PROGRAM 1:

Index.html

```
<html>
   <center>
   <head>
      <title>BANKING PORTAL</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <h1>BANKING</h1>
   </head>
   <body>
      <form action="transact">
         Enter Amount: <input type="text" name="trans_val" value = '0'/><br>
         Select Type of Transaction:<br>
         <input type="radio" name="transaction" value="deposit">Deposit<br>
         <input type="radio" name="transaction" value="withdraw">Withdraw<br>
         <input type="radio" name="transaction" value="balance">View Balance<br>
         <input type="submit" value="TRANSACT"/>
      </form>
   </body>
</center>
</html>
```

BankTransact.java

```
import javax.ejb.Stateful;

@Stateful
public class BankTransact implements BankTransactRemote{

   int balance = 10000;
   public int deposit(int amount) {
      balance = balance + amount;
      return balance;
   }
```

```
    public int withdraw(int amount) {
        balance = balance - amount;
        return balance;
    }

    public int getBalance() {
        return balance;
    }

}
```

BankTransactRemote.java

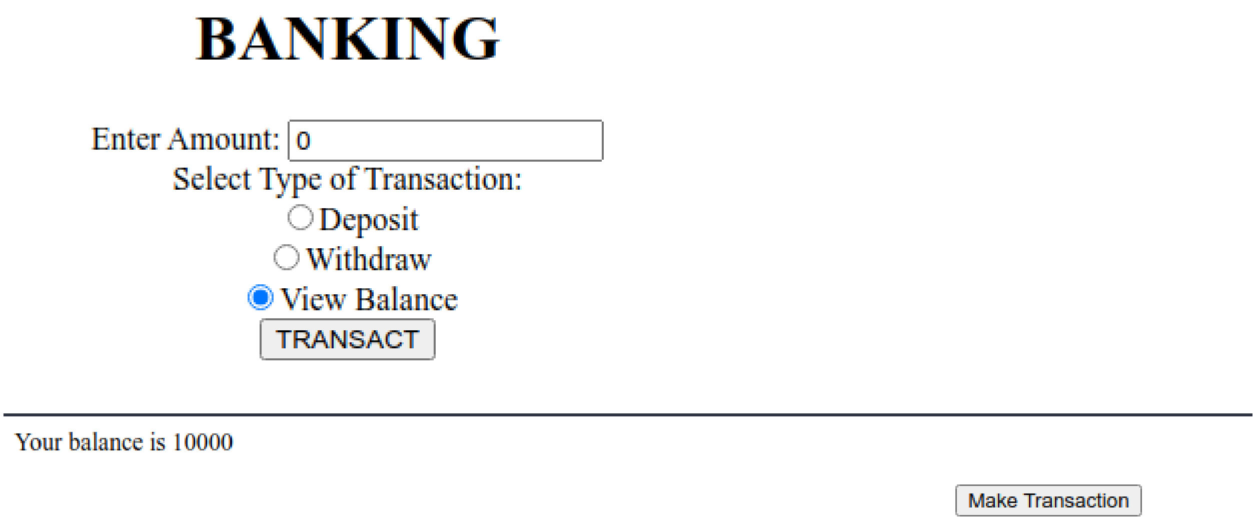
```
import javax.ejb.Remote;

/**
 *
 * @author shankar
 */
@Remote
public interface BankTransactRemote {
    public int deposit(int amount);
    public int withdraw(int amount);
    public int getBalance() ;
}
```

Transact.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
public class transact extends HttpServlet {

    BankTransact bankTransact = lookupBankTransactBean();


    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String s = request.getParameter("transaction");

            int amount = Integer.parseInt(request.getParameter("trans_val"));

            if(s.equals("deposit")){
                out.println(amount+" Successfully Deposited. Your balance is
"+bankTransact.deposit(amount));

            }
            else if(s.equals("withdraw")){
                out.println(amount+"Successfully Withdrawn. Your balance is
"+bankTransact.withdraw(amount));
            }
            else if(s.equals("balance")){
                out.println("Your balance is "+bankTransact.withdraw(0));
            }
              out.println("<center><br> <a href = 'index.html'> <input type = 'button' value = 'Make
Transaction'> </a></center>");

        }
    }

}
```

Output:



BANKING

Enter Amount: 0

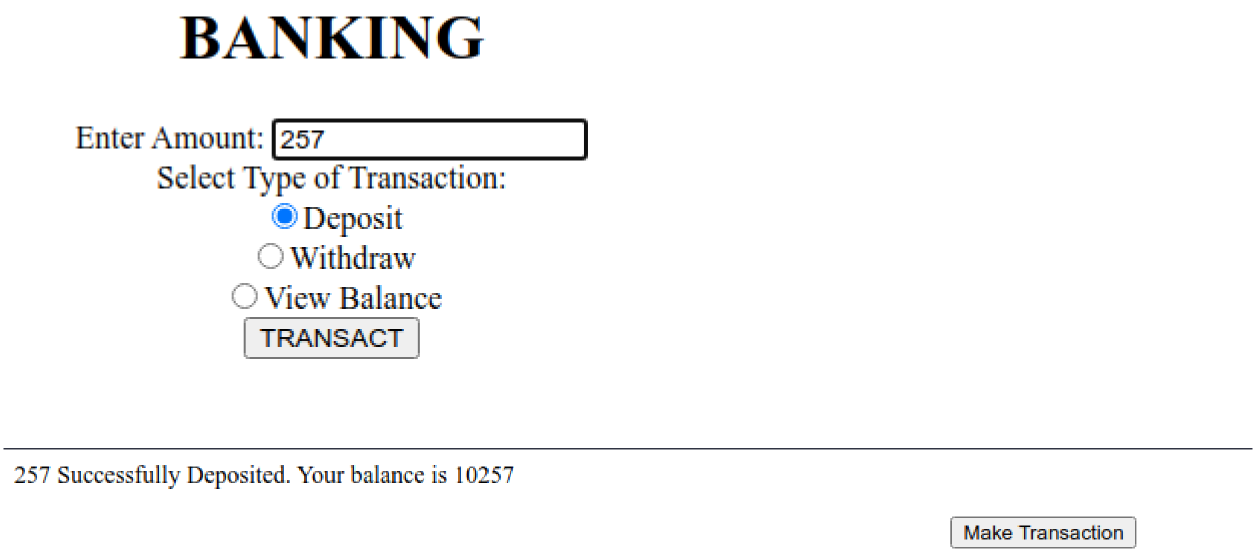
Select Type of Transaction:

○ Deposit

○ Withdraw

● View Balance

Your balance is 10000



BANKING

Enter Amount: 257

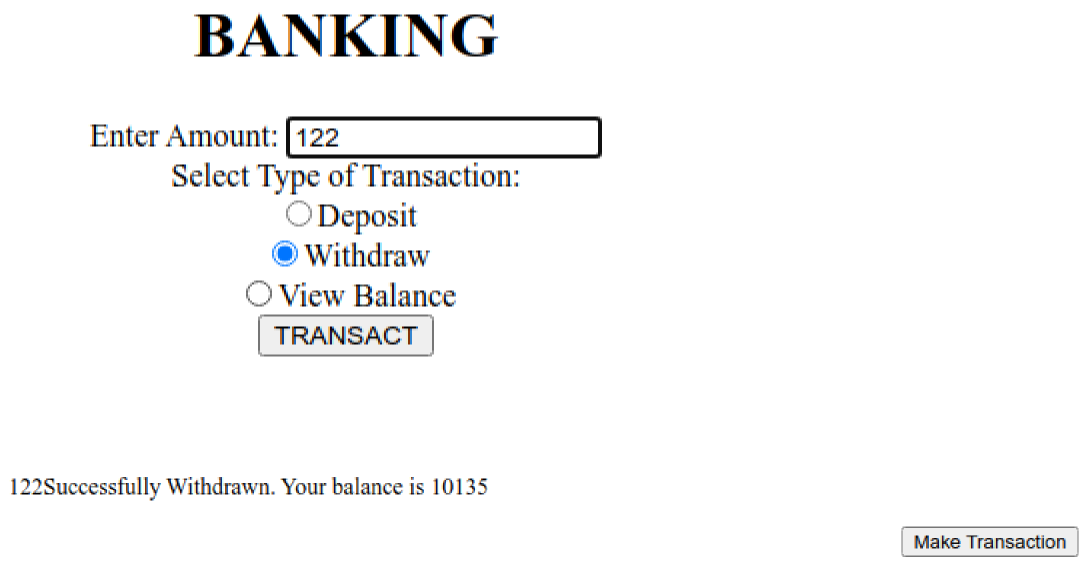
Select Type of Transaction:

● Deposit

○ Withdraw

○ View Balance

257 Successfully Deposited. Your balance is 10257



BANKING

Enter Amount: 122

Select Type of Transaction:

○ Deposit

● Withdraw

○ View Balance

```
122Successfully Withdrawn. Your balance is 10135
```

Program 2:

CalculatorWS.java

```
package org.me.calculator;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.ejb.Stateless;
/**
 *
 * @author shankar
 */
@WebService(serviceName = "CalculatorWS")
@Stateless()
public class CalculatorWS {
   /**
    * Web service operation
    */
   @WebMethod(operationName = "add")
```

```
public double add(@WebParam(name = "i") double i, @WebParam(name = "j") double j) {
    //TODO write your implementation code here:
    double k = i + j;
    return k;
}

/**
 * Web service operation
 */
@WebMethod(operationName = "sub")
public double sub(@WebParam(name = "i") double i, @WebParam(name = "j") double j) {
    //TODO write your implementation code here:
    return (i-j);
}

@WebMethod(operationName = "mul")
public double mul(@WebParam(name = "i") double i, @WebParam(name = "j") double j) {
    //TODO write your implementation code here:
    return (i*j);
}

@WebMethod(operationName = "div")
public double div(@WebParam(name = "i") double i, @WebParam(name = "j") double j) {
    //TODO write your implementation code here:
    return (1.0 * i/j);
}

/**
 * Web service operation
 */
@WebMethod(operationName = "root")
public double root(@WebParam(name = "i") double i) {
    //TODO write your implementation code here:
    return Math.sqrt(i);
}

/**
 * Web service operation
 */
@WebMethod(operationName = "prime")
public String prime(@WebParam(name = "i") int i) {

    for(int j = 2; j<i/2; j++){
        if( i%j == 0){
                return "Not prime";
            }
        }
        return "Prime";
    }
}
```

Factorial.java

```
package org.me.calculator;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.ejb.Stateless;

/**
 *
 * @author shankar
 */
@WebService(serviceName = "Factorial")
@Stateless()
public class Factorial {

    /**
     * Web service operation
     */
    @WebMethod(operationName = "factorial")
    public int factorial(@WebParam(name = "i") int i) {
        //TODO write your implementation code here:
        int fact = 1;

        for(int j = 1; j< i; j++){
            fact = fact * j;
        }
        return fact;
    }

    /**
     * This is a sample web service operation
     */

}
```

WSDL - CALCULATORWS

```
<definitions
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://calculator.me.org/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://calculator.me.org/" name="CalculatorWS">
<types>
<xsd:schema>
<xsd:import namespace="http://calculator.me.org/"
schemaLocation="http://localhost:8080/CalculatorWS/CalculatorWS?xsd=1"/>
</xsd:schema>
</types>
<message name="prime">
<part name="parameters" element="tns:prime"/>
</message>
<message name="primeResponse">
<part name="parameters" element="tns:primeResponse"/>
</message>
<message name="div">
<part name="parameters" element="tns:div"/>
</message>
<message name="divResponse">
<part name="parameters" element="tns:divResponse"/>
</message>
<message name="mul">
<part name="parameters" element="tns:mul"/>
</message>
<message name="mulResponse">
<part name="parameters" element="tns:mulResponse"/>
</message>
<message name="sub">
<part name="parameters" element="tns:sub"/>
</message>
<message name="subResponse">
<part name="parameters" element="tns:subResponse"/>
</message>
<message name="add">
```

```xml
<part name="parameters" element="tns:add"/>
</message>
<message name="addResponse">
<part name="parameters" element="tns:addResponse"/>
</message>
<message name="root">
<part name="parameters" element="tns:root"/>
</message>
<message name="rootResponse">
<part name="parameters" element="tns:rootResponse"/>
</message>
<portType name="CalculatorWS">
<operation name="prime">
<input wsam:Action="http://calculator.me.org/CalculatorWS/primeRequest"
message="tns:prime"/>
<output wsam:Action="http://calculator.me.org/CalculatorWS/primeResponse"
message="tns:primeResponse"/>
</operation>
<operation name="div">
<input wsam:Action="http://calculator.me.org/CalculatorWS/divRequest" message="tns:div"/>
<output wsam:Action="http://calculator.me.org/CalculatorWS/divResponse"
message="tns:divResponse"/>
</operation>
<operation name="mul">
<input wsam:Action="http://calculator.me.org/CalculatorWS/mulRequest" message="tns:mul"/>
<output wsam:Action="http://calculator.me.org/CalculatorWS/mulResponse"
message="tns:mulResponse"/>
</operation>
<operation name="sub">
<input wsam:Action="http://calculator.me.org/CalculatorWS/subRequest" message="tns:sub"/>
<output wsam:Action="http://calculator.me.org/CalculatorWS/subResponse"
message="tns:subResponse"/>
</operation>
<operation name="add">
<input wsam:Action="http://calculator.me.org/CalculatorWS/addRequest" message="tns:add"/>
<output wsam:Action="http://calculator.me.org/CalculatorWS/addResponse"
message="tns:addResponse"/>
</operation>
<operation name="root">
<input wsam:Action="http://calculator.me.org/CalculatorWS/rootRequest" message="tns:root"/>
<output wsam:Action="http://calculator.me.org/CalculatorWS/rootResponse"
message="tns:rootResponse"/>
</operation>
</portType>
```

```
<binding name="CalculatorWSPortBinding" type="tns:CalculatorWS">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="prime">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="div">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="mul">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="sub">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="add">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
```

```
<soap:body use="literal"/>
</output>
</operation>
<operation name="root">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="CalculatorWS">
<port name="CalculatorWSPort" binding="tns:CalculatorWSPortBinding">
<soap:address location="http://localhost:8080/CalculatorWS/CalculatorWS"/>
</port>
</service>
</definitions>
```

WSDL - FACTORIAL

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--  Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.4.0
(wsit240-7e98ff4; 2017-08-03T21:19:54+0200) JAXWS-RI/2.3.0 JAXWS-API/2.3.0
JAXB-RI/2.3.0 JAXB-API/2.3.0 svn-revision#unknown.  -->
<!--  Generated by JAX-WS RI (http://javaee.github.io/metro-jax-ws). RI's version is Metro/2.4.0
(wsit240-7e98ff4; 2017-08-03T21:19:54+0200) JAXWS-RI/2.3.0 JAXWS-API/2.3.0
JAXB-RI/2.3.0 JAXB-API/2.3.0 svn-revision#unknown.  -->
<definitions
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
" xmlns:wsp="http://www.w3.org/ns/ws-policy"
xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://calculator.me.org/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://calculator.me.org/" name="Factorial">
<types>
<xsd:schema>
<xsd:import namespace="http://calculator.me.org/"
schemaLocation="http://localhost:8080/Factorial/Factorial?xsd=1"/>
```

```
</xsd:schema>
</types>
<message name="factorial">
<part name="parameters" element="tns:factorial"/>
</message>
<message name="factorialResponse">
<part name="parameters" element="tns:factorialResponse"/>
</message>
<portType name="Factorial">
<operation name="factorial">
<input wsam:Action="http://calculator.me.org/Factorial/factorialRequest"
message="tns:factorial"/>
<output wsam:Action="http://calculator.me.org/Factorial/factorialResponse"
message="tns:factorialResponse"/>
</operation>
</portType>
<binding name="FactorialPortBinding" type="tns:Factorial">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="factorial">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="Factorial">
<port name="FactorialPort" binding="tns:FactorialPortBinding">
<soap:address location="http://localhost:8080/Factorial/Factorial"/>
</port>
</service>
</definitions>
```

Calculator.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.mycompany.calculatorclient;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author shankar
 */
public class Calculator extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try ( PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet Calculator</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<hr>Calculator <br>");


            out.println("<form action = '/CalculatorClient/calc'>");
            out.println("<input type = 'text' name = 'i'>");
            out.println("<input type = 'text' name = 'j'> <br>");
            out.println("<input type = 'submit' name = 'op' value = 'add'>");
            out.println("<input type = 'submit' name = 'op' value = 'sub'>");
            out.println("<input type = 'submit' name = 'op' value = 'mul'>");
            out.println("<input type = 'submit' name = 'op' value = 'div'> <br>");
            out.println("<input type = 'submit' name = 'op' value = 'root'>");
            out.println("<input type = 'submit' name = 'op' value = 'prime'>");
            out.println("<input type = 'submit' name = 'op' value = 'fact'>");
            out.println("</form><hr>");

            out.println("</body>");
            out.println("</html>");
        }
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
```

```
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```

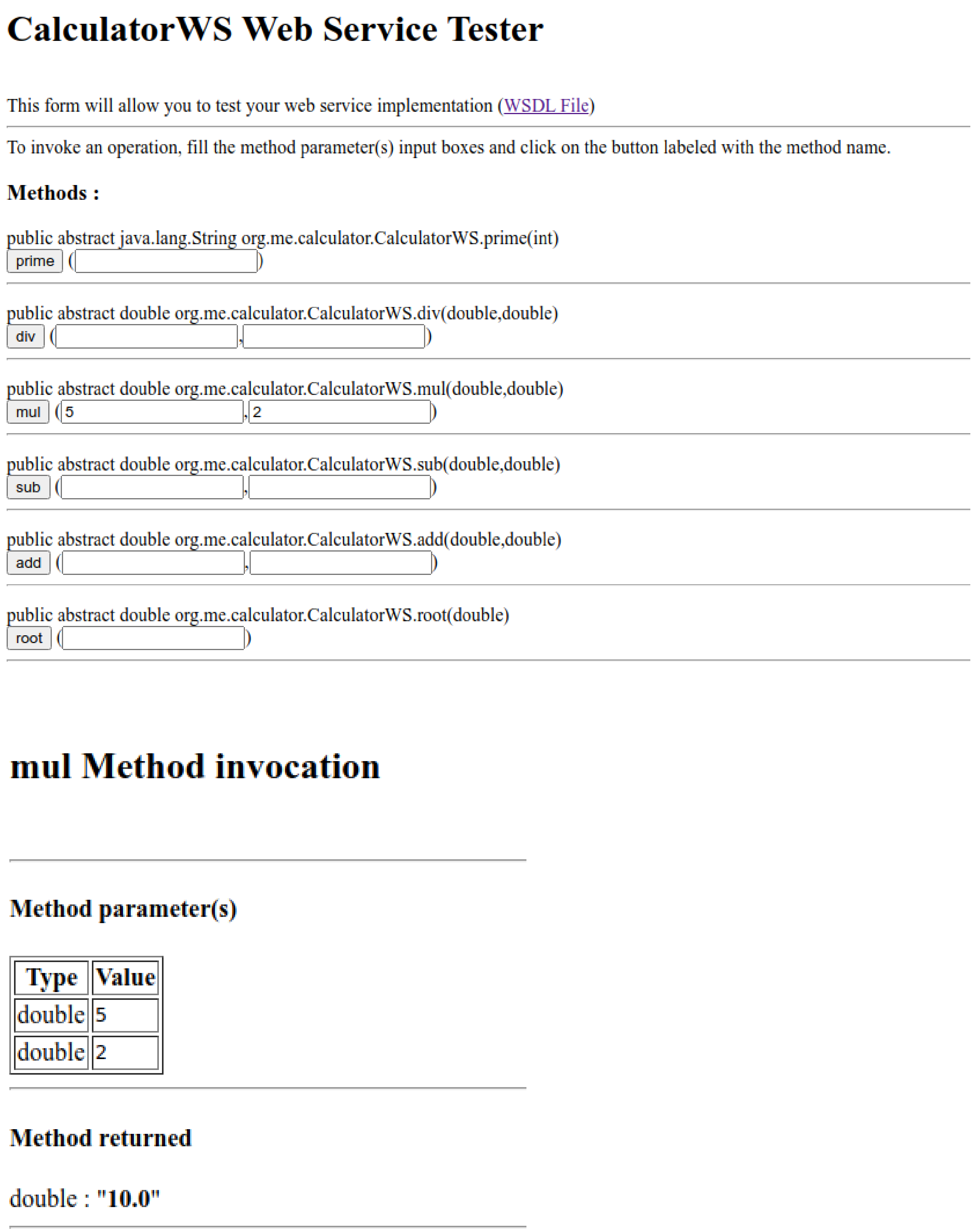
Calc.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.mycompany.calculatorclient;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.ws.WebServiceRef;
import org.me.calculator.CalculatorWS_Service;
import org.me.calculator.Factorial_Service;

/**
 *
 * @author shankar
 */
public class calc extends HttpServlet {

    @WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost_8080/Factorial/Factorial.wsdl")
    private Factorial_Service service_1;

    @WebServiceRef(wsdlLocation =
"WEB-INF/wsdl/localhost_8080/CalculatorWS/CalculatorWS.wsdl")
    private CalculatorWS_Service service;
```

```
/**
 * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
 * methods.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try ( PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet calc</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet calc at " + request.getContextPath() + "</h1>");
        //int i = Integer.parseInt((String) request.getParameter("i"));
        //int j = Integer.parseInt((String) request.getParameter("j"));
        double i = 0.0;
        String i_temp = request.getParameter("i");
        String j_temp = request.getParameter("j");

        if (i_temp.isEmpty()) {
            i = Double.parseDouble(j_temp);
        } else {
            i = Double.parseDouble(i_temp);
        }
        String op = (String) request.getParameter("op");

        double j = Double.parseDouble(j_temp);
        //out.println(i);
        //out.println(j);
        out.println(op);

        switch (op) {
            case "add":

                try { // Call Web Service Operation
                org.me.calculator.CalculatorWS port = service.getCalculatorWSPort();
    // TODO initialize WS operation arguments here
    // TODO process result here
    double result = port.add(i, j);
    out.println("Result = " + result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
break;
case "sub":
    try { // Call Web Service Operation
    org.me.calculator.CalculatorWS port = service.getCalculatorWSPort();
    // TODO initialize WS operation arguments here
    // TODO process result here
    double result = port.sub(i, j);
    out.println("Result = " + result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
break;
case "div":
    try { // Call Web Service Operation
    org.me.calculator.CalculatorWS port = service.getCalculatorWSPort();
    // TODO initialize WS operation arguments here
    // TODO process result here
    double result = port.div(i, j);
    out.println("Result = " + result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
break;
case "mul":
    try { // Call Web Service Operation
    org.me.calculator.CalculatorWS port = service.getCalculatorWSPort();
    // TODO initialize WS operation arguments here
    // TODO process result here
    double result = port.mul(i, j);
    out.println("Result = " + result);
} catch (Exception ex) {
    // TODO handle custom exceptions here
}
break;
case "root":
    try { // Call Web Service Operation
    org.me.calculator.CalculatorWS port = service.getCalculatorWSPort();
```

```
                    // TODO initialize WS operation arguments here
                    // TODO process result here
                    double result = port.root(i);
                    out.println("Result = " + result);
                } catch (Exception ex) {
                    // TODO handle custom exceptions here
                }
                break;

                case "prime":
                    try { // Call Web Service Operation
                    org.me.calculator.CalculatorWS port = service.getCalculatorWSPort();
                    // TODO initialize WS operation arguments here
                    // TODO process result here
                    java.lang.String result = port.prime((int) i);
                    out.println("Result = " + result);
                } catch (Exception ex) {
                    // TODO handle custom exceptions here
                }
                break;

                case "fact":
                    try { // Call Web Service Operation
                        org.me.calculator.Factorial port = service_1.getFactorialPort();
                        // TODO initialize WS operation arguments here
                        // TODO process result here
                        int result = port.factorial((int) i);
                        out.println("Result = "+result);
                    } catch (Exception ex) {
                        // TODO handle custom exceptions here
                    }



            }
            out.println("</body>");
            out.println("</html>");
        }
    }

}
```

Output:



CalculatorWS Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String org.me.calculator.CalculatorWS.prime(int)
prime ( )

public abstract double org.me.calculator.CalculatorWS.div(double,double)
div (,)

public abstract double org.me.calculator.CalculatorWS.mul(double,double)
mul (5 ,2)

public abstract double org.me.calculator.CalculatorWS.sub(double,double)
sub (,)

public abstract double org.me.calculator.CalculatorWS.add(double,double)
add (,)

public abstract double org.me.calculator.CalculatorWS.root(double)
root ( )

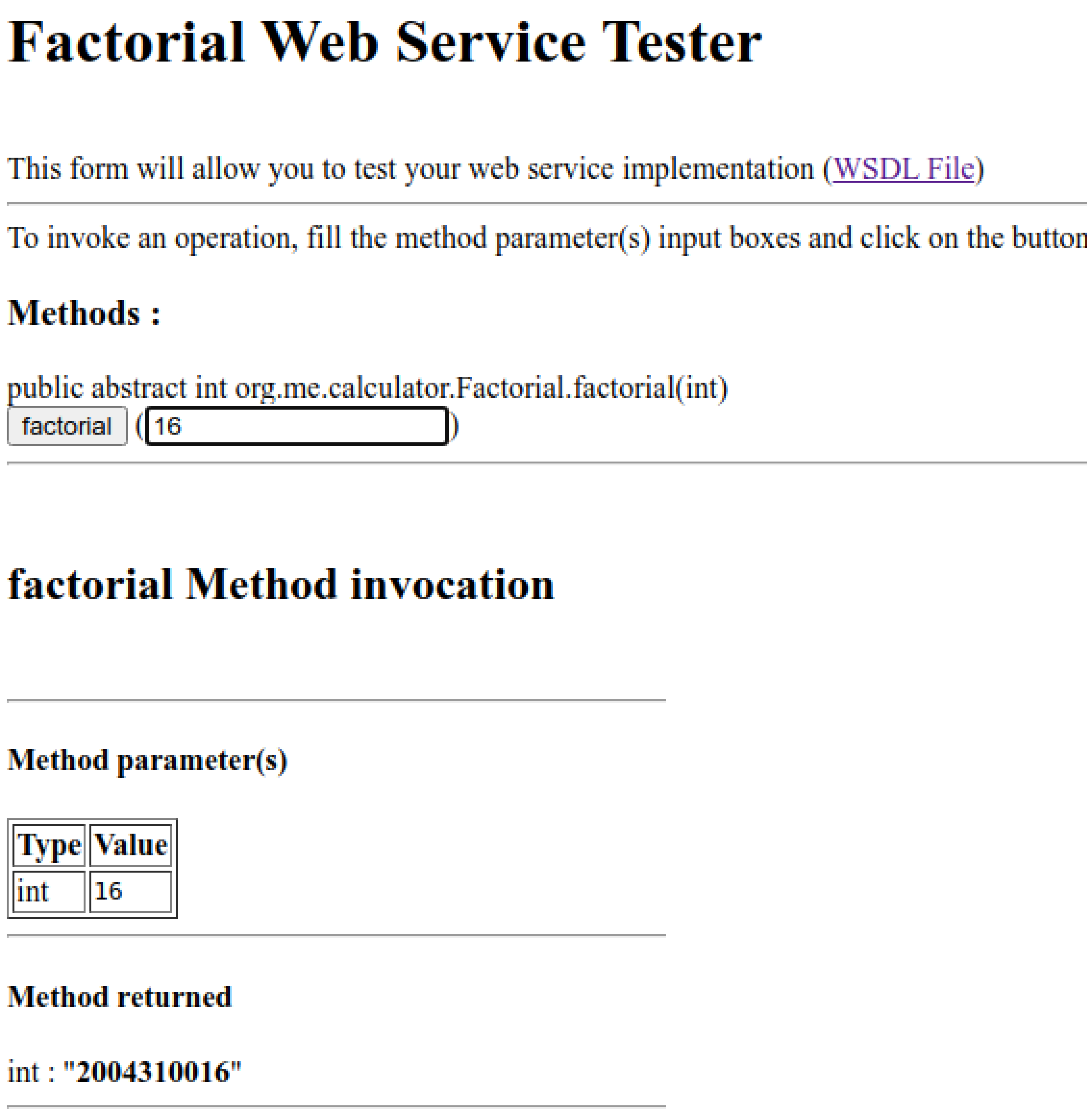
mul Method invocation

Method parameter(s)

Type	Value
double	5
double	2

Method returned

double : **"10.0"**



Factorial Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button

Methods :

public abstract int org.me.calculator.Factorial.factorial(int)
factorial (16)

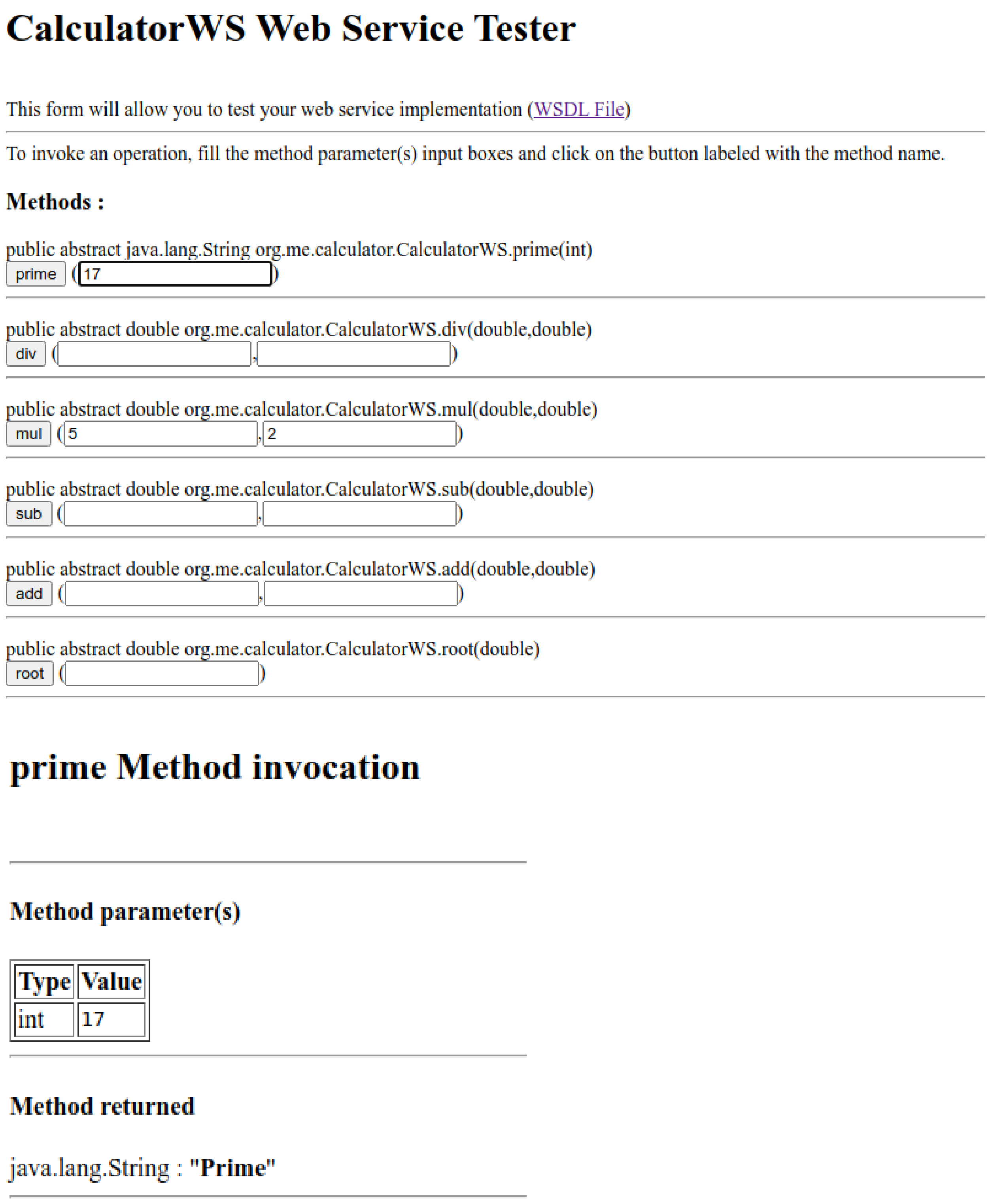
factorial Method invocation

Method parameter(s)

Type	Value
int	16

Method returned

int : "**2004310016**"



CalculatorWS Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.lang.String org.me.calculator.CalculatorWS.prime(int)
prime (17)

public abstract double org.me.calculator.CalculatorWS.div(double,double)
div (,)

public abstract double org.me.calculator.CalculatorWS.mul(double,double)
mul (5,2)

public abstract double org.me.calculator.CalculatorWS.sub(double,double)
sub (,)

public abstract double org.me.calculator.CalculatorWS.add(double,double)
add (,)

public abstract double org.me.calculator.CalculatorWS.root(double)
root ( )

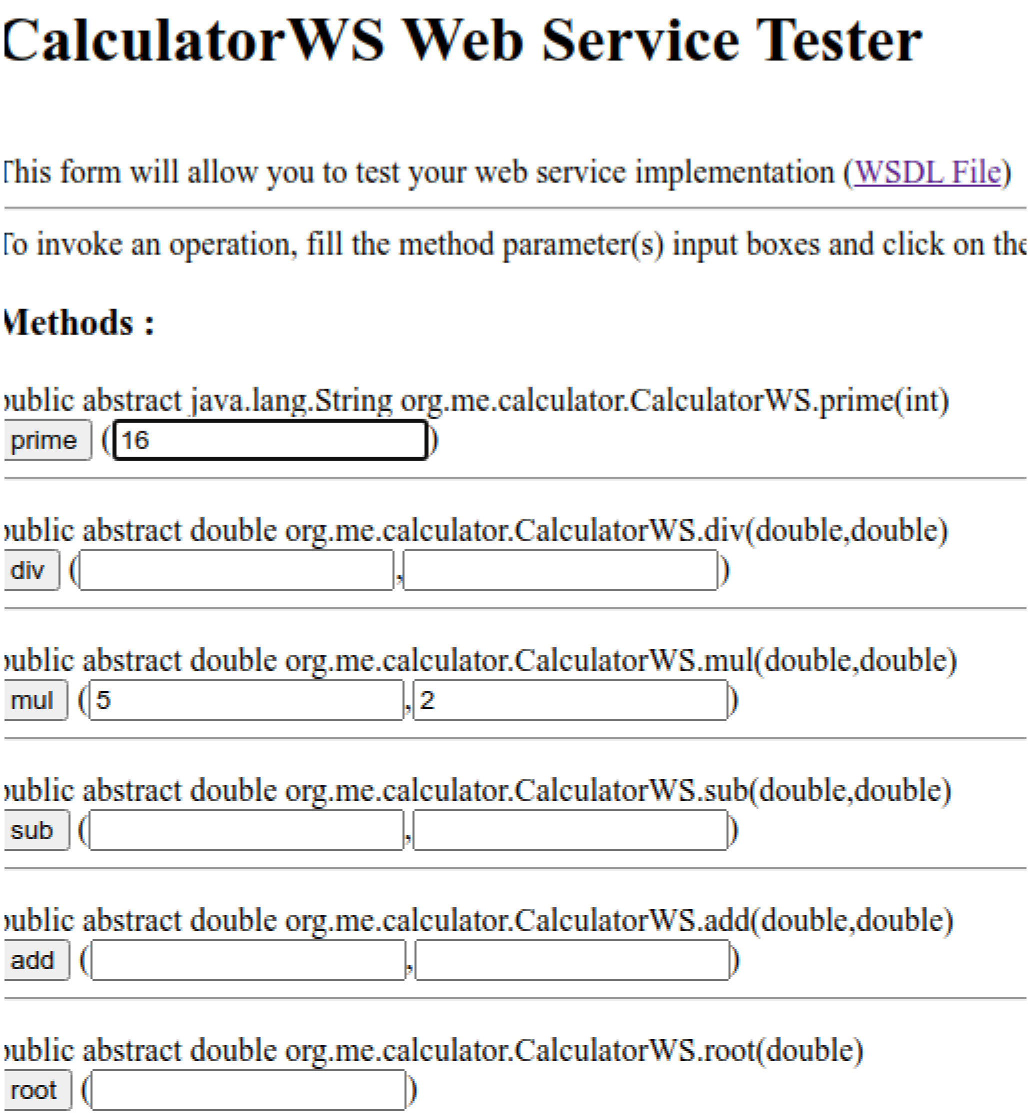
prime Method invocation

Method parameter(s)

Type	Value
int	17

Method returned

java.lang.String : "**Prime**"



CalculatorWS Web Service Tester

This form will allow you to test your web service implementation (WSDL File)

To invoke an operation, fill the method parameter(s) input boxes and click on the

Methods :

public abstract java.lang.String org.me.calculator.CalculatorWS.prime(int)
prime (16)

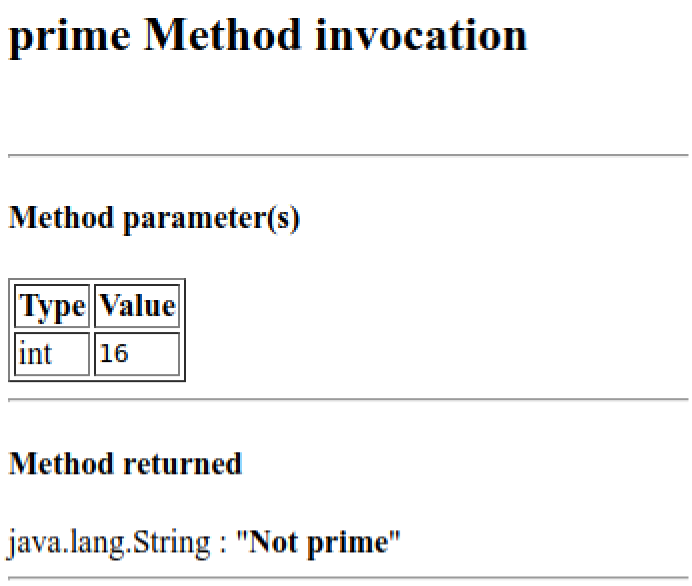
public abstract double org.me.calculator.CalculatorWS.div(double,double)
div (,)

public abstract double org.me.calculator.CalculatorWS.mul(double,double)
mul (5 ,2)

public abstract double org.me.calculator.CalculatorWS.sub(double,double)
sub (,)

public abstract double org.me.calculator.CalculatorWS.add(double,double)
add (,)

public abstract double org.me.calculator.CalculatorWS.root(double)
root ( )



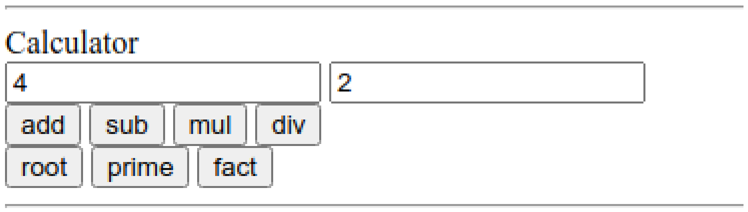
prime Method invocation

Method parameter(s)

Type	Value
int	16

Method returned

java.lang.String : "**Not prime**"



Calculator

4

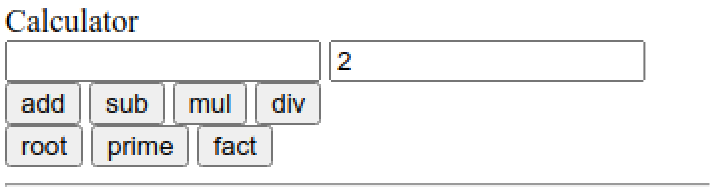
2

add Result = 6.0

sub Result = 2.0

mul Result = 8.0

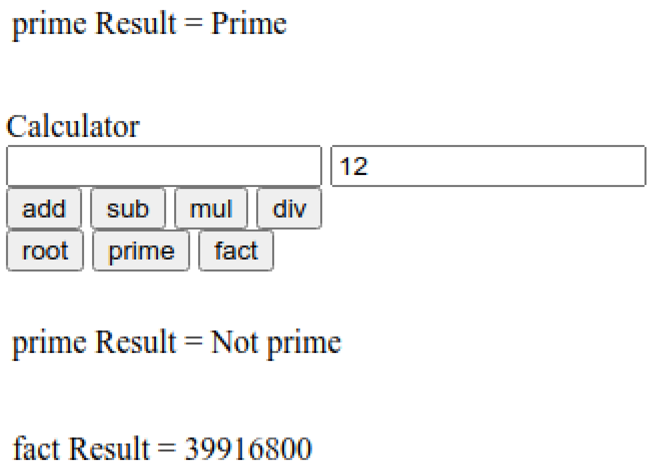
div Result = 2.0



Calculator

2

root Result = 1.4142135623730951



prime Result = Prime

Calculator

12

prime Result = Not prime

fact Result = 39916800