

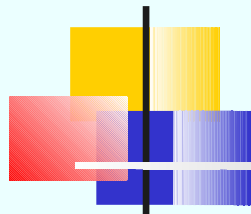


# Lập trình mạng

# Lập trình với web service

---

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

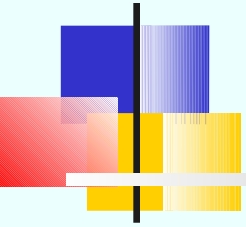


# Nội dung

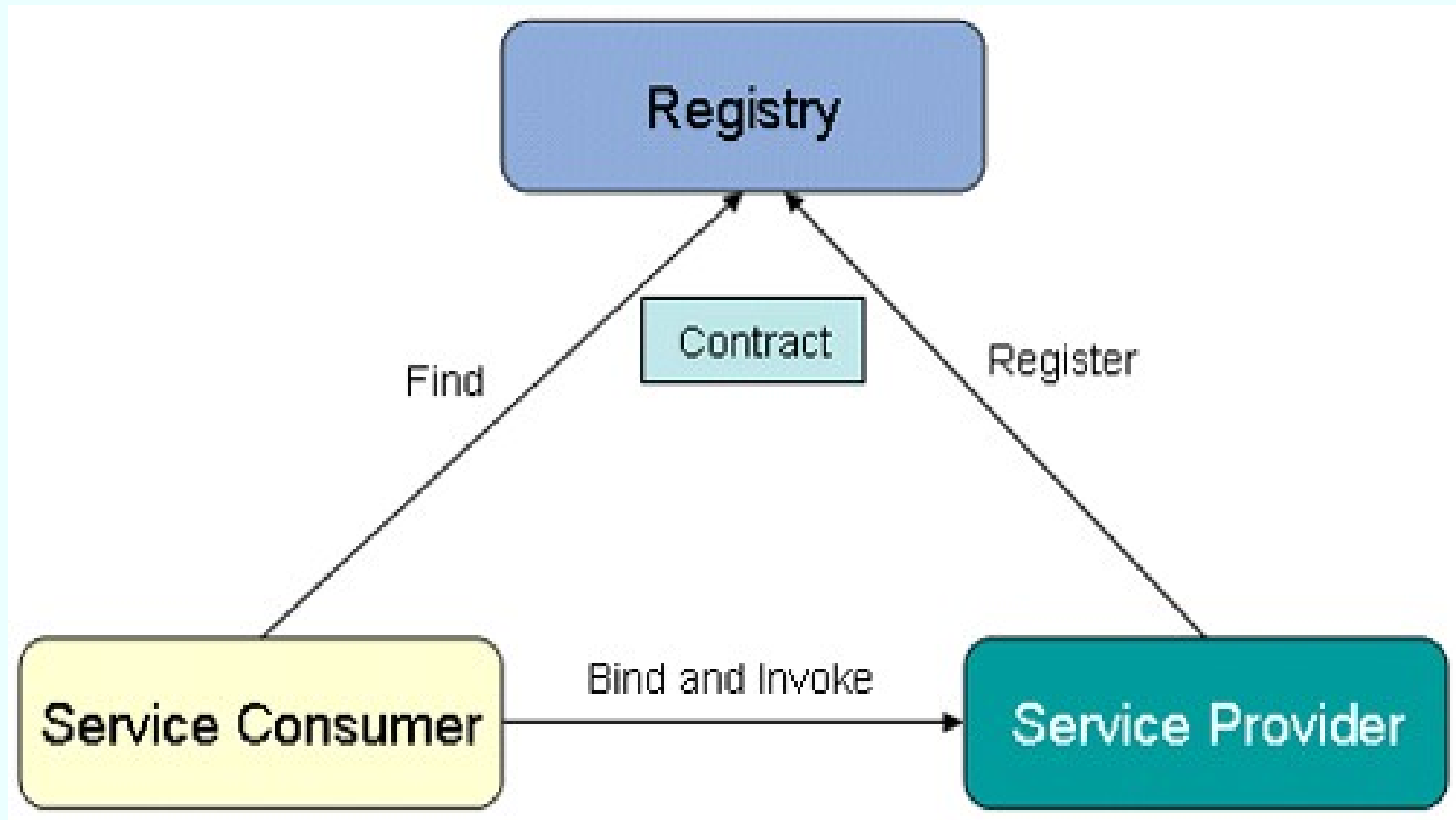
---

- Giới thiệu về công nghệ hướng dịch vụ và web service
- Cài đặt và đăng kí web service
- Sử dụng web service phía client
- Bài tập

# Công nghệ hướng dịch vụ (SOA)



# Công nghệ hướng dịch vụ (SOA)



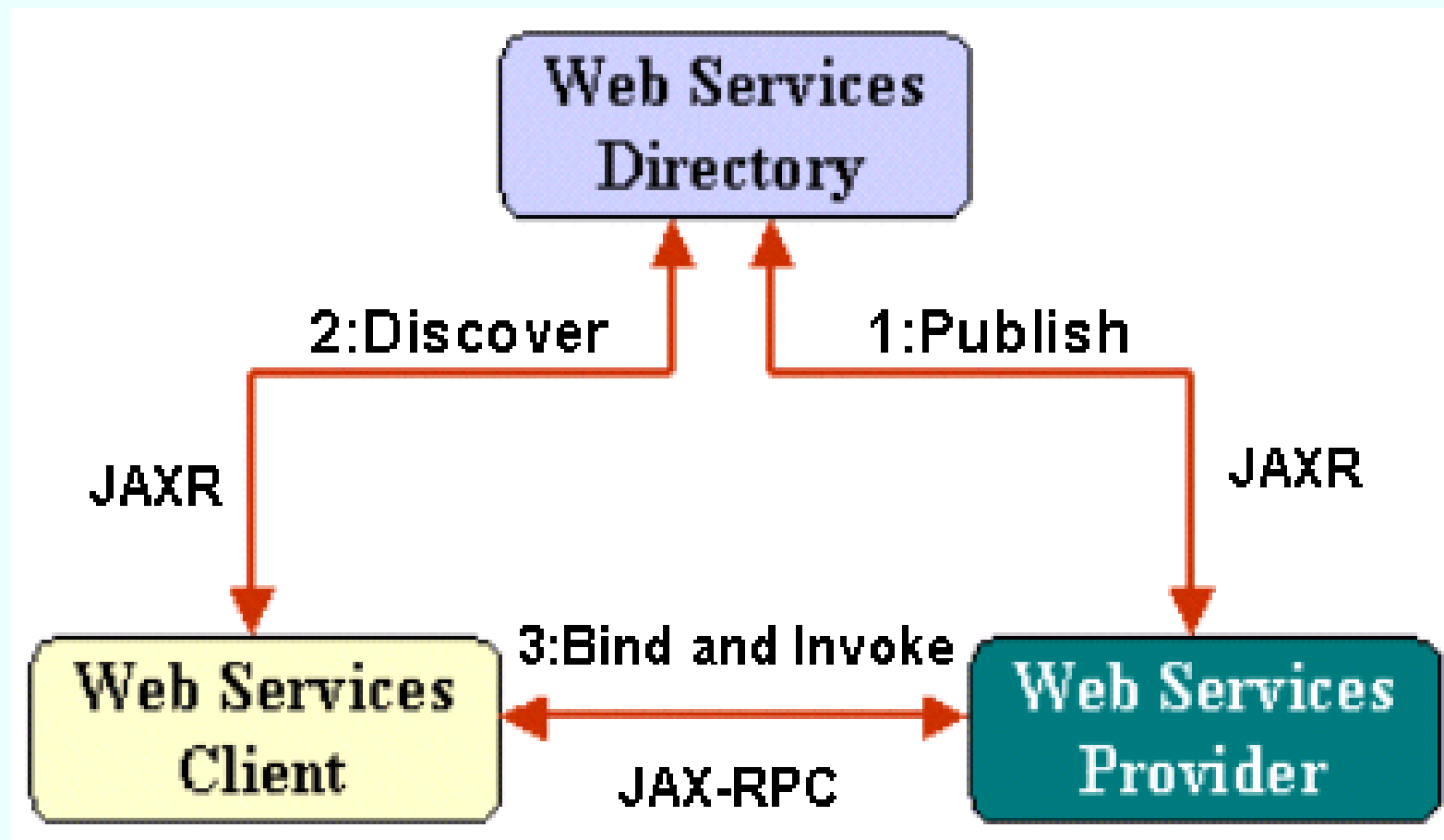
[image source: <http://www.oracle.com/technetwork>]



# Web service

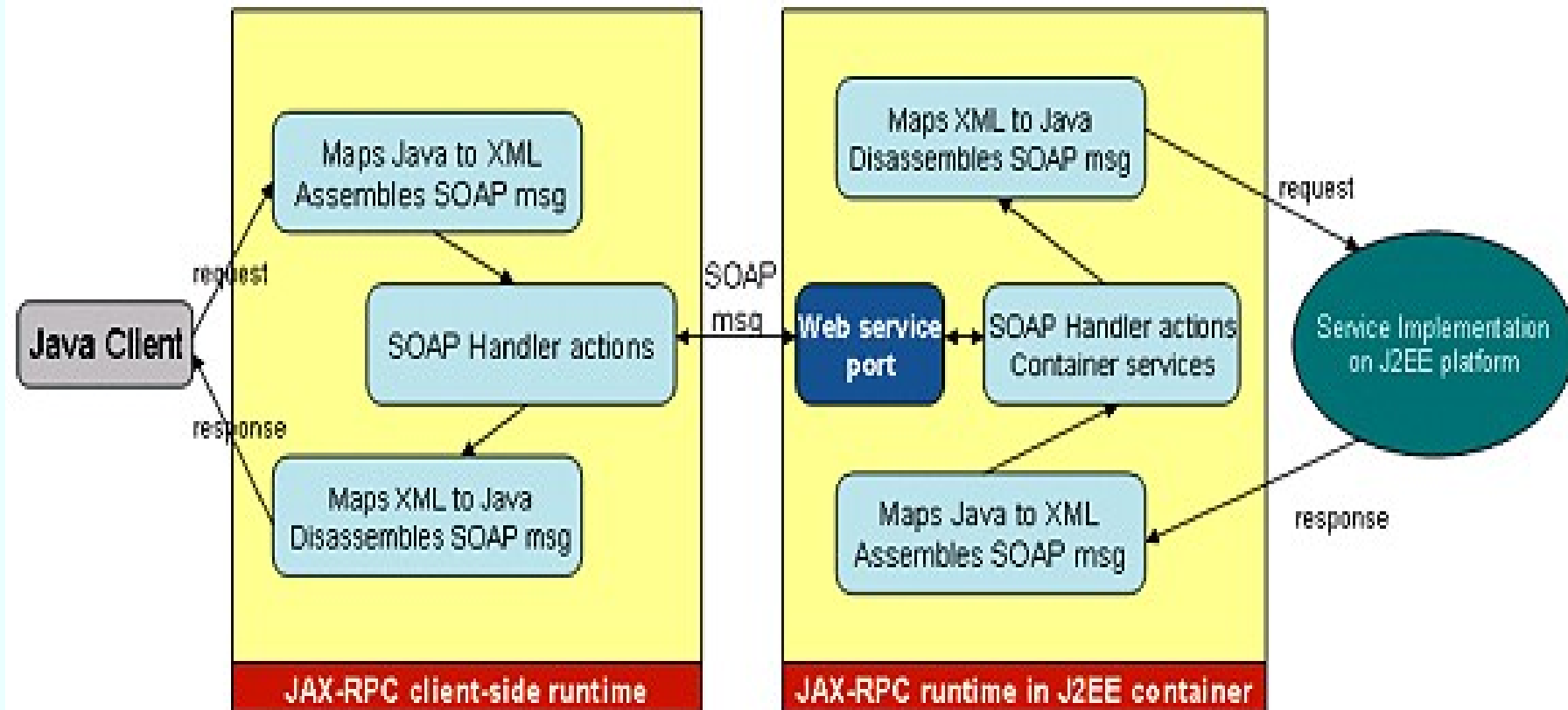
---

# Khái niệm



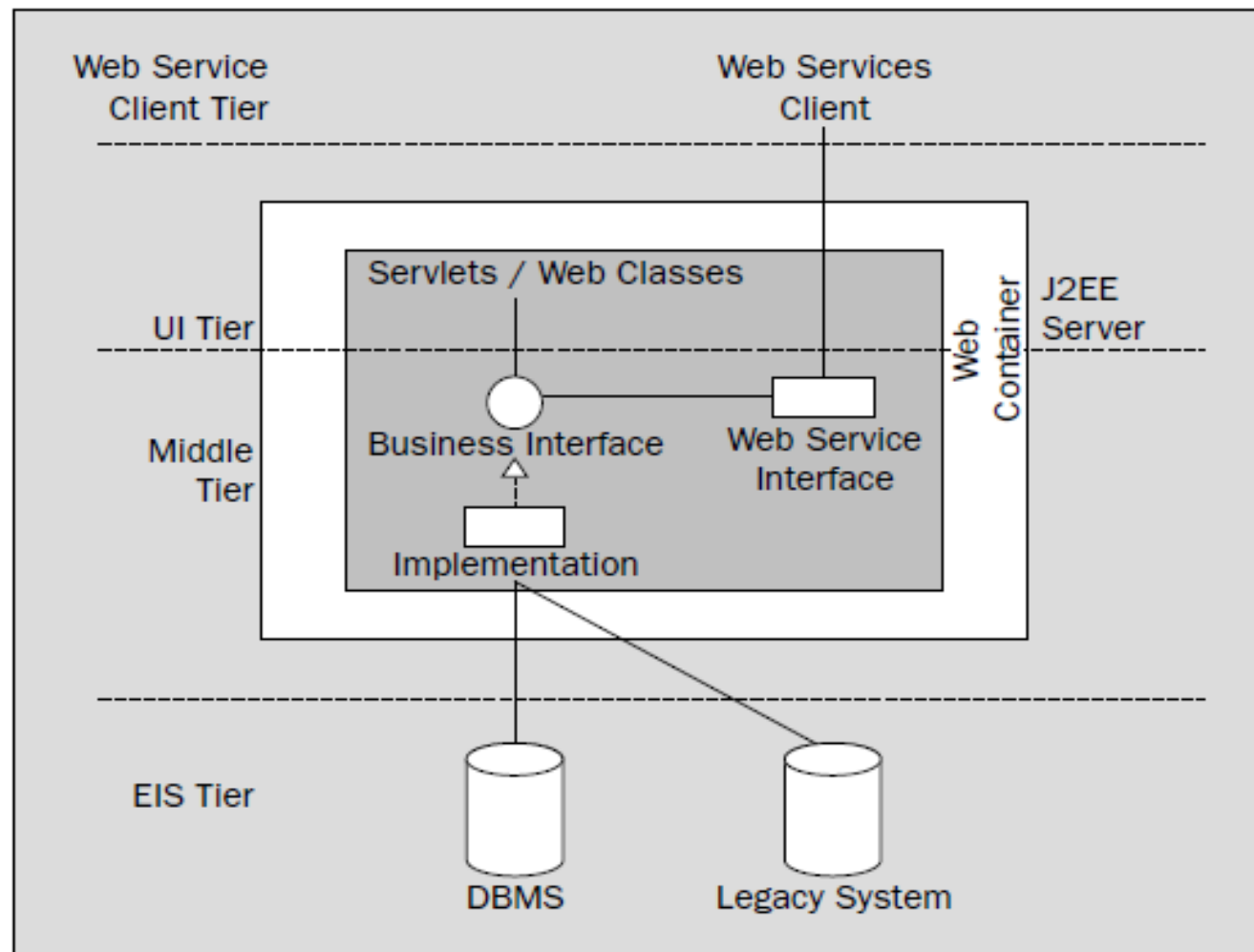
[image source: <http://www.oracle.com/technetwork>]

# WS – RPC – J2EE



[image source: <http://www.oracle.com/technetwork>]

# Kiến trúc J2EE với WS



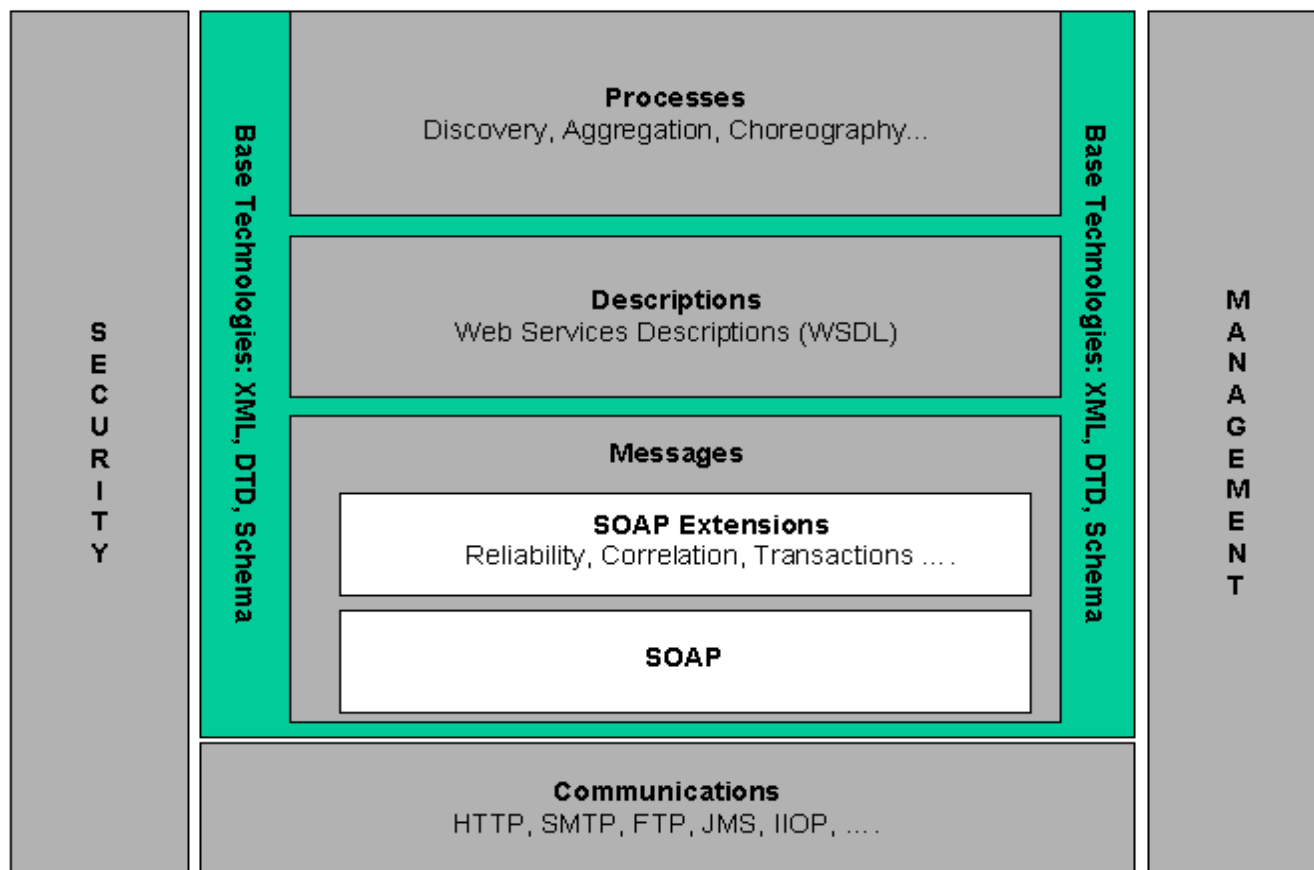
[image source: <http://media.wiley.com/>]



# Vòng đời của web service (1)

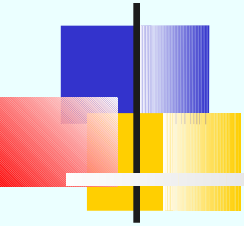


# Vòng đời của web service (2)



[image source: <http://www.w3.org/>]

# Cài đặt Web service với axis và tomcat





# Cài đặt web service (1)

---

Bước 1: Cài đặt môi trường JDK, tomcat và axis:

- Giải nén tập tin tomcat.zip (gọi là thư mục gốc tomcat)
- Giải nén tập tin xml-axis-beta1.zip (gọi là thư mục gốc axis)
- Cài đặt JDK (gọi là thư mục gốc JDK)
- Copy thư mục axis (trong thư mục gốc axis\webapps\axis) vào trong thư mục webapps (thư mục gốc tomcat\webapps)
- Vào thư mục gốc java, chọn lib, copy file tools.jar và paste vào thư mục lib của tomcat



# Cài đặt web service (2)

---

## Bước 2: Cài đặt môi trường JDK, tomcat và axis:

Variable name	Variable value
JAVA_HOME	(đường dẫn đến thư mục gốc JDK)
AXIS_HOME	(đường dẫn đến thư mục gốc axis)
CATALINA_HOME	(đường dẫn đến thư mục gốc tomcat)
DEPLOY_HOME	(Đường dẫn đến thư mục gốc tomcat)\webapps\axis\WEB-INF\lib



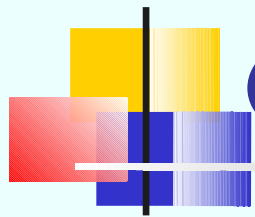
# Cài đặt web service (3)

---

Bước 3: Tạo file java chứa dịch vụ (cộng hai số nguyên)

```
public class Calculator{  
    public String add(double x, double y){  
        return (x+y)+"";  
    }  
}
```

Sau đó, lưu với định dạng .jws tại thư mục axis (Thư mục gốc tomcat)\webapps\axis



# Cài đặt web service (4)

---

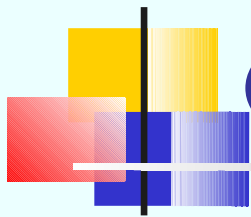
Bước 4: kiểm tra lại trên server tomcat

Khởi động Tomcat server (bằng cách chạy tập tin thư mục gốc tomcat\bin\startup.bat)

Sau đó truy cập địa chỉ:

<http://localhost:8080/axis/Calculator.jws?wsdl>

Nếu thành công, chúng ta sẽ thấy được nội dung của tập tin wsdl của WS như sau:



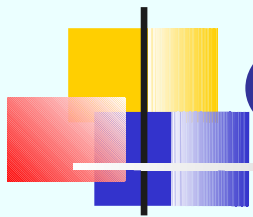
# Cài đặt web service (5)

---

## Bước 4: kiểm tra lại trên server tomcat

```
definitions
targetNamespace="http://localhost:8080/axis/Calculator.jws">
<message name="AddRequest">
<part name="x" type="xsd:int"/>
<part name="y" type="xsd:int"/>
</message>
<message name="AddResponse">
<part name="return" type="xsd:string"/>
</message>
<portType name="Calculator">
<operation name="Add" parameterOrder="x y">
<input message="intf:AddRequest"/>
<output message="intf:AddResponse"/>
</operation>
</portType>
```





# Cài đặt web service (6)

---

## Bước 4: kiểm tra lại trên server tomcat

```
<binding name="CalculatorSoapBinding" type="intf:Calculator">
<wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="Add">
<wsdlsoap:operation soapAction=""/>
<input>
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/axis/Calculator.jws" use="encoded"/>
</input>
<output>
<wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://localhost:8080/axis/Calculator.jws" use="encoded"/>
</output>
</operation>
</binding>
```



# Cài đặt web service (7)

---

## Bước 4: kiểm tra lại trên server tomcat

```
<service name="CalculatorService">  
<port binding="intf:CalculatorSoapBinding" name="Calculator">  
<wsdlsoap:address location="http://localhost:8080/axis/Calculator.jws"/>  
</port>  
</service>  
</definitions>
```



# Cài đặt web service (8)

---

## Bước 5: cài đặt ứng dụng client

```
public class CalculatorClient {
    private int x;
    private int y;
    public void Calculator(String methodName,int a,int b){
        String endpointURL =
            "http://localhost:8080/axis/Calculator.jws";
        Service service=new Service();
        try{
            Call call=(Call)service.createCall();
            call.setTargetEndpointAddress(new java.net.URL(endpointURL));
            call.setOperationName(methodName);
            call.addParameter("x",XMLType.XSD_DOUBLE,
                ParameterMode.PARAM_MODE_IN);
            call.addParameter("y", XMLType.XSD_DOUBLE,
                ParameterMode.PARAM_MODE_IN);
            call.setReturnType(XMLType.XSD_STRING);
            String result=(String)call.invoke(new Object[]{a,b});
            System.out.println("Ket qua : "+result);
        }catch(Exception ex){
            System.out.println("Loi: "+ex);
        }
    }
}
```



# Cài đặt web service (9)

---

## Bước 5: cài đặt ứng dụng client

```
public static void main(String args[]){  
    CalculatorClient cal=new CalculatorClient();  
    Scanner input=new Scanner(System.in);  
    int x,y;  
  
    x = input.nextInt();  
    y = input.nextInt();  
    cal.Calculator("Add", x,y);  
}
```

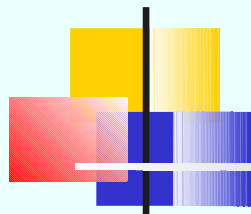


# Cài đặt web service (10)

---

Bước 6: thực thi ứng dụng client

- chạy tomcat server
- thiết lập các biến classpath
- mở cửa sổ chạy lên xem các thông điệp SOAP:  
`java org.apache.axis.utils.tcpmon 9000 localhost 8080`
- chạy client

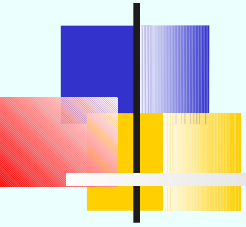


# Bài tập

---

- Cài đặt theo mô hình web service cho bài toán tìm USCLN
- Cài đặt theo mô hình web service cho bài toán giải phương trình bậc hai

# Ứng dụng các phép toán cơ bản dùng web service



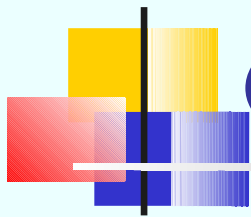


# Bài toán

---

- Định nghĩa một web service cung cấp chức năng tính toán 4 phép toán cơ bản: nhân, chia, cộng, trừ hai số double
- Cài đặt ứng dụng sử dụng web service này để tính toán





# Cài đặt web service (1)

---

Tạo file jws chứa 4 dịch vụ (cộng, trừ, nhân, chia)

```
public class Calculators{  
    public String Add(double x, double y){  
        return (x+y)+"";  
    }  
    public String Sub(double x, double y){  
        return (x-y)+"";  
    }  
    public String Mul(double x, double y){  
        return (x*y)+"";  
    }  
    public String Div(double x, double y){  
        return (x/y)+"";  
    }  
}
```

Sau đó, lưu với định dạng Calculators.jws tại thư mục axis  
(C:\Webservice\jakarta-tomcat-4.0.6\webapps\axis)



## Cài đặt web service (2)

---

Kiểm tra lại trên server tomcat

Khởi động Tomcat server (bằng cách chạy tập tin  
C:\Webservice\jakarta-tomcat-4.0.6\bin\startup.bat)

Sau đó truy cập địa chỉ:

<http://localhost:8080/axis/Calculators.jws?wsdl>

Nếu thành công, chúng ta sẽ thấy được nội dung của tập tin  
wsdl của WS như sau:



# Cài đặt web service (3)

```
- <definitions targetNamespace="http://localhost:8080/axis/Calculators.jws">
  - <message name="SubResponse">
    <part name="return" type="xsd:string"/>
  </message>
  - <message name="MulRequest">
    <part name="x" type="xsd:double"/>
    <part name="y" type="xsd:double"/>
  </message>
  - <message name="DivResponse">
    <part name="return" type="xsd:string"/>
  </message>
  - <message name="AddRequest">
    <part name="x" type="xsd:double"/>
    <part name="y" type="xsd:double"/>
  </message>
  - <message name="DivRequest">
    <part name="x" type="xsd:double"/>
    <part name="y" type="xsd:double"/>
  </message>
  - <message name="AddResponse">
    <part name="return" type="xsd:string"/>
  </message>
  - <message name="SubRequest">
    <part name="x" type="xsd:double"/>
    <part name="y" type="xsd:double"/>
  </message>
  - <message name="MulResponse">
    <part name="return" type="xsd:string"/>
  </message>
```



# Cài đặt web service (4)

---

```
- <portType name="Calculators">
  - <operation name="Add" parameterOrder="x y">
    <input message="intf.AddRequest"/>
    <output message="intf.AddResponse"/>
  </operation>
  - <operation name="Div" parameterOrder="x y">
    <input message="intf.DivRequest"/>
    <output message="intf.DivResponse"/>
  </operation>
  - <operation name="Mul" parameterOrder="x y">
    <input message="intf.MulRequest"/>
    <output message="intf.MulResponse"/>
  </operation>
  - <operation name="Sub" parameterOrder="x y">
    <input message="intf.SubRequest"/>
    <output message="intf.SubResponse"/>
  </operation>
</portType>
```



# Cài đặt web service (5)

---

```
- <binding name="CalculatorsSoapBinding" type="intf:Calculators">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
- <operation name="Add">
  <wsdlsoap:operation soapAction=""/>
  - <input>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/Calculators.jws" use="encoded"/>
  </input>
  - <output>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/Calculators.jws" use="encoded"/>
  </output>
</operation>
- <operation name="Div">
  <wsdlsoap:operation soapAction=""/>
  - <input>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/Calculators.jws" use="encoded"/>
  </input>
  - <output>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/Calculators.jws" use="encoded"/>
  </output>
</operation>
```

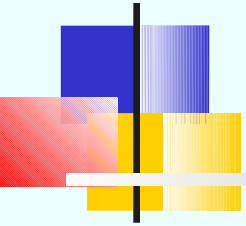


# Cài đặt web service (6)

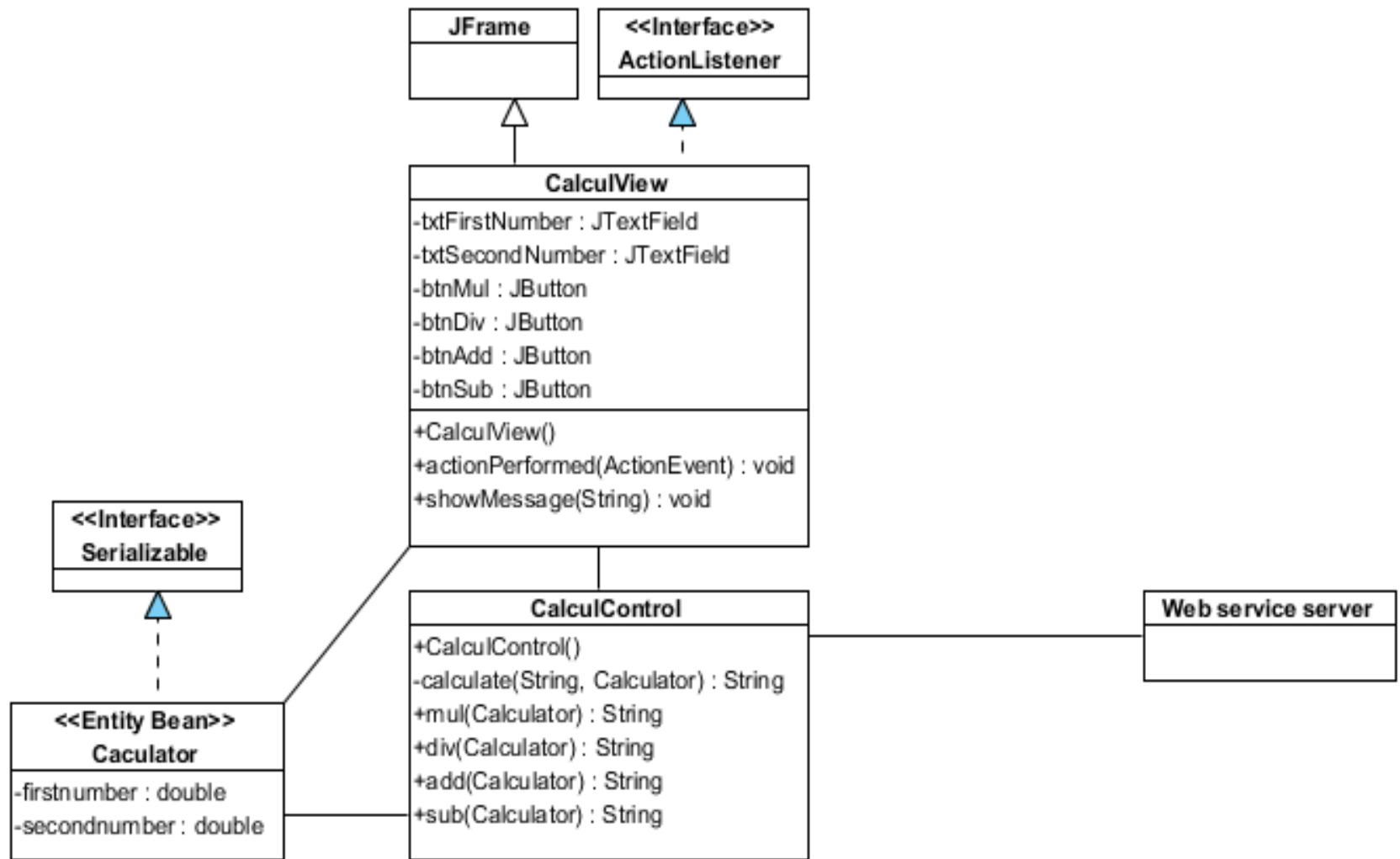
---

```
- <operation name="Mul">
  <wsdlsoap:operation soapAction=""/>
  - <input>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/Calculators.jws" use="encoded"/>
  </input>
  - <output>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/Calculators.jws" use="encoded"/>
  </output>
</operation>
- <operation name="Sub">
  <wsdlsoap:operation soapAction=""/>
  - <input>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/Calculators.jws" use="encoded"/>
  </input>
  - <output>
    <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/Calculators.jws" use="encoded"/>
  </output>
</operation>
</binding>
- <service name="CalculatorsService">
  - <port binding="intf:CalculatorsSoapBinding" name="Calculators">
    <wsdlsoap:address location="http://localhost:8080/axis/Calculators.jws"/>
  </port>
</service>
```

# Cài đặt ứng dụng theo mô hình MVC cổ điển



# Sơ đồ quan hệ các lớp







# Lớp Calculator

---

```
import java.io.Serializable;

public class Calculator implements Serializable{
    private double firstnumber;
    private double secondnumber;

    public Calculator(){
    }
    public Calculator(double a, double b){
        firstnumber = a;
        secondnumber = b;
    }

    public double getFirstnumber() {
        return firstnumber;
    }

    public void setFirstnumber(double firstnumber) {
        this.firstnumber = firstnumber;
    }

    public double getSecondnumber() {
        return secondnumber;
    }

    public void setSecondnumber(double secondnumber) {
        this.secondnumber = secondnumber;
    }
}
```



# Lớp CalculView (1)

---

```
import java.awt.FlowLayout;
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 CalculView extends JFrame implements ActionListener{
    private JTextField txtFirstnumber;
    private JTextField txtSecondnumber;
    private JButton btnAdd, btnSub, btnMul, btnDiv;
```



# Lớp CalculView (2)

---

```
public CalculView(){
    super("Calculator using web service");

    txtFirstnumber = new JTextField(8);
    txtSecondnumber = new JTextField(8);
    btnAdd = new JButton("Add");
    btnSub = new JButton("Subvision");
    btnMul = new JButton("Multiple");
    btnDiv = new JButton("Division");

    JPanel content = new JPanel();
    content.setLayout(new FlowLayout());
    content.add(new JLabel("First number:"));
    content.add(txtFirstnumber);
    content.add(new JLabel("Second number:"));
    content.add(txtSecondnumber);
    content.add(btnAdd);
    content.add(btnSub);
    content.add(btnMul);
    content.add(btnDiv);

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



# Lớp CalculView (3)

---

```
public void actionPerformed(ActionEvent e) {
    Calculator cal = null;
    try{
        double a = Double.parseDouble(txtFirstnumber.getText());
        double b = Double.parseDouble(txtSecondnumber.getText());
        cal = new Calculator(a,b);
    }catch(Exception ex){
        ex.printStackTrace();
    }
    CalculControl cc = new CalculControl();
    if(e.getSource().equals(btnAdd)){
        showMessage("Addition of " + cal.getFirstnumber() + " and "
            + cal.getSecondnumber() + " is: " + cc.add(cal));
    }else if(e.getSource().equals(btnSub)){
        showMessage("Subvision of " + cal.getFirstnumber() + " and "
            + cal.getSecondnumber() + " is: " + cc.sub(cal));
    }else if(e.getSource().equals(btnMul)){
        showMessage("Multiple of " + cal.getFirstnumber() + " and "
            + cal.getSecondnumber() + " is: " + cc.multiple(cal));
    }else if(e.getSource().equals(btnDiv)){
        showMessage("Division of " + cal.getFirstnumber() + " and "
            + cal.getSecondnumber() + " is: " + cc.div(cal));
    }
}

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



# Lớp CalculControl (1)

---

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import org.apache.axis.client.Service;
import org.apache.axis.client.Call;
import org.apache.axis.encoding.XMLType;
import javax.xml.rpc.ParameterMode;

public class CalculControl {

    public CalculControl(){

    }

}
```



# Lớp CalculControl (2)

---

```
public String multiple(Calculator cal){  
    String result = null;  
    try {  
        result = calculate("Mul", cal);  
    } catch (Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
    return result;  
}
```

```
public String div(Calculator cal){  
    String result = null;  
    try {  
        result = calculate("Div", cal);  
    } catch (Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
    return result;  
}
```



# Lớp CalculControl (3)

---

```
public String add(Calculator cal){  
    String result = null;  
    try {  
        result = calculate("Add", cal);  
    } catch (Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
    return result;  
}
```

```
public String sub(Calculator cal){  
    String result = null;  
    try {  
        result = calculate("Sub", cal);  
    } catch (Exception e) {  
        e.printStackTrace();  
        return null;  
    }  
    return result;  
}
```



# Lớp CalculControl (4)

---

```
private String calculate(String methodName, Calculator cal){
    String result = "";
    String endpointURL = "http://localhost:8080/axis/Calculators.jws";
    Service service = new Service();
    try{
        Call call = (Call)service.createCall();
        call.setTargetEndpointAddress(new java.net.URL(endpointURL));
        call.setOperationName(methodName);
        call.addParameter("a", XMLType.XSD_DOUBLE,
                           ParameterMode.PARAM_MODE_IN);
        call.addParameter("b", XMLType.XSD_DOUBLE,
                           ParameterMode.PARAM_MODE_IN);
        call.setReturnType(XMLType.XSD_STRING);
        Result = (String)call.invoke(new Object[]
                                     {cal.getFirstnumber(),cal.getSecondnumber()});
    }catch(Exception ex){
        ex.printStackTrace();
    }
    return result;
}
```



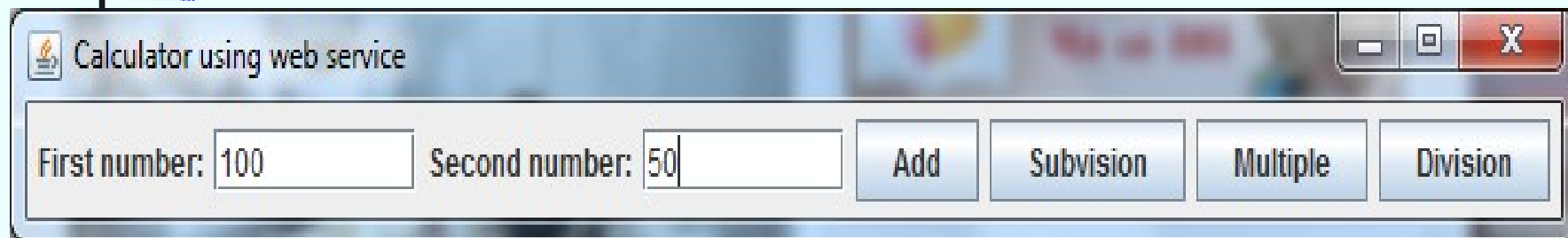


# Lớp Run

---

```
public class Run {  
  
    public static void main(String[] args) {  
        CalculView view = new CalculView();  
        view.setVisible(true);  
    }  
  
}
```

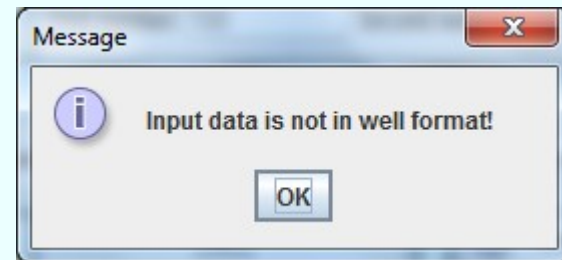
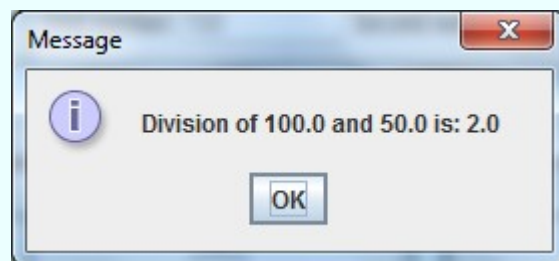
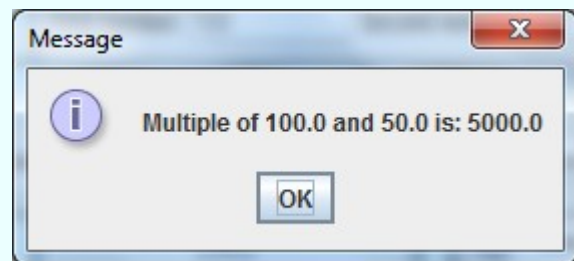
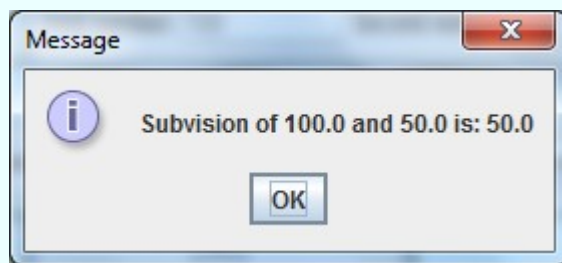
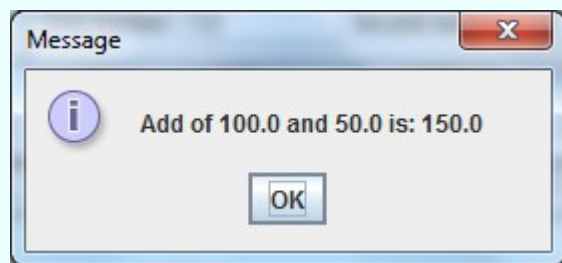
# Kết quả

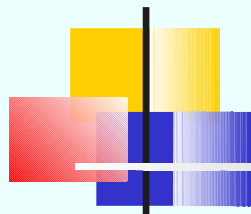


Calculator using web service

First number:  Second number:

Add Subvision Multiple Division



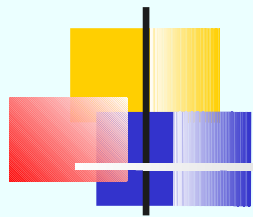


# Bài tập (1)

---

Cài đặt đúng mô hình MVC cổ điển cho bài toán quản lí người dùng theo WS:

- Server chứa CSDL về người dùng, có bảng tbluser chứa các cột: id, username, password, address, birthday, sex, description
- Client có giao diện nhập thông tin đăng kí người dùng mới
- Sau khi nhập thông tin và click submit, client gửi thông tin đăng kí đến server
- Server kiểm tra xem có trùng username không, nếu không thì thêm vào CSDL và báo thành công, nếu trùng thì thông báo trùng cho client
- Client nhận được thông tin sẽ hiển thị yêu cầu người dùng nhập lại khi trùng, hoặc báo đăng kí thành công.



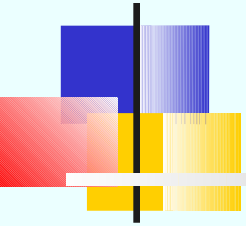
# Bài tập (2)

---

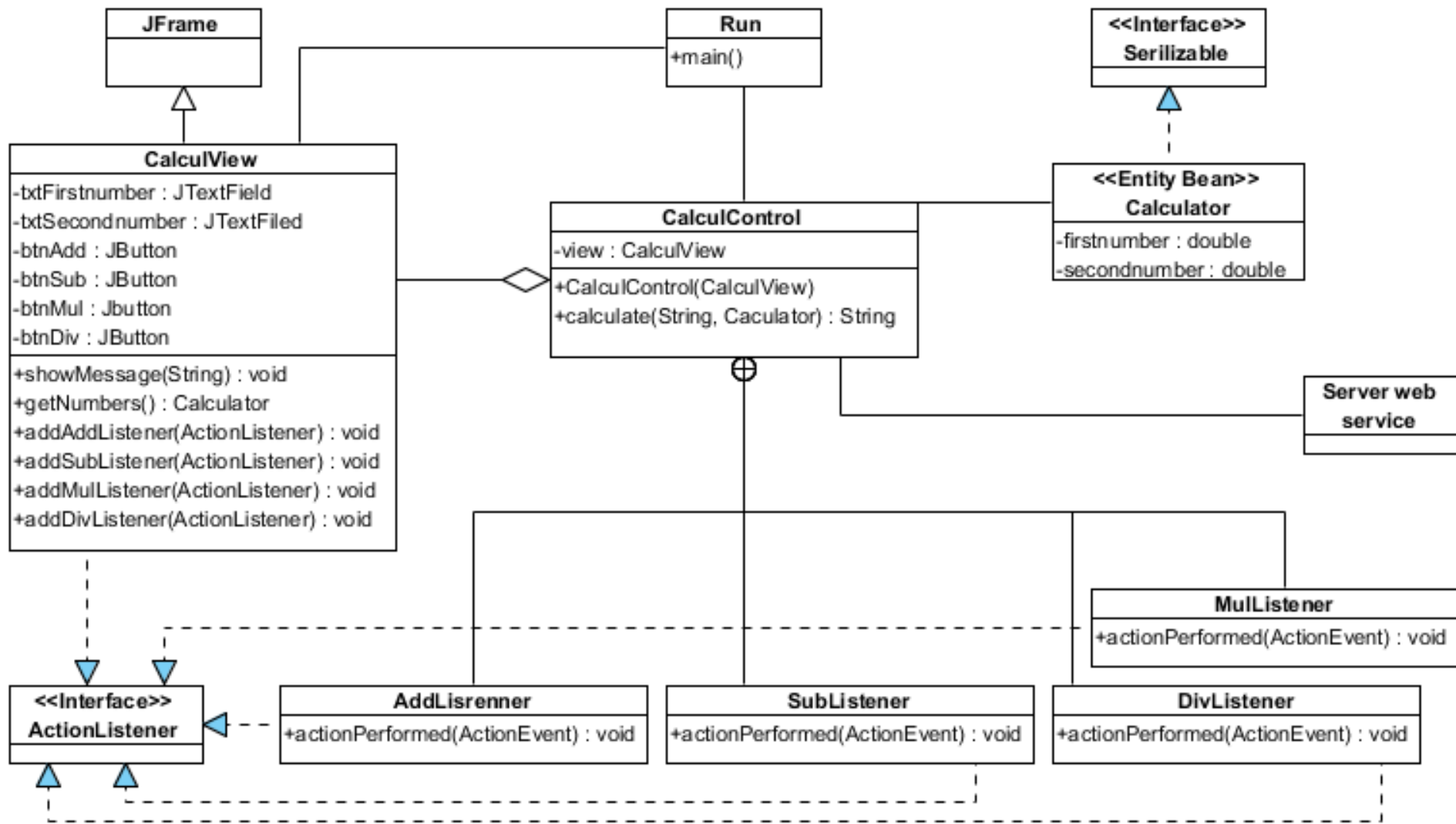
Cài đặt đúng mô hình MVC cổ điển cho bài toán quản lí người dùng theo WS:

- Server chứa CSDL về người dùng, có bảng tbluser chứa các cột: id, username, password, address, birthday, sex, description
- Client có giao diện nhập thông tin tìm kiếm người dùng theo tên
- Sau khi nhập thông tin và click submit, client gửi thông tin tìm kiếm đến server
- Server tìm kiếm thông tin người dùng từ CSDL và trả kết quả về cho client
- Client nhận được thông tin sẽ hiển thị danh sách người dùng có tên chứa từ khóa đã nhập.

# Cài đặt ứng dụng theo mô hình MVC cải tiến



# Sơ đồ quan hệ các lớp





# Lớp Calculator

---

```
import java.io.Serializable;

public class Calculator implements Serializable{
    private double firstnumber;
    private double secondnumber;

    public Calculator(){
    }
    public Calculator(double a, double b){
        firstnumber = a;
        secondnumber = b;
    }

    public double getFirstnumber() {
        return firstnumber;
    }

    public void setFirstnumber(double firstnumber) {
        this.firstnumber = firstnumber;
    }

    public double getSecondnumber() {
        return secondnumber;
    }

    public void setSecondnumber(double secondnumber) {
        this.secondnumber = secondnumber;
    }
}
```



# Lớp CalculView (1)

---

```
import java.awt.FlowLayout;
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 CalculView extends JFrame implements ActionListener{
    private JTextField txtFirstnumber;
    private JTextField txtSecondnumber;
    private JButton btnAdd, btnSub, btnMul, btnDiv;
```





# Lớp CalculView (2)

---

```
public CalculView(){
    super("Calculator using web service");

    txtFirstnumber = new JTextField(8);
    txtSecondnumber = new JTextField(8);
    btnAdd = new JButton("Add");
    btnSub = new JButton("Subvision");
    btnMul = new JButton("Multiple");
    btnDiv = new JButton("Division");

    JPanel content = new JPanel();
    content.setLayout(new FlowLayout());
    content.add(new JLabel("First number:"));
    content.add(txtFirstnumber);
    content.add(new JLabel("Second number:"));
    content.add(txtSecondnumber);
    content.add(btnAdd);
    content.add(btnSub);
    content.add(btnMul);
    content.add(btnDiv);

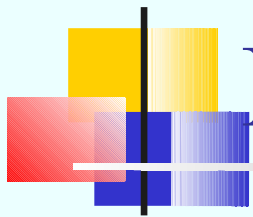
    this.setContentPane(content);
    this.pack();
    this.addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent e){
            System.exit(0);
        }
    });
}
```



# Lớp CalculView (3)

---

```
public void actionPerformed(ActionEvent e) {  
  
    public Calculator getNumbers(){  
        Calculator cal = null;  
        try{  
            double a = Double.parseDouble(txtFirstnumber.getText());  
            double b = Double.parseDouble(txtSecondnumber.getText());  
            cal = new Calculator(a,b);  
        }catch(Exception e){  
  
        }  
        return cal;  
    }  
  
    public void showMessage(String msg){  
        JOptionPane.showMessageDialog(this, msg);  
    }  
}
```



# Lớp CalculView (4)

---

```
public void addAddListener(ActionListener log) {  
    btnAdd.addActionListener(log);  
}  
  
public void addSubListener(ActionListener log) {  
    btnSub.addActionListener(log);  
}  
  
public void addMulListener(ActionListener log) {  
    btnMul.addActionListener(log);  
}  
  
public void addDivListener(ActionListener log) {  
    btnDiv.addActionListener(log);  
}  
}
```



# Lớp CalculControl (1)

---

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import org.apache.axis.client.Service;
import org.apache.axis.client.Call;
import org.apache.axis.encoding.XMLType;
import javax.xml.rpc.ParameterMode;

public class CalculControl {
    private CalculView view;

    public CalculControl(CalculView view){
        this.view = view;
        view.addAddListener(new AddListener());
        view.addSubListener(new SubListener());
        view.addMulListener(new MulListener());
        view.addDivListener(new DivListener());
    }
}
```



# Lớp CalculControl (2)

---

```
class AddListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        try {
            Calculator cal = view.getNumbers();
            if(cal != null){
                String result = calculate("Add", cal);
                view.showMessageDialog("Add of " + cal.getFirstnumber()
                    + " and " + cal.getSecondnumber()
                    + " is: " + result);
            }else{
                view.showMessageDialog("Input data is not in well format!");
            }
        } catch (Exception ex) {
            view.showMessageDialog(ex.getStackTrace().toString());
        }
    }
}
```



# Lớp CalculControl (3)

---

```
class SubListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        try {
            Calculator cal = view.getNumbers();
            if(cal != null){
                String result = calculate("Sub", cal);
                view.showMessageDialog("Subvision of "
                    + cal.getFirstnumber() + " and "
                    + cal.getSecondnumber() + " is: " + result);
            } else{
                view.showMessageDialog("Input data is not in well format!");
            }
        } catch (Exception ex) {
            view.showMessageDialog(ex.getStackTrace().toString());
        }
    }
}
```



# Lớp CalculControl (4)

---

```
class MulListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        try {
            Calculator cal = view.getNumbers();
            if(cal != null){
                String result = calculate("Mul", cal);
                view.showMessageDialog("Multiple of "
                    + cal.getFirstnumber() + " and " +
                    cal.getSecondnumber() + " is: " + result);
            }else{
                view.showMessageDialog("Input data is not in well format!");
            }
        } catch (Exception ex) {
            view.showMessageDialog(ex.getStackTrace().toString());
        }
    }
}
```



# Lớp CalculControl (5)

---

```
class DivListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        try {
            Calculator cal = view.getNumbers();
            if((cal != null)&&(cal.getSecondnumber() != 0)){
                String result = calculate("Div", cal);
                view.showMessageDialog("Division of " +
                    cal.getFirstnumber() + " and " +
                    cal.getSecondnumber() + " is: " + result);
            }else{
                view.showMessageDialog("Input data is not in well format!");
            }
        } catch (Exception ex) {
            view.showMessageDialog(ex.getStackTrace().toString());
        }
    }
}
```





# Lớp CalculControl (6)

---

```
private String calculate(String methodName, Calculator cal){
    String result = "";
    String endpointURL = "http://localhost:8080/axis/Calculators.jws";
    Service service = new Service();
    try{
        Call call = (Call)service.createCall();
        call.setTargetEndpointAddress(new java.net.URL(endpointURL));
        call.setOperationName(methodName);
        call.addParameter("a", XMLType.XSD_DOUBLE,
                           ParameterMode.PARAM_MODE_IN);
        call.addParameter("b", XMLType.XSD_DOUBLE,
                           ParameterMode.PARAM_MODE_IN);
        call.setReturnType(XMLType.XSD_STRING);
        Result = (String)call.invoke(new Object[]
                                     {cal.getFirstnumber(),cal.getSecondnumber()});
    }catch(Exception ex){
        view.showMessageDialog("Ex: "+ex);
    }
    return result;
}
```

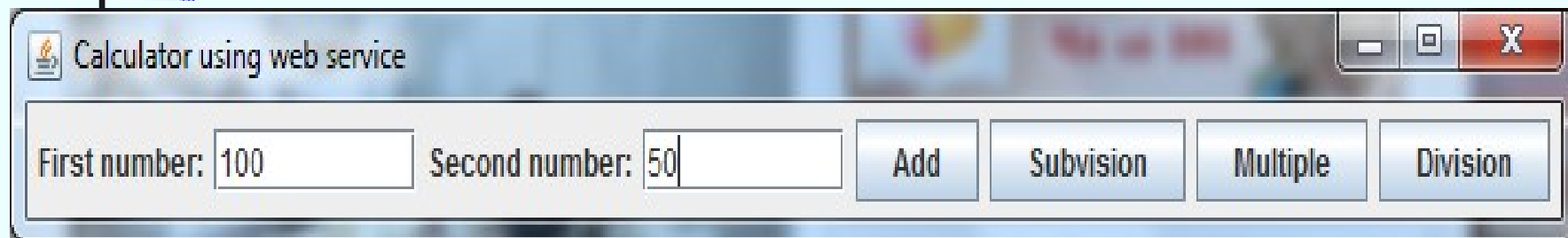


# Lớp Run

---

```
public class Run {  
  
    public static void main(String[] args) {  
        CalculView view = new CalculView();  
        CalculControl control = new CalculControl(view);  
        view.setVisible(true);  
    }  
}
```

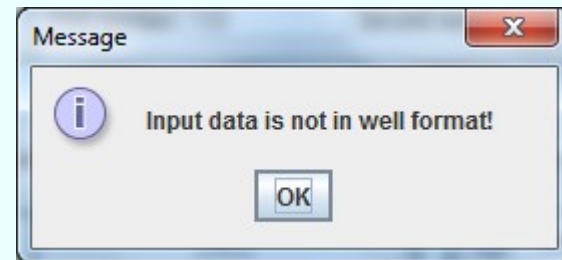
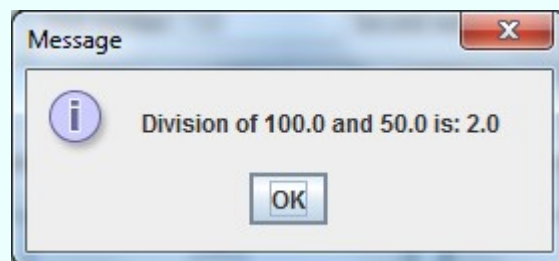
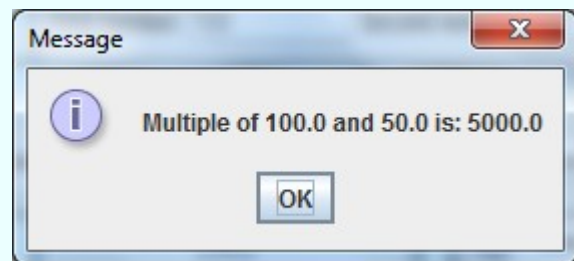
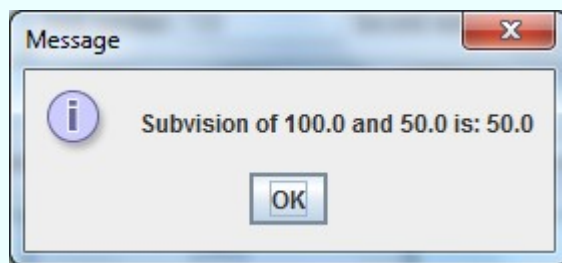
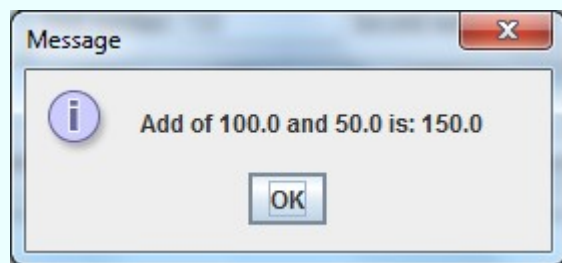
# Kết quả

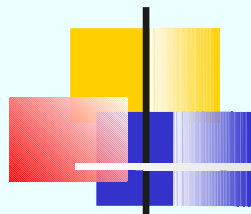


Calculator using web service

First number:  Second number:

Add Subvision Multiple Division



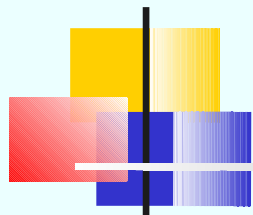


# Bài tập (1)

---

Cài đặt đúng mô hình MVC cải tiến cho bài toán quản lí người dùng theo WS:

- Server chứa CSDL về người dùng, có bảng tbluser chứa các cột: id, username, password, address, birthday, sex, description
- Client có giao diện nhập thông tin đăng kí người dùng mới
- Sau khi nhập thông tin và click submit, client gửi thông tin đăng kí đến server
- Server kiểm tra xem có trùng username không, nếu không thì thêm vào CSDL và báo thành công, nếu trùng thì thông báo trùng cho client
- Client nhận được thông tin sẽ hiển thị yêu cầu người dùng nhập lại khi trùng, hoặc báo đăng kí thành công.



## Bài tập (2)

---

Cài đặt đúng mô hình MVC cải tiến cho bài toán quản lí người dùng theo WS:

- Server chứa CSDL về người dùng, có bảng tbluser chứa các cột: id, username, password, address, birthday, sex, description
- Client có giao diện nhập thông tin tìm kiếm người dùng theo tên
- Sau khi nhập thông tin và click submit, client gửi thông tin tìm kiếm đến server
- Server tìm kiếm thông tin người dùng từ CSDL và trả kết quả về cho client
- Client nhận được thông tin sẽ hiển thị danh sách người dùng có tên chứa từ khóa đã nhập.



# Questions?

---