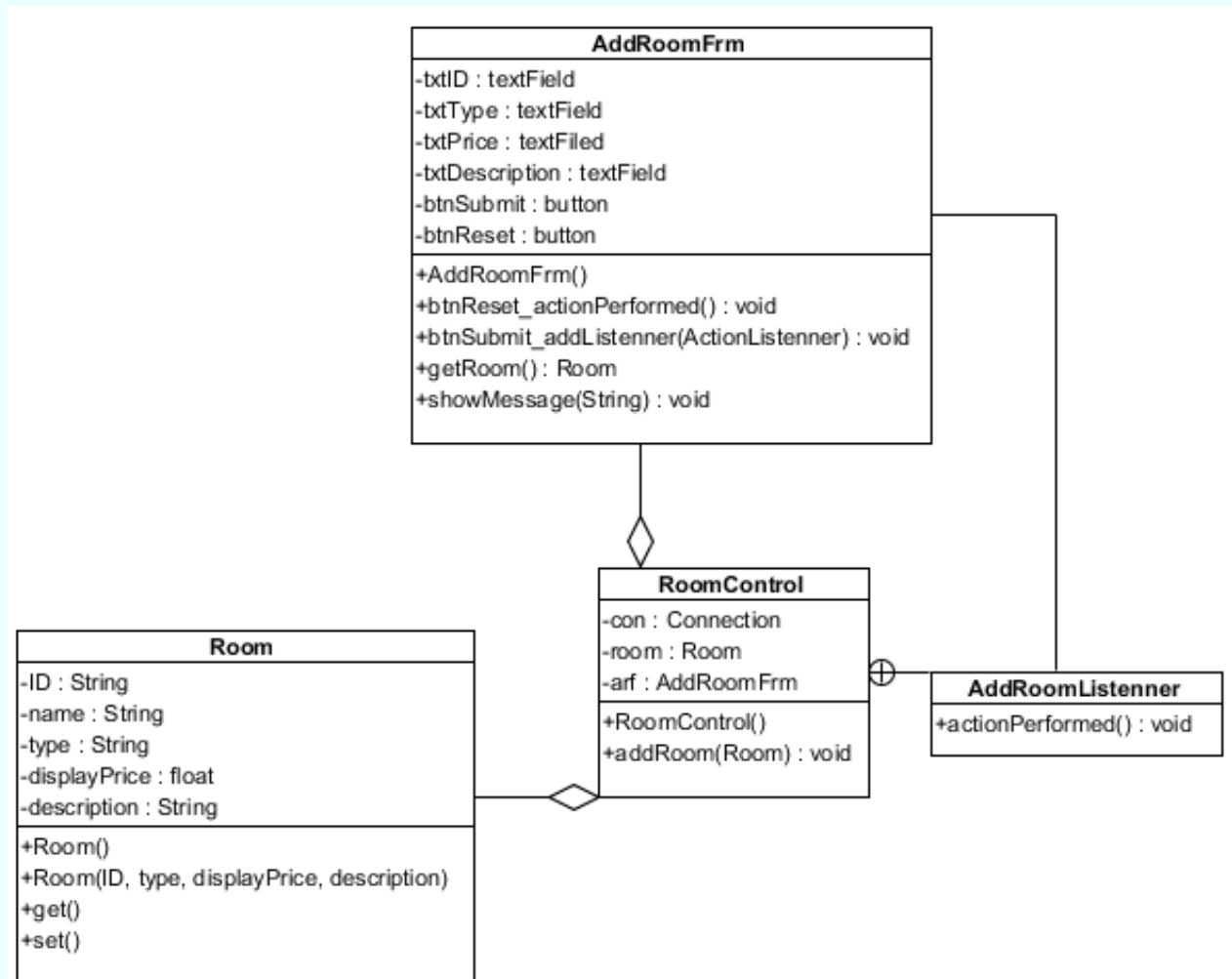


Ví dụ Giải đặt theo thiết kế MVC cải tiến

Tham khảo chi tiết tại:
<http://coderandcode.blogspot.com/2014/02/variations-of-mvc-model.html>

Sơ đồ lớp pha thiết kế

Với chức năng thêm phòng:





Lớp Room (1)

```
package mvcNew;
```

```
public class Room {  
    private String id;  
    private String name;  
    private String type;  
    private float displayPrice;  
    private String description;
```

```
    public Room() {  
        super();  
    }
```

```
    public Room(String id, String name, String type,  
        float displayPrice, String description) {  
        super();  
        this.id = id;  
        this.name = name;  
        this.type = type;  
        this.displayPrice = displayPrice;  
        this.description = description;  
    }
```



Lớp Room (2)

```
public String getId() {  
    return id;  
}  
public void setId(String id) {  
    this.id = id;  
}  
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public String getType() {  
    return type;  
}  
public void setType(String type) {  
    this.type = type;  
}  
public float getDisplayPrice() {  
    return displayPrice;  
}  
public void setDisplayPrice(float displayPrice) {  
    this.displayPrice = displayPrice;  
}  
public String getDescription() {  
    return description;  
}  
public void setDescription(String description) {  
    this.description = description;  
}
```



Lớp AddRoomFrm (1)

```
package mvcNew;
```

```
import java.awt.GridLayout;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.WindowAdapter;  
import java.awt.event.WindowEvent;
```

```
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.JTextField;
```

```
public class AddRoomFrm extends JFrame implements ActionListener{  
    private JTextField txtID;  
    private JTextField txtName;  
    private JTextField txtType;  
    private JTextField txtDisplayPrice;  
    private JTextField txtDescription;  
    private JButton btnSubmit;  
    private JButton btnReset;
```



Lớp AddRoomFrm (2)

```
public AddRoomFrm(){
    super("Room management pure-MVC");
    txtID = new JTextField(15);
    txtName = new JTextField(15);
    txtType = new JTextField(15);
    txtDisplayPrice = new JTextField(15);
    txtDescription = new JTextField(15);
    btnSubmit = new JButton("Submit");
    btnReset = new JButton("Reset");

    JPanel content = new JPanel();
    content.setLayout(new GridLayout(6,2));
    content.add(new JLabel("ID:")); content.add(txtID);
    content.add(new JLabel("Name:")); content.add(txtName);
    content.add(new JLabel("Type:")); content.add(txtType);
    content.add(new JLabel("Display price:")); content.add(txtDisplayPrice);
    content.add(new JLabel("Description:")); content.add(txtDescription);
    content.add(btnReset); content.add(btnSubmit);
    btnSubmit.addActionListener(this);
    btnReset.addActionListener(this);
    this.setContentPane(content);
    this.pack();

    this.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e){
            System.exit(0);
        }
    });
}
```



Lớp AddRoomFrm (3)

```
public void actionPerformed(ActionEvent e) {  
    JButton btn = (JButton) e.getSource();  
    if(btn.equals(btnReset)){  
        btnReset_actionPerformed();  
    }  
}
```

```
public void btnReset_actionPerformed() {  
    txtID.setText("");  
    txtName.setText("");  
    txtType.setText("");  
    txtDisplayPrice.setText("");  
    txtDescription.setText("");  
}
```



Lớp AddRoomFrm (4)

```
public Room getRoom(){
    Room room = new Room();
    room.setId(txtID.getText());
    room.setName(txtName.getText());
    room.setType(txtType.getText());
    room.setDisplayPrice(
        Float.parseFloat(txtDisplayPrice.getText()));
    room.setDescription(txtDescription.getText());
    return room;
}

public void showMessage(String msg){
    JOptionPane.showMessageDialog(this, msg);
}

public void addSubmitListener(ActionListener log) {
    btnSubmit.addActionListener(log);
}
```



Lớp RoomControl (1)

```
package mvcNew;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class RoomControl {
    private Connection con;
    private Room room;
    private AddRoomFrm arf;

    public RoomControl(){
        String dbUrl = "jdbc:mysql://localhost:3306/hotel";
        String dbClass = "com.mysql.jdbc.Driver";
        try {
            Class.forName(dbClass);
            con = DriverManager.getConnection (dbUrl, "root", "12345678");
        }catch(Exception e) {
            e.printStackTrace();
        }

        arf = new AddRoomFrm();
        arf.addSubmitListener(new AddRoomListener());
        arf.setVisible(true);
    }
}
```



Lớp RoomControl (2)

```
public void addRoom(Room room){
    String sql = "INSERT INTO tblRoom(id, name, type, displayPrice,
        description) VALUES(?,?,?,?,?)";
    try{
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, room.getId());
        ps.setString(2, room.getName());
        ps.setString(3, room.getType());
        ps.setFloat(4, room.getDisplayPrice());
        ps.setString(5, room.getDescription());

        ps.executeUpdate();
    }catch(Exception e){
        e.printStackTrace();
    }
}

class AddRoomListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        try {
            room = arf.getRoom();
            addRoom(room);
            arf.showMessage("Add room successfullly!");
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}
```

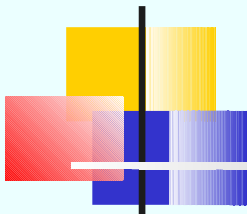



Lớp Test

```
package mvcNew;  
  
public class Test {  
  
    public static void main(String[] args) {  
        RoomControl rc = new RoomControl();  
    }  
}
```

Lưu ý trước khi chạy, phải:

- Cài đặt CSDL và bật server MySQL
- Add driver của Jdbc mysql vào library của project



Bài tập

Cài đặt theo kiến trúc đã thiết kế các modul sau:

- Chức năng sửa thông tin phòng
- Chức năng xóa thông tin phòng
- Chức năng đặt phòng
- Chức năng checkin
- Chức năng trả phòng và thanh toán



Bài tập về nhà

Cài đặt modul cá nhân theo kiến trúc đã thiết kế:

- Trình bày sơ đồ lớp đã thiết kế
- Trình bày sơ đồ cơ sở dữ liệu đã thiết kế
- Cài đặt các lớp theo đúng thiết kế
- Demo chương trình



Questions?
