

CreditCard: la classe per la carta di credito

```
@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class CreditCard {

    @XmlAttribute(required = true)
    private String number;

    @XmlAttribute(name = "expiry_date", required = true)
    private String expiryDate;

    @XmlAttribute(name = "control_number", required = true)
    private Integer controlNumber;

    @XmlAttribute(required = true)
    private String type;

    //Constructors, getters, setters
}
```

- > Inizio dell'elemento root dell'XML
- > Tutti i campi saranno mappati su XML
- > Attributo obbligatorio
- > Attributo obbligatorio, con il nome XML diverso
- > Attributo obbligatorio, con il nome XML diverso

POJO usato come parametro per il metodo **validate()** del Web Service

CreditCard: la classe per la carta di credito

```
@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class CreditCard {

    @XmlAttribute(required = true)
    private String number;

    @XmlAttribute(name = "expiry_date", required = true)
    private String expiryDate;

    @XmlAttribute(name = "control_number", required = true)
    private Integer controlNumber;

    @XmlAttribute(required = true)
    private String type;

    //Constructors, getters, setters
}
```

- > Inizio dell'elemento root dell'XML
- > Tutti i campi saranno mappati su XML
- > Attributo obbligatorio
- > Attributo obbligatorio, con il nome XML diverso
- > Attributo obbligatorio, con il nome XML diverso
- > Attributo obbligatorio

POJO usato come parametro per il metodo **validate()** del Web Service

II Web Service

```

@WebService
public interface Validator {
    public boolean validate(CreditCard creditCard);
}

@WebService(endpointInterface =
    "org.agoncal.book.javaee7.chapter14.Validator")
public class CardValidator implements Validator {
    public boolean validate(CreditCard creditCard) {
        Character lastDigit = creditCard.getNumber().charAt(
            creditCard.getNumber().length() - 1);
        if(Integer.parseInt(lastDigit.toString()) % 2 == 0) {
            return true;
        }else{
            return false;
        }
    }
}

```

Con annotazione da WS...

II Web Service

```

@WebService
public interface Validator {
    public boolean validate(CreditCard creditCard);
}

@WebService(endpointInterface =
    "org.agoncal.book.javaee7.chapter14.Validator")
public class CardValidator implements Validator {
    public boolean validate(CreditCard creditCard) {
        Character lastDigit = creditCard.getNumber().charAt(
            creditCard.getNumber().length() - 1);
        if(Integer.parseInt(lastDigit.toString()) % 2 == 0) {
            return true;
        }else{
            return false;
        }
    }
}

```

Con annotazione da WS...

... si dichiara l'interfaccia

Il Web Service

```
@WebService
public interface Validator {
    public boolean validate(CreditCard creditCard);
}

@WebService(endpointInterface =
    "org.agoncal.book.javaee7.chapter14.Validator")
public class CardValidator implements Validator {
    public boolean validate(CreditCard creditCard) {
        Character lastDigit = creditCard.getNumber().charAt(
            creditCard.getNumber().length() - 1);
        if(Integer.parseInt(lastDigit.toString()) % 2 == 0) {
            return true;
        }else{
            return false;
        }
    }
}
```

› Con annotazione da WS...

› ... si dichiara l'interfaccia

› Poi si dichiara il WS

Il Web Service

```
@WebService
public interface Validator {
    public boolean validate(CreditCard creditCard);
}

@WebService(endpointInterface =
    "org.agoncal.book.javaee7.chapter14.Validator")
public class CardValidator implements Validator {
    public boolean validate(CreditCard creditCard) {
        Character lastDigit = creditCard.getNumber().charAt(
            creditCard.getNumber().length() - 1);
        if(Integer.parseInt(lastDigit.toString()) % 2 == 0) {
            return true;
        }else{
            return false;
        }
    }
}
```

› Con annotazione da WS...

› ... si dichiara l'interfaccia

› Poi si dichiara il WS

› Con la classe che implementa l'interfaccia

II Web Service

```
@WebService
public interface Validator {
    public boolean validate(CreditCard creditCard);
}

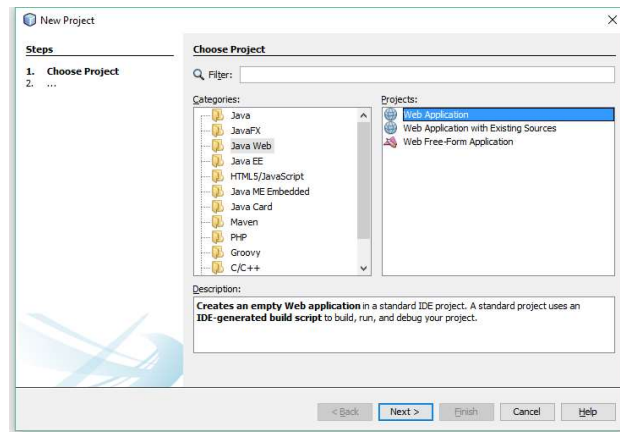
@WebService(endpointInterface =
    "org.agoncal.book.javaee7.chapter14.Validator")
public class CardValidator implements Validator {
    public boolean validate(CreditCard creditCard) {
        Character lastDigit = creditCard.getNumber().charAt(
            creditCard.getNumber().length() - 1);
        if(Integer.parseInt(lastDigit.toString()) % 2 != 0) {
            return true;
        }else{
            return false;
        }
    }
}
```

- › Con annotazione da WS...
- › ... si dichiara l'interfaccia
- › Poi si dichiara il WS
- › Con la classe che implementa l'interfaccia
- › Metodo offerto come WS

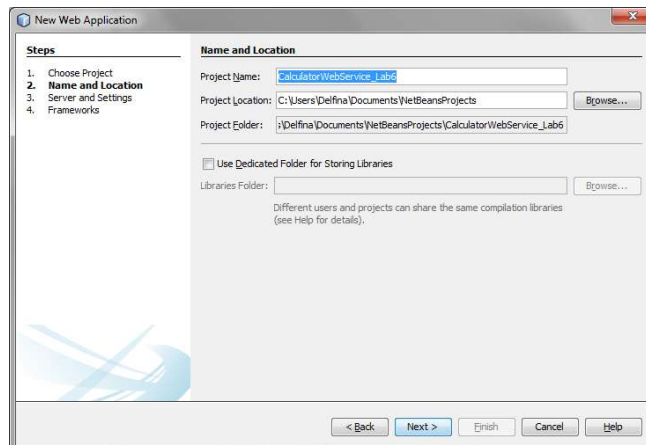
Organizzazione della lezione

- WS in Java
 - WSDL Mapping
 - Eccezioni e Fault
 - Contesto e ciclo di vita
- Putting It All Together
 - Un esempio riassuntivo
- Supporto ai WS in Netbeans
 - Il progetto per WS
 - Testing
 - WS Client
- Conclusioni

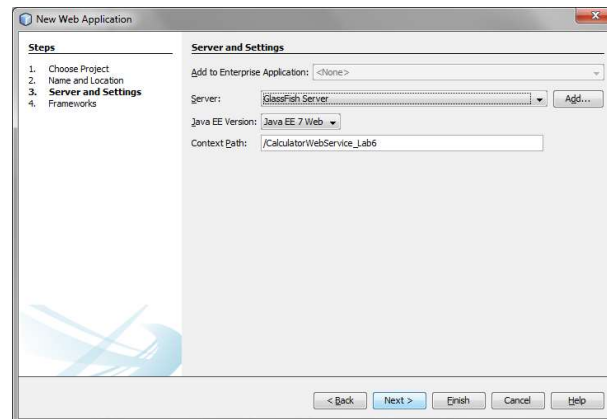
WS in NetBeans: creazione di un progetto



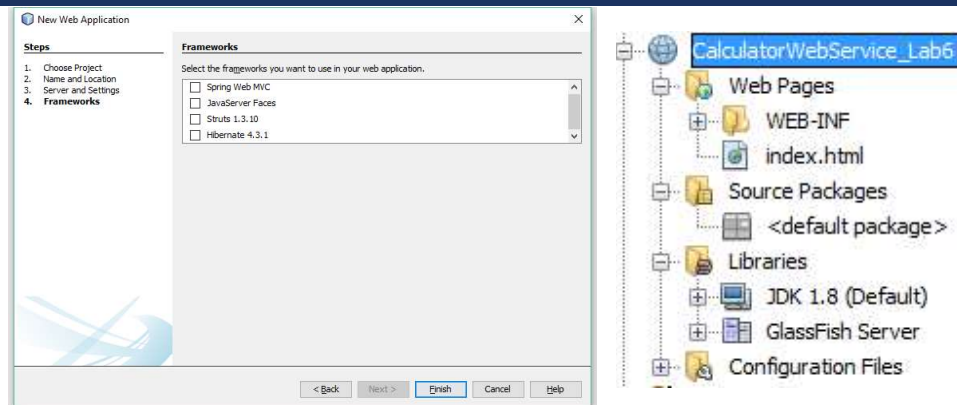
WS in NetBeans: creazione di un progetto



WS in NetBeans: creazione di un progetto

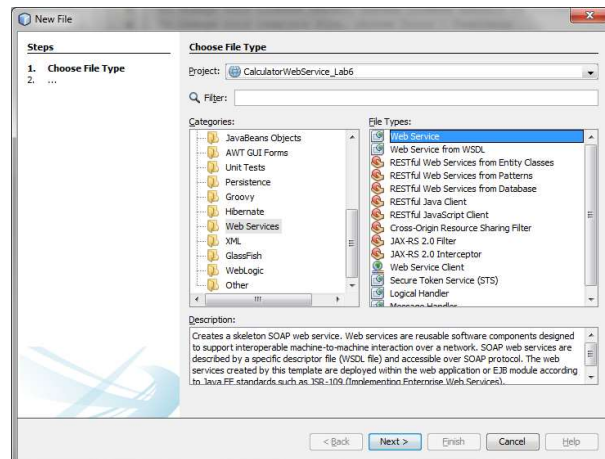


WS in NetBeans: creazione di un progetto

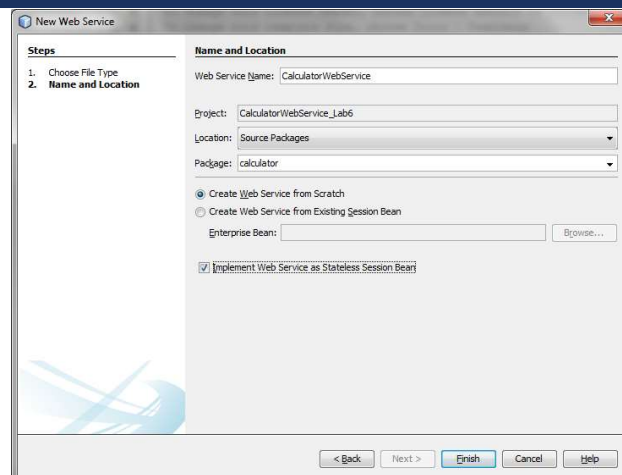


WS in NetBeans: creazione di un progetto

Right-click sul progetto →
New



WS in NetBeans: creazione di un WS



WS in NetBeans: la vista di SOURCE

CalculatorWebService_Lab6

- Web Pages
 - WEB-INF
 - index.html
- Source Packages
 - calculator
 - CalculatorWebService.java
- Libraries
 - JDK 1.8 (Default)
 - GlassFish Server
 - Enterprise Beans
 - Web Services
 - Configuration Files

```

1  /*
2  * To change this license header, choose License Headers in Project Properties
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package calculator;
7
8  import javax.ws.rs.WebServiceException;
9  import javax.ws.rs.WebMethodException;
10 import javax.ws.rs.WebParam;
11 import javax.ws.rs.Stateless;
12
13 /**
14  *
15  * @author Delfina
16  */
17 @WebService(serviceName = "CalculatorWebService")
18 @Stateless()
19 public class CalculatorWebService {
20
21     /**
22      * This is a sample web service operation
23      */
24     @WebMethod(operationName = "hello")
25     public String hello(@WebParam(name = "name") String txt) {
26         return "Hello " + txt + " !";
27     }
28 }

```

WS in NetBeans: la vista di DESIGN

CalculatorWebService

Operations (1)

Add Operation... Remove Operation

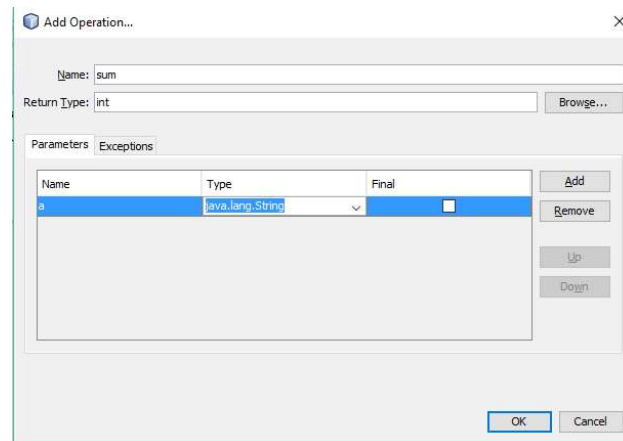
Parameters	Output	Faults	Description				
<table border="1"> <thead> <tr> <th>Parameter Name</th> <th>Parameter Type</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>java.lang.String</td> </tr> </tbody> </table>	Parameter Name	Parameter Type	name	java.lang.String			
Parameter Name	Parameter Type						
name	java.lang.String						

Quality Of Service

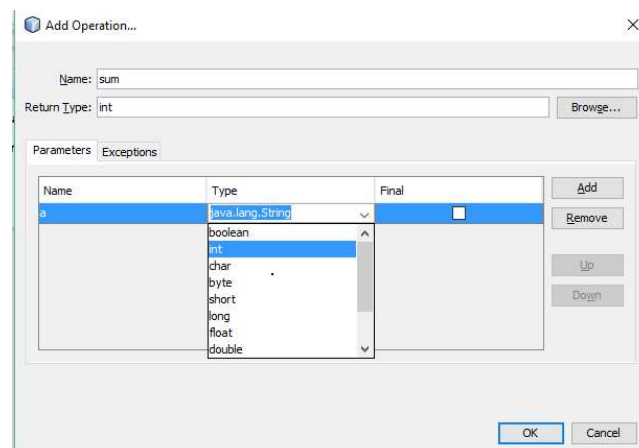
- ☐ Optimize Transfer Of Binary Data (MTOM)
- ☐ Reliable Message Delivery
- ☐ Secure Service

Edit Web Service Attributes...

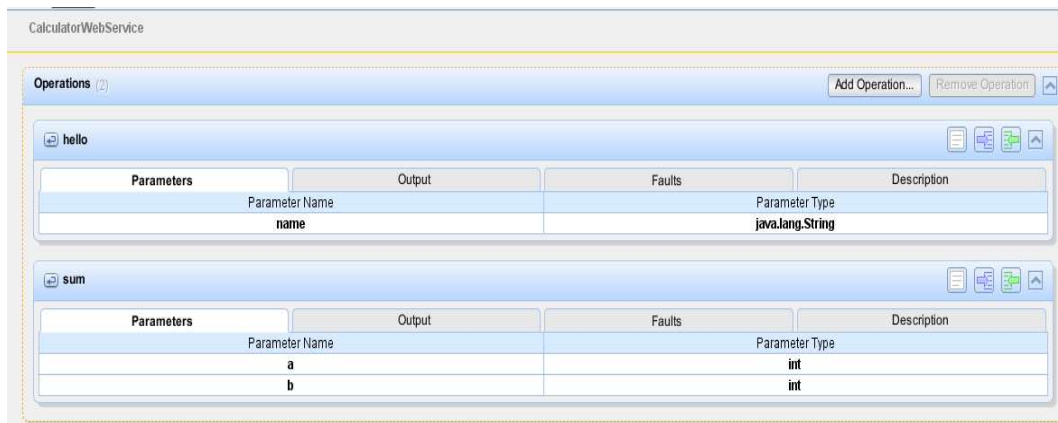
WS in NetBeans: aggiungiamo una operazione



WS in NetBeans: aggiungiamo un parametro



WS in NetBeans: dopo aver aggiunto un secondo parametro



WS in NetBeans: il codice

```

10  import javax.jws.WebParam;
11  import javax.jws.WebService;
12  import javax.xml.ws.Stateless;
13
14  /**
15   * @author Delfina
16   */
17  @WebService(serviceName = "CalculatorWebService")
18  @Stateless()
19  public class CalculatorWebService {
20
21      /**
22       * This is a sample web service operation
23       */
24      @WebMethod(operationName = "hello")
25      public String hello(@WebParam(name = "name") String txt) {
26          return "Hello " + txt + " !";
27      }
28
29      /**
30       * Web service operation
31       */
32      @WebMethod(operationName = "sum")
33      public int sum(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
34          //TODO write your implementation code here:
35          return 0;
36      }
37  }

```

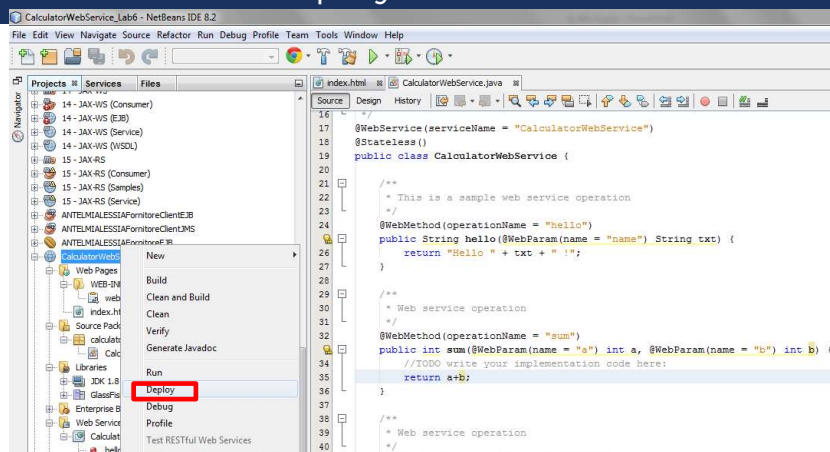
WS in NetBeans

WS in NetBeans: scriviamo qualcosa (implementiamo somma e moltiplicazione)

```
@WebMethod(operationName = "sum")
public int sum(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
    //TODO write your implementation code here:
    return a+b;
}

/**
 * Web service operation
 */
@WebMethod(operationName = "multiply")
public int multiply(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {
    //TODO write your implementation code here:
    return 0;
}
```

WS in NetBeans: deploy



WS in NetBeans: sul server

```

Output | Test Results
Java DB Database Process | GlassFish Server | CalculatorWebService_Lab6 (run-deploy)
-----|-----|-----
Informazioni: Created HTTP listener http-listener-1 on host/port 0.0.0.0:8080
Informazioni: Grizzly Framework 2.3.15 started in: 15ms - bound to [/0.0.0.0:8080]
Informazioni: visiting unvisited references
Informazioni: Web service endpoint deployment events listener registered successfully.
Informazioni: visiting unvisited references
Informazioni: visiting unvisited references
Informazioni: visiting unvisited references
Avvertenza: Container org.glassfish.webservices.JAXWSContainer@4dd412a8 doesn't support class com.sun.xml.ws.api
Informazioni: Portable JNDI names for EJB CalculatorWebService: [java:global/CalculatorWebService_Lab6/Calculato
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] org.glassfish.sse.impl.ServerSentEventCdiExtension.pr
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] private org.glassfish.jersey.gf.cdi.internal.CdiCompo
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] public org.glassfish.jms.injection.JMSCDIExtension.pr
Informazioni: EJB Endpoint deployed CalculatorWebService_Lab6
listening at address at http://Delfina-PC:8080/CalculatorWebService/CalculatorWebService
Informazioni: EJB Endpoint deployed CalculatorWebService_Lab6
listening at address at http://Delfina-PC:8080/CalculatorWebService/CalculatorWebService
Informazioni: Loading application [CalculatorWebService_Lab6] at [/CalculatorWebService_Lab6]
Informazioni: CalculatorWebService Lab6 was successfully deployed in 8.957 milliseconds.
  
```

Organizzazione della lezione

- WS in Java
 - WSDL Mapping
 - Eccezioni e Fault
 - Contesto e ciclo di vita
- Putting it All Together
 - Un esempio riassuntivo
- Supporto ai WS in Netbeans
 - Il progetto per WS
 - Testing
 - WS Client
- Conclusioni

WS in NetBeans: testing

The screenshot shows the NetBeans IDE interface. On the left, the 'Project Explorer' displays a project named 'CalculatorWebService_Lab6'. Under 'Web Services', there is a package named 'calculator'. A right-click context menu is open over the 'calculator' package, and the 'Test Web Service' option is highlighted with a red rectangle. The 'Output' window at the bottom shows the deployment logs for 'CalculatorWebService_Lab6', indicating successful deployment and endpoint listening at http://Delfina-PC:8080/CalculatorWebService/CalculatorWebService.

WS in NetBeans: Test

The screenshot shows a web browser window displaying the 'CalculatorWebService Web Service Tester' application. The page title is 'CalculatorWebService Web Service Tester'. Below the title, there is a description: 'This form will allow you to test your web service implementation (WSDL File)'. A note states: 'To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.' Under the heading 'Methods:', there are three methods listed with their respective input fields:

- `public abstract int calculator.CalculatorWebService.sum(int,int)`: The input field contains 'sum'.
- `public abstract int calculator.CalculatorWebService.multiply(int,int)`: The input field is empty.
- `public abstract java.lang.String calculator.CalculatorWebService.hello(java.lang.String)`: The input field contains 'hello'.

 Each method has a corresponding button to the right of its input field.

WS in NetBeans: testing

Cosa è successo
sul server

```

Output | Test Results
Java DB Database Process | GlassFish Server | CalculatorWebService_Lab6 (run-deploy)
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] private org.glassfish.jersey.gf.cdi.internal.CdiC
WARN: WELD-000411: Observer method [BackedAnnotatedMethod] public org.glassfish.jms.injection.JMSCDIExtension
Informazioni: EJB Endpoint deployed CalculatorWebService_Lab6
Informazioni: EJB Endpoint deployed CalculatorWebService_Lab6
Informazioni: listening at address at http://Delfina-PC:8080/CalculatorWebService/CalculatorWebService
Informazioni: listening at address at http://Delfina-PC:8080/CalculatorWebService/CalculatorWebService
Informazioni: Loading application [CalculatorWebService_Lab6] at [/CalculatorWebService_Lab6]
Informazioni: CalculatorWebService_Lab6 was successfully deployed in 8.957 milliseconds.
Informazioni: Invoking wsimport with http://localhost:8080/CalculatorWebService/CalculatorWebService?WSDL
Informazioni: analisi di WSDL in corso...
Informazioni: Generazione del codice in corso...
Informazioni: Compilazione del codice in corso...
Informazioni: wsimport successful
Informazioni: Invoking wsimport with http://localhost:8080/CalculatorWebService/CalculatorWebService?WSDL
Informazioni: analisi di WSDL in corso...
Informazioni: Generazione del codice in corso...
Informazioni: Compilazione del codice in corso...
Informazioni: wsimport successful
  
```

WS i

```

This XML file does not appear to have any style information associated with it. The document tree is shown below:
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2 -->
  <!-- Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.3.1-b419 (branches/2.3.1.x-7937; 2014-08-04T08:11:03+0000) JAXWS-RI/2 -->
  <definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wspl="http://www.w3.org/ns/ws-policy" xmlns:wspl2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/soap/" xmlns:tns="http://calculator/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://calculator/" name="CalculatorWebService">
    <types>
      <xsd:schema base="http://calculator/" schemaLocation="http://localhost:8080/CalculatorWebService/CalculatorWebService?xsd=1"/>
      </xsd:schema>
    </types>
    <message name="sum">
      <part name="parameters" element="tns:sum"/>
    </message>
    <message name="sumResponse">
      <part name="parameters" element="tns:sumResponse"/>
    </message>
    <message name="multiply">
      <part name="parameters" element="tns:multiply"/>
    </message>
    <message name="multiplyResponse">
      <part name="parameters" element="tns:multiplyResponse"/>
    </message>
    <message name="hello">
      <part name="parameters" element="tns:hello"/>
    </message>
    <message name="helloResponse">
      <part name="parameters" element="tns:helloResponse"/>
    </message>
    <portType name="CalculatorWebService">
      <operation name="sum">
        <input wsam:Action="http://calculator/CalculatorWebService/sumRequest" message="tns:sum"/>
        <output wsam:Action="http://calculator/CalculatorWebService/sumResponse" message="tns:sumResponse"/>
      </operation>
      <operation name="multiply">
  
```

WS in NetBeans: Test

CalculatorWebService 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 int calculator.CalculatorWebService.sum(int,int)
sum (3) (5)

public abstract int calculator.CalculatorWebService.multiply(int,int)
multiply () ()

public abstract java.lang.String calculator.CalculatorWebService.hello(java.lang.String)
hello ()

WS in

Method invocation trace

localhost:8080/CalculatorWebService/CalculatorWebService?Tester

Type	Value
int	3
int	5

Method returned

int : "8"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-ENV:Header/><S:Body><ns2:sum xmlns:ns2="http://calculator/"><a>3</a><b>5</b></ns2:sum></S:Body></S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"><SOAP-ENV:Header/><S:Body><ns2:sumResponse xmlns:ns2="http://calculator/"><return>8</return></ns2:sumResponse></S:Body></S:Envelope>
```

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/" >
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:sum xmlns:ns2="http://calculator/">
      <a>3</a>
      <b>5</b>
    </ns2:sum>
  </S:Body>
</S:Envelope>
```

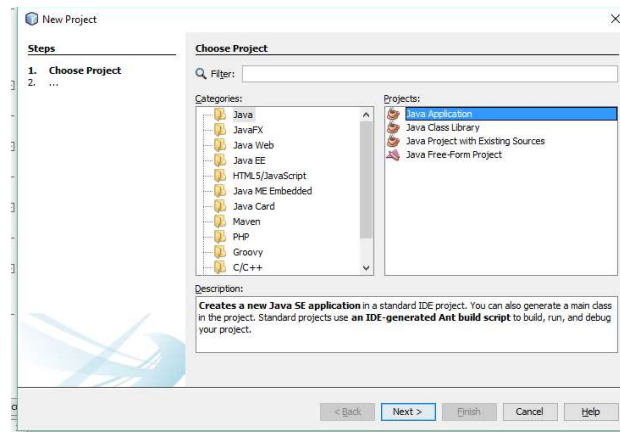
SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  <SOAP-ENV:Header/>
  <S:Body>
    <ns2:sumResponse xmlns:ns2="http://calculator/">
      <return>8</return>
    </ns2:sumResponse>
  </S:Body>
</S:Envelope>
```

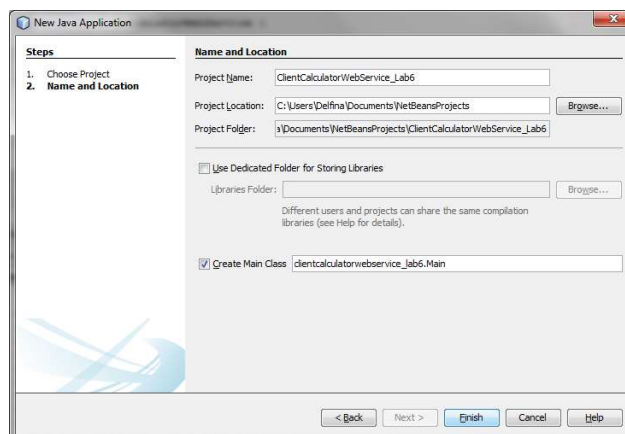
Organizzazione della lezione

- WS in Java
 - WSDL Mapping
 - Eccezioni e Fault
 - Contesto e ciclo di vita
- Invocare un WS
 - Un esempio riassuntivo
- Supporto ai WS in Netbeans
 - Il progetto per WS
 - Testing
 - WS Client
- Conclusioni

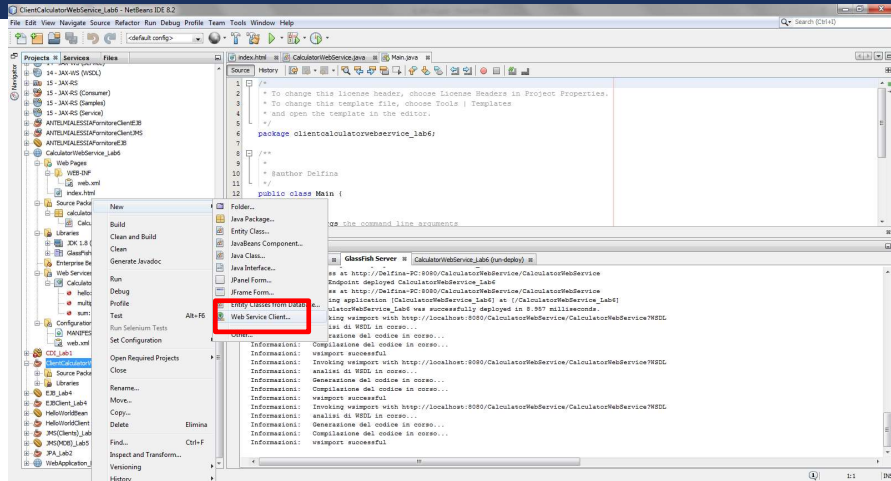
WS in NetBeans: Client



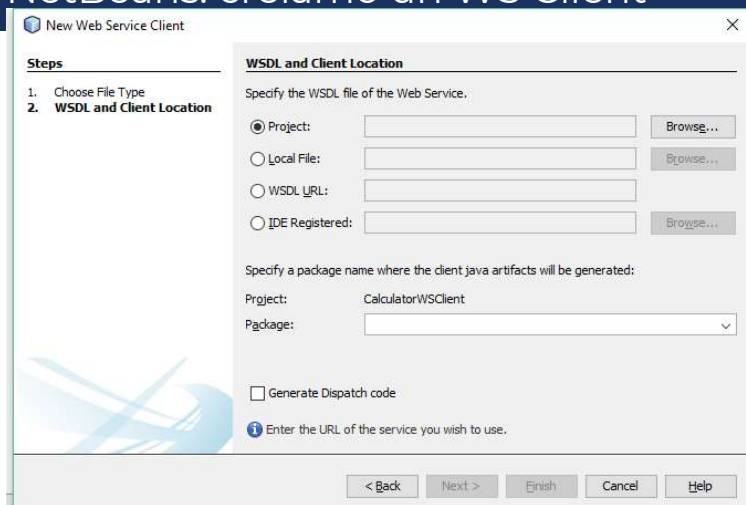
WS in NetBeans: Client



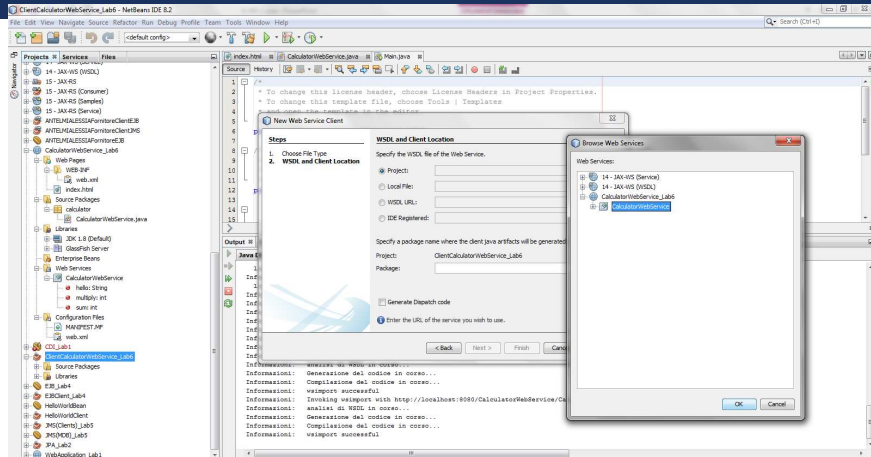
WS in NetBeans: creiamo un WS Client



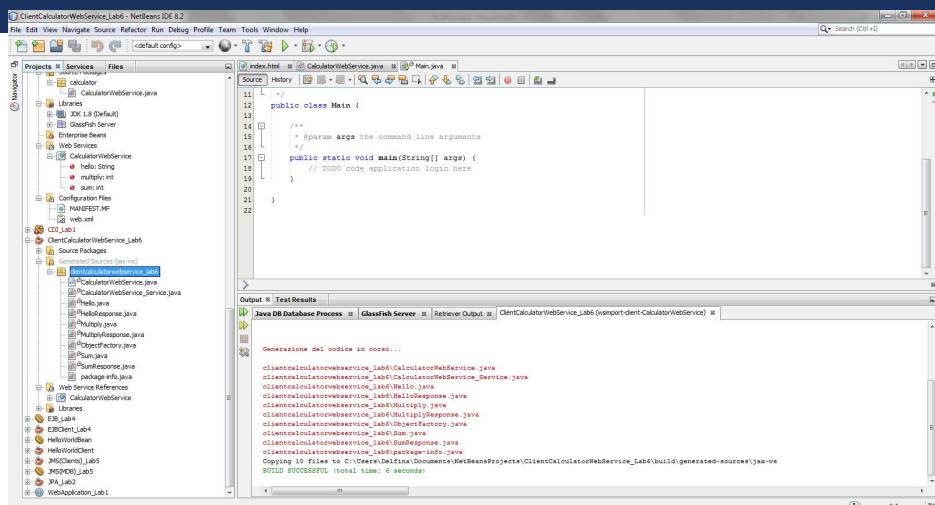
WS in NetBeans: creiamo un WS Client



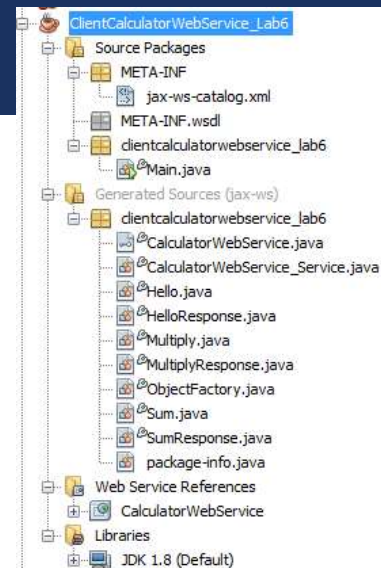
WS in NetBeans: creiamo un WS Client



La situazione attuale



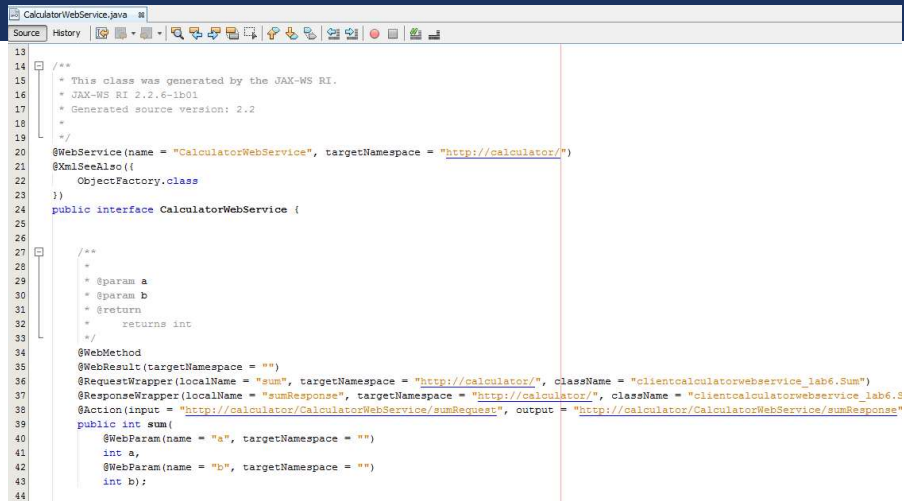
WS in NetBeans: i files generati



WS in NetBeans: Output



WS in NetBeans: l'interfaccia generata (sul client)

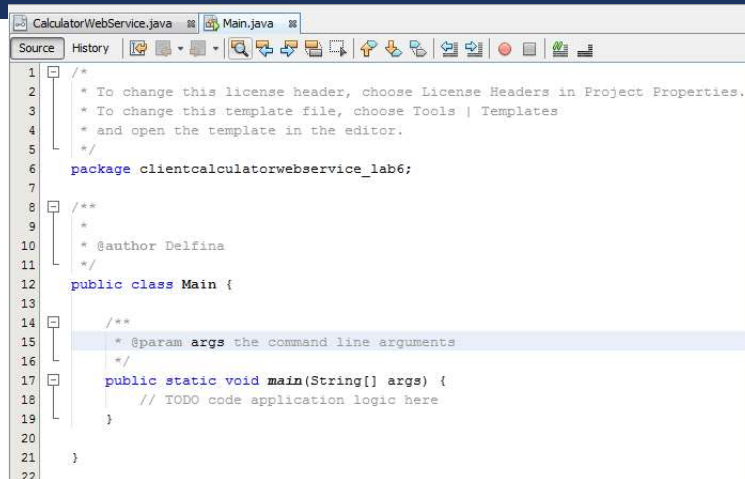


```

13
14 /**
15  * This class was generated by the JAX-WS RI.
16  * JAX-WS RI 2.2.6-1b01
17  * Generated source version: 2.2
18  */
19
20 @WebService(name = "CalculatorWebService", targetNamespace = "http://calculator/")
21 @XmlSeeAlso({
22     ObjectFactory.class
23 })
24 public interface CalculatorWebService {
25
26
27     /**
28      *
29      * @param a
30      * @param b
31      * @return
32      * Returns int
33      */
34     @WebMethod
35     @WebResult(targetNamespace = "")
36     @RequestWrapper(localName = "sum", targetNamespace = "http://calculator/", className = "clientcalculatorwebservice_lab6.Sum")
37     @ResponseWrapper(localName = "sumResponse", targetNamespace = "http://calculator/", className = "clientcalculatorwebservice_lab6.SumResponse")
38     @Action(input = "http://calculator/CalculatorWebService/sumRequest", output = "http://calculator/CalculatorWebService/sumResponse")
39     public int sum(
40         @WebParam(name = "a", targetNamespace = "")
41         int a,
42         @WebParam(name = "b", targetNamespace = "")
43         int b);
44

```

WS in NetBeans: il Client (VUOTO)

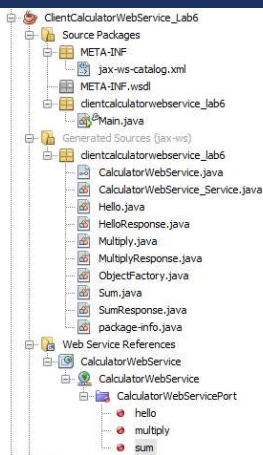


```

1
2 /**
3  * To change this license header, choose License Headers in Project Properties.
4  * To change this template file, choose Tools | Templates
5  * and open the template in the editor.
6  */
7
8 package clientcalculatorwebservice_lab6;
9
10 /**
11  *
12  * @author Delfina
13  */
14 public class Main {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21     }
22

```

Dal progetto WS...



WS in NetBeans: Drag & Drop operazione hello, sum multiply

```

7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
Source History
CalculatorWebService.java Main.java
/**
 *
 * @author Delfina
 */
public class Main {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

    private static String hello(java.lang.String name) {
        clientcalculatorwebservice_lab6.CalculatorWebService_Service service = new clientcalculatorwebservice_lab6.CalculatorWebService_Service();
        clientcalculatorwebservice_lab6.CalculatorWebService port = service.getCalculatorWebServicePort();
        return port.hello(name);
    }

    private static int multiply(int a, int b) {
        clientcalculatorwebservice_lab6.CalculatorWebService_Service service = new clientcalculatorwebservice_lab6.CalculatorWebService_Service();
        clientcalculatorwebservice_lab6.CalculatorWebService port = service.getCalculatorWebServicePort();
        return port.multiply(a, b);
    }

    private static int sum(int a, int b) {
        clientcalculatorwebservice_lab6.CalculatorWebService_Service service = new clientcalculatorwebservice_lab6.CalculatorWebService_Service();
        clientcalculatorwebservice_lab6.CalculatorWebService port = service.getCalculatorWebServicePort();
        return port.sum(a, b);
    }
}

```

WS in NetBeans: Drag & Drop operazione hello, sum multiply

```
private static String hello(java.lang.String name) {
    clientcalculatorwebservice_lab6.CalculatorWebService_Service service =
        new clientcalculatorwebservice_lab6.CalculatorWebService_Service();
    clientcalculatorwebservice_lab6.CalculatorWebService port = service.getCalculatorWebServicePort();
    return port.hello(name);
}

private static int multiply(int a, int b) {
    clientcalculatorwebservice_lab6.CalculatorWebService_Service service =
        new clientcalculatorwebservice_lab6.CalculatorWebService_Service();
    clientcalculatorwebservice_lab6.CalculatorWebService port = service.getCalculatorWebServicePort();
    return port.multiply(a, b);
}

private static int sum(int a, int b) {
    clientcalculatorwebservice_lab6.CalculatorWebService_Service service =
        new clientcalculatorwebservice_lab6.CalculatorWebService_Service();
    clientcalculatorwebservice_lab6.CalculatorWebService port = service.getCalculatorWebServicePort();
    return port.sum(a, b);
}
```

WS in NetBeans: esecuzione

WS in NetBeans:
... un po' di codice

```
Source History
4  * and open the template in the editor.
5  */
6  package clientcalculatorwebservice_lab6;
7
8  /**
9   *
10  * @author Delfina
11  */
12  public class Main {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          System.out.println(hello("Delfina"));
19          System.out.println("la somma di 5+3= "+sum(5,3));
20      }
21  }
```

WS in NetBeans: esecuzione

Output Test Results

Java DB Database Process GlassFish Server Retriever Output ClientCalculatorWebService_Lab6 (run-single)

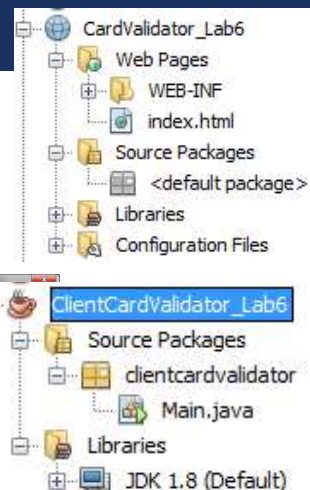
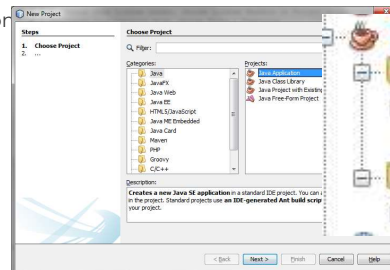
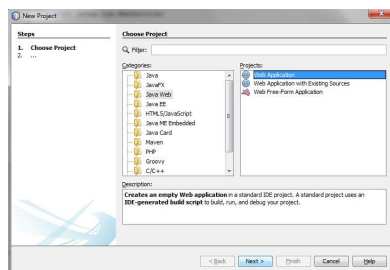
```

ant -f C:\Users\Delfina\Documents\NetBeansProjects\ClientCalculatorWebService_Lab6 -Djava:
init:
Deleting: C:\Users\Delfina\Documents\NetBeansProjects\ClientCalculatorWebService_Lab6\build\t
deps-jar:
Updating property file: C:\Users\Delfina\Documents\NetBeansProjects\ClientCalculatorWebServic
wsimport-init:
wsimport-client-ClientCalculatorWebService:
files are up to date
wsimport-client-generate:
Compiling 1 source file to C:\Users\Delfina\Documents\NetBeansProjects\ClientCalculatorWebSe:
compile-single:
run-single:
Hello Delfina !
la somma di 5+3= 8
BUILD SUCCESSFUL (total time: 6 seconds)

```

CreditCardValidator Web Service

- Passo1: Creazione del progetto
 - CardValidator_Lab6



CreditCardValidator Web Service

Listing 14-34. The CreditCard Class with JAXB Annotations

- La classe CreditCard

```
@XmlElement
@XmlAccessorType(XmlAccessType.FIELD)
public class CreditCard {

    @XmlAttribute(required = true)
    private String number;
    @XmlAttribute(name = "expiry_date", required = true)
    private String expiryDate;
    @XmlAttribute(name = "control_number", required = true)
    private Integer controlNumber;
    @XmlAttribute(required = true)
    private String type;

    // Constructors, getters, setters
}
```

CreditCardValidator Web Service

Listing 14-35. The Validator Web Service Interface

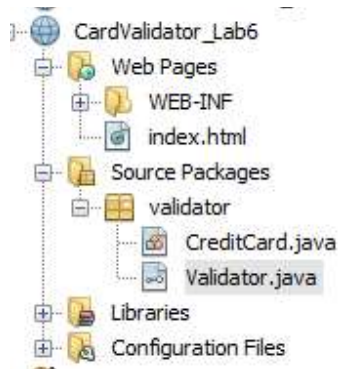
- L'interfaccia

```
@WebService
public interface Validator {

    public boolean validate(CreditCard creditCard);
}
```

La situazione attuale

- In NetBeans



CreditCardValidator Web Service

Listing 14-36. The CardValidator Web Service Bean

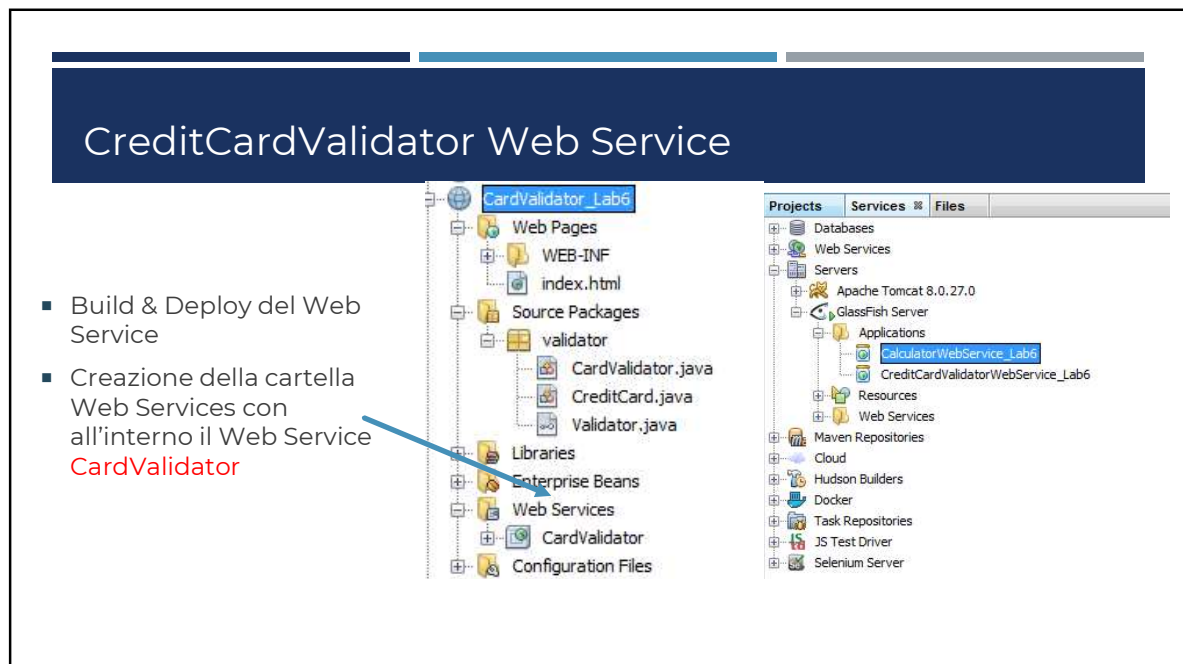
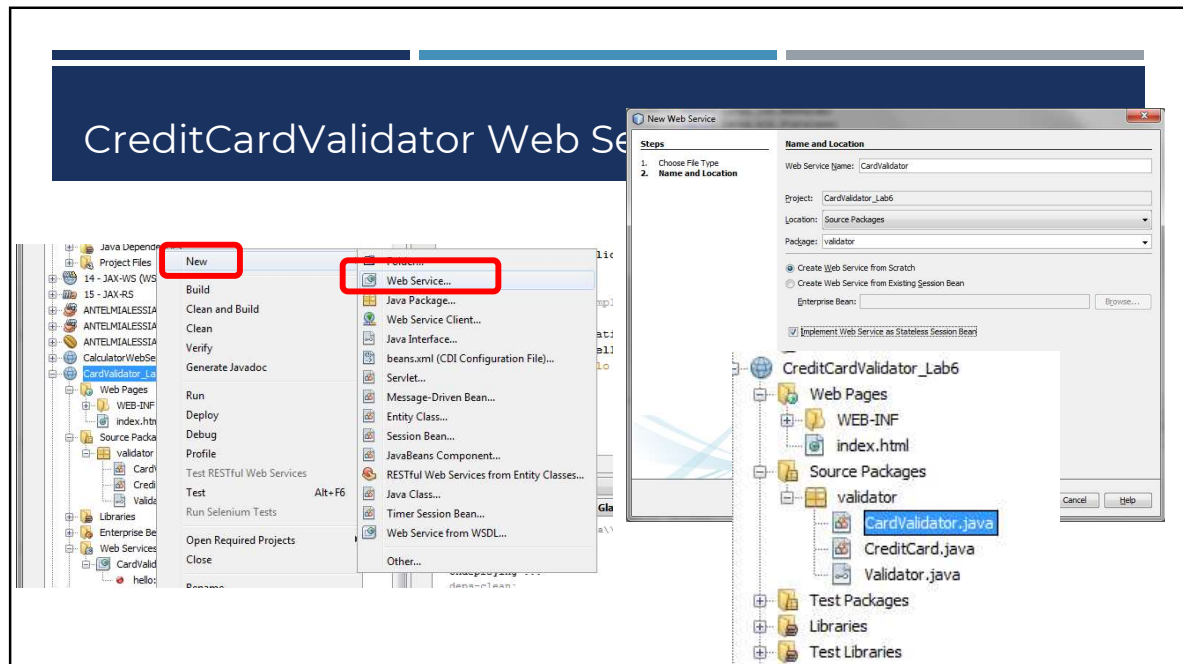
```
@WebService(endpointInterface = "org.agoncal.book.javaee7.chapter14.Validator")
public class CardValidator implements Validator {

    public boolean validate(CreditCard creditCard) {

        Character lastDigit = creditCard.getNumber().charAt(
            creditCard.getNumber().length() - 1);

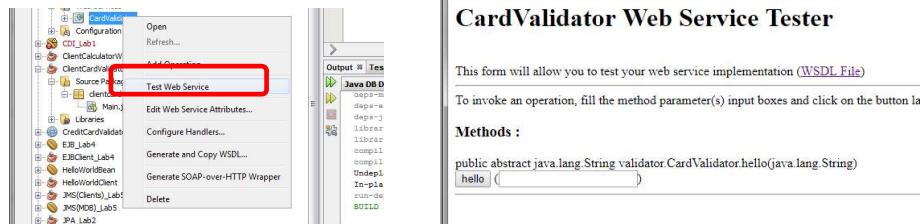
        if (Integer.parseInt(lastDigit.toString()) % 2 == 0) {
            return true;
        } else {
            return false;
        }
    }
}
```

- La classe CardValidator



CreditCardValidator Web Service

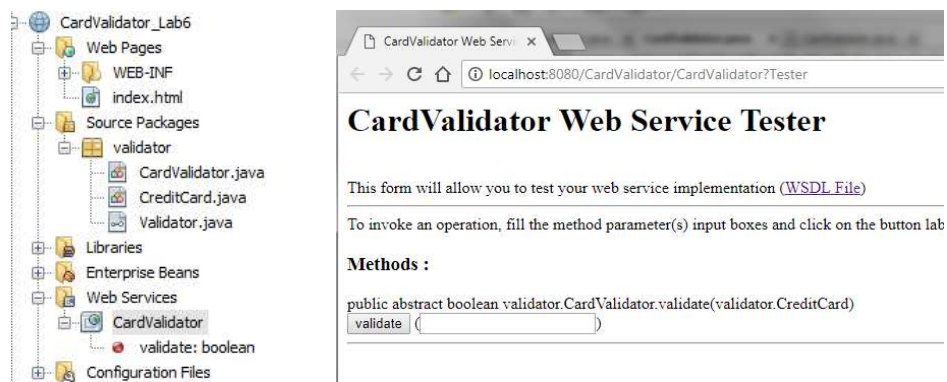
- Per verificare che tutto funzioni



- E' presente il Web Service di default!
- Rimuoviamo hello(), ed aggiungiamo validate()

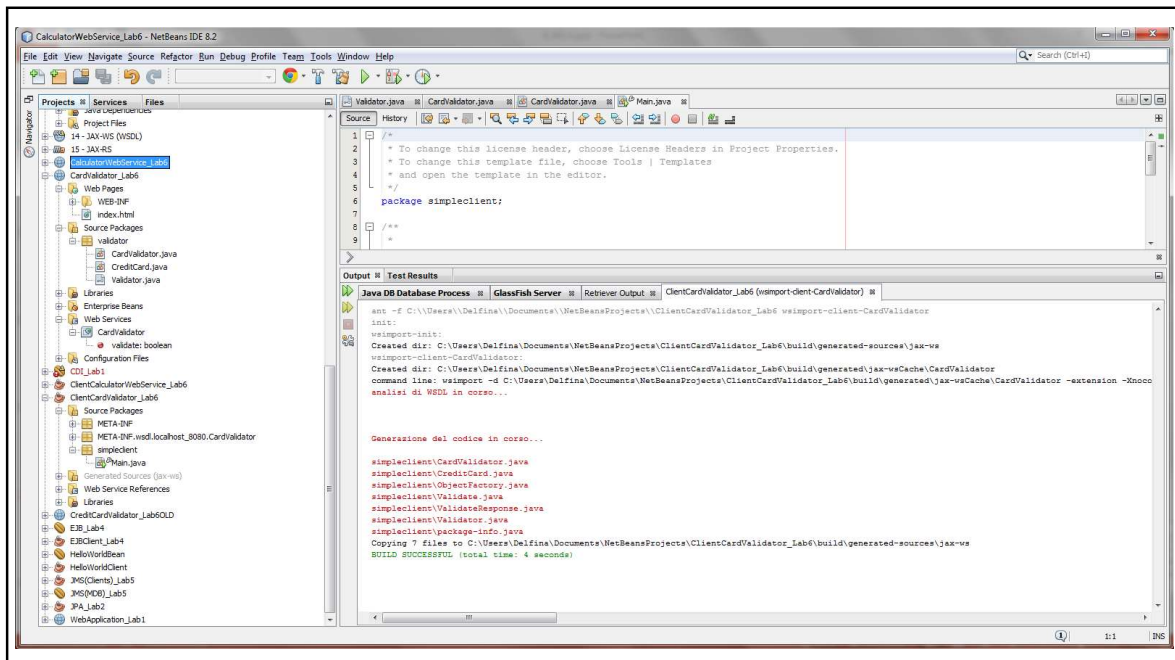
CreditCardValidator Web Service

- Nuova situazione



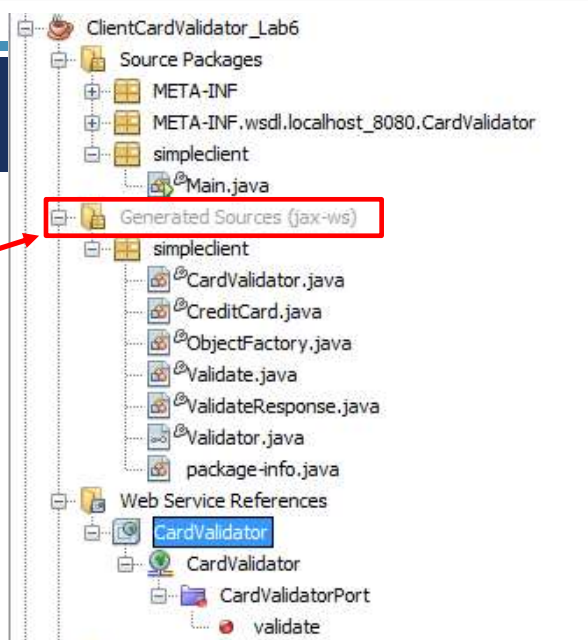
CreditCardValidator Web Service

- Ora il Consumer: ClientCardValidator_Lab6
- Right-click sul progetto client → new Web Service client



CreditCardValidator Web Service

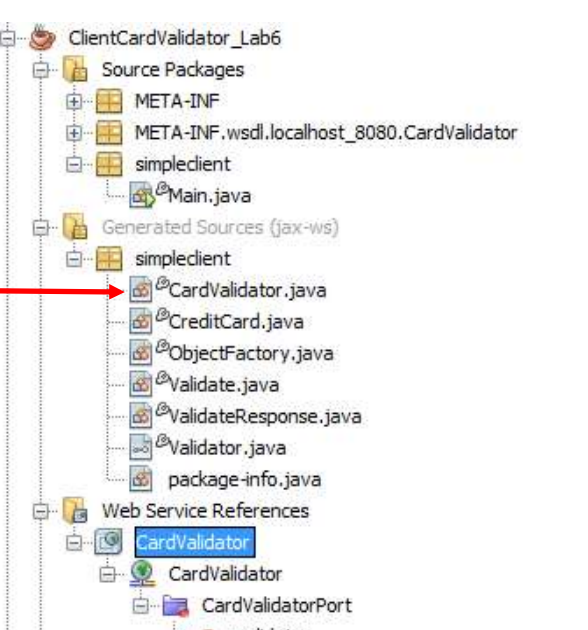
- La situazione:
 - Generata la cartella Generated Sources (jax-ws)



The screenshot shows the project structure of 'ClientCardValidator_Lab6'. Under 'Source Packages', there are 'META-INF', 'META-INF.wsdl.localhost_8080.CardValidator', and 'simpleclient'. The 'simpleclient' package contains several Java files: '@Main.java', '@CardValidator.java', '@CreditCard.java', '@ObjectFactory.java', '@Validate.java', '@ValidateResponse.java', '@Validator.java', and 'package-info.java'. A red box highlights the 'Generated Sources (jax-ws)' folder, and a red arrow points to it from the text 'Generata la cartella Generated Sources (jax-ws)'. Under 'Web Service References', there is a 'CardValidator' reference, which includes a 'CardValidatorPort' and a 'validate' operation.

CreditCardValidator Web Service

- CardValidator rappresenta l'istanza generata dal file WSDL con wsimport



The screenshot shows the same project structure as the previous one. Under 'Generated Sources (jax-ws)', the 'simpleclient' package contains the same Java files. A red box highlights the '@CardValidator.java' file, and a red arrow points to it from the text 'CardValidator rappresenta l'istanza generata dal file WSDL con wsimport'. The 'Web Service References' section remains the same, showing the 'CardValidator' reference with its 'CardValidatorPort' and 'validate' operation.

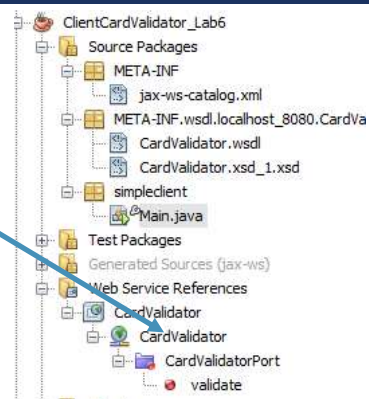
Il client: invoking programmatically

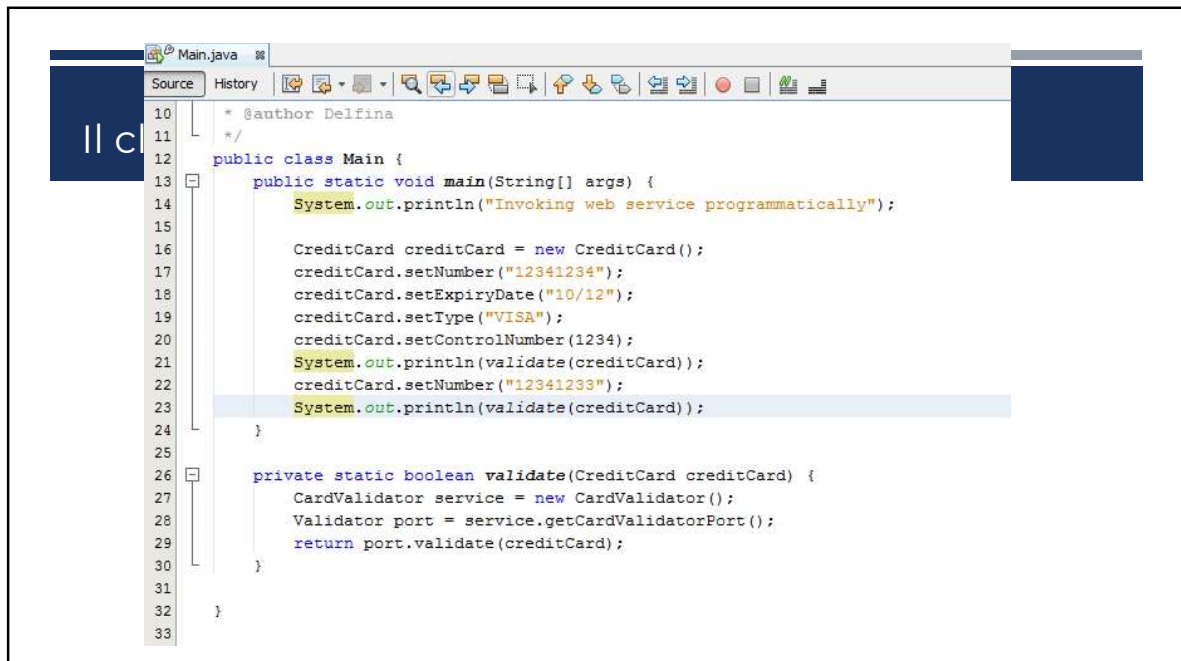
- Il consumer è esterno al container, bisogna invocarlo da codice
- Il CardValidator web service non invocato direttamente
- Il consumer usa una istanza di CardValidator (generata dal WSDL grazie a wsimport) usando la keyword `new`
- Deve a questo punto ottenere il proxy CardValidator class (getCardValidatorPort()) per invocare il metodo di business localmente
- Una chiamata viene fatta sul metodo `validate()` del proxy che a sua volta invocherà il web service remote
 - Creando una richiesta SOAP
 - Facendo il marshalling del credit card messages, ecc

```
private static boolean validate(CreditCard creditCard) {
    CardValidator service = new CardValidator();
    Validator port = service.getCardValidatorPort();
    return port.validate(creditCard);
}
```

Il client: invoking programmatically

- Drag & Drop da qui nella classe del client





```

10  * @author Delfina
11  */
12  public class Main {
13      public static void main(String[] args) {
14          System.out.println("Invoking web service programmatically");
15
16          CreditCard creditCard = new CreditCard();
17          creditCard.setNumber("12341234");
18          creditCard.setExpiryDate("10/12");
19          creditCard.setType("VISA");
20          creditCard.setControlNumber(1234);
21          System.out.println(validate(creditCard));
22          creditCard.setNumber("12341233");
23          System.out.println(validate(creditCard));
24      }
25
26      private static boolean validate(CreditCard creditCard) {
27          CardValidator service = new CardValidator();
28          Validator port = service.getCardValidatorPort();
29          return port.validate(creditCard);
30      }
31
32  }
33

```

II client: invoking programmatically



```

Output  Test Results
Java DB Database Process  GlassFish Server  Retriever Output  ClientCardValidator_Lab6 (run-single)

deps-jar:
Updating property file: C:\Users\Delfina\Documents\NetBeansProjects\ClientCardValida:
wsimport-init:
wsimport-client-CardValidator:
files are up to date
wsimport-client-generate:
Compiling 1 source file to C:\Users\Delfina\Documents\NetBeansProjects\ClientCardVa:
compile-single:
run-single:
Invoking web service programmatically
true
false
BUILD SUCCESSFUL (total time: 4 seconds)

```


Organizzazione della lezione

- WS in Java
 - WSDL Mapping
 - Eccezioni e Fault
 - Contesto e ciclo di vita
- Putting It All Together
 - Un esempio riassuntivo
- Supporto ai WS in Netbeans
 - Il progetto per WS
 - Testing
 - WS Client
- Conclusioni

