TDM SOAP – Correction

serveurEntiers

CalculatriceWebService.java | CalculatriceWSImpl.java | CalculatriceWSPublisher.java | EntierNegatif.java

```java
package     calculatriceWebService;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.soap.SOAPBinding;

@WebService(name="CalculatriceWebService")
@SOAPBinding(style=SOAPBinding.Style.RPC)
public interface CalculatriceWebService {
   @WebMethod(action="additionner")
   public int additionner(@WebParam(name="z1") int z1, @WebParam(name="z2") int z2) throws EntierNegatif;
   @WebMethod(action="soustraire")
   public int soustraire(@WebParam(name="z1") int z1, @WebParam(name="z2") int z2) throws EntierNegatif;
   @WebMethod(action="multiplier")
   public int multiplier(@WebParam(name="z1") int z1, @WebParam(name="z2") int z2) throws EntierNegatif;
   @WebMethod(action="diviser")
   public int diviser(@WebParam(name="z1") int z1, @WebParam(name="z2") int z2) throws EntierNegatif;
}
```

CalculatriceWSImpl.java | CalculatriceWebService.java | CalculatriceWSPublisher.java | EntierNegatif.java

```java
package calculatriceWebService;

import javax.jws.WebService;

@WebService(endpointInterface="calculatriceWebService.CalculatriceWebService")
public class CalculatriceWSImpl implements CalculatriceWebService {

  @Override
  public int additionner(int e1, int e2) throws EntierNegatif {
    int resultat = e1+e2;
    if (e1<0 || e2<0)
      throw new EntierNegatif();
    else
      return resultat;
   }

  @Override
   public int soustraire(int e1, int e2) throws EntierNegatif {
    int resultat = e1-e2;
    if (e1<0 || e2<0 || resultat<0)
      throw new EntierNegatif();
    else
      return resultat;
   }

  @Override
   public int multiplier(int e1, int e2) throws EntierNegatif {
    int resultat = e1*e2;
    if (e1<0 || e2<0)
      throw new EntierNegatif();
    else
      return resultat;
   }

  @Override
   public int diviser(int e1, int e2) throws EntierNegatif {
    int resultat = e1/e2;
    if (e1<0 || e2<0)
      throw new EntierNegatif();
    else
      return resultat;
   }
}
```

EntierNegatif.java | CalculatriceWSImpl.java | CalculatriceWebService.java | CalculatriceWSPublisher.java

```java
package calculatriceWebService;

public class EntierNegatif extends Exception {
  public EntierNegatif(){
    super("Entier non positif !");
  }
}
```

CalculatriceWSPublisher.java | EntierNegatif.java | CalculatriceWSImpl.java | CalculatriceWebService.java

```java
package calculatriceWebService;

import javax.xml.ws.Endpoint;

public class CalculatriceWSPublisher {
    public static void main(String[ ] args) {
        int port = 8888;
        String url = "http://localhost:" + port + "/calculatriceWS";
        System.out.println("Publishing CalculatriceWS on port " + port);
        Endpoint.publish(url, new CalculatriceWSImpl());
    }
}
```

Mise en œuvre

- Compilation classique
  `javac -d . *.java`
- Utilisation de `wsgen` pour pour générer les classes et fichiers requis pour l'exécution du service web
  `wsgen -cp . calculatriceWebService.CalculatriceWSImpl`
- Publication du service
  `java calculatriceWebService.CalculatriceWSPublisher`

serveurComplexes

CalculatriceWebService.java | CalculatriceWSImpl.java | CalculatriceWSPublisher.java | Complexe.java

```java
package calculatriceComplexesWebService;

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.soap.SOAPBinding;

@WebService(name="CalculatriceComplexesWebService")
@SOAPBinding(style=SOAPBinding.Style.RPC)
public interface CalculatriceWebService {
  @WebMethod(action="additionner")
  public Complexe additionner(@WebParam(name="c1") Complexe c1, @WebParam(name="c2") Complexe c2);
  @WebMethod(action="soustraire")
  public Complexe soustraire(@WebParam(name="c1") Complexe c1, @WebParam(name="c2") Complexe c2);
  @WebMethod(action="multiplier")
  public Complexe multiplier(@WebParam(name="c1") Complexe c1, @WebParam(name="c2") Complexe c2);
  @WebMethod(action="diviser")
  public Complexe diviser(@WebParam(name="c1") Complexe c1, @WebParam(name="c2") Complexe c2);
}
```

CalculatriceWSImpl.java | CalculatriceWebService.java | CalculatriceWSPublisher.java | Complexe.java

```java
package calculatriceComplexesWebService;

import javax.jws.WebService;

@WebService(endpointInterface="calculatriceComplexesWebService.CalculatriceWebService")
public class CalculatriceWSImpl implements CalculatriceWebService {
  @Override
  public Complexe additionner(Complexe c1, Complexe c2) {
    return new Complexe(c1.getRe()+c2.getRe(), c1.getIm()+c2.getIm());
  }

  @Override
  public Complexe soustraire(Complexe c1, Complexe c2) {
    return new Complexe(c1.getRe()-c2.getRe(), c1.getIm()-c2.getIm());
  }

  @Override
  public Complexe multiplier(Complexe c1, Complexe c2) {
    return new Complexe(c1.getRe()*c2.getRe()-c1.getIm()*c2.getIm(), c1.getRe()*c2.getIm()+c1.getIm()*c2.getRe());
  }

  @Override
  public Complexe diviser(Complexe c1, Complexe c2) {
    float diviseur = c2.getRe()*c2.getRe()+c2.getIm()*c2.getIm();
    if(diviseur == 0)
    throw new ArithmeticException("Division par zero !");
    return new Complexe((c1.getRe()*c2.getRe()+c1.getIm()*c2.getIm())/diviseur, (c1.getIm()*c2.getRe()-c1.getRe()*c2.getIm())/diviseur);
  }
}
```

Complexe.java | CalculatriceWSImpl.java | CalculatriceWebService.java | CalculatriceWSPublisher.java

```java
package calculatriceComplexesWebService;

public class Complexe {
  private float re,im;

  public Complexe() {
    this.re = (float)0;
    this.im = (float)0;   }

  public Complexe(int re,int im) {
    this.re = (float)re;
    this.im = (float)im;   }

  public Complexe(float re,float im) {
    this.re = re;
    this.im = im;   }

  public float getRe() {
    return this.re;   }

  public void setRe(float re) {
    this.re = re;   }

  public float getIm() {
    return this.im;   }

  public void setIm(float im) {
    this.im = im;   }

  public String toString(){
    String resultat="";
    if (this.re!=0)
      resultat=String.valueOf(this.re);
    if (this.im!=0) {
      if (this.im>0){
    if (resultat!="")
      resultat=resultat+"+"+String.valueOf(this.im);
    else
      resultat=String.valueOf(this.im);
      }else
    resultat=resultat+String.valueOf(this.im);
      resultat=resultat+"i";     }
    return resultat;   }
}
```

CalculatriceWSPublisher.java

```java
package calculatriceComplexesWebService;

import javax.xml.ws.Endpoint;

public class CalculatriceWSPublisher {
    public static void main(String[ ] args) {
        int port = 8880;
        String url = "http://localhost:" + port + "/calculatriceWSComplexe";
        System.out.println("Publishing CalculatriceWSComplexe on port " + port);
        Endpoint.publish(url, new CalculatriceWSImpl());
    }
}
```

Mise en œuvre

- Compilation classique
  `javac -d . *.java`
- Utilisation de `wsgen` pour pour générer les classes et fichiers requis pour l'exécution du service web
  `wsgen -cp . calculatriceComplexesWebService.CalculatriceWSImpl`
- Publication du service
  `java calculatriceComplexesWebService.CalculatriceWSPublisher`

Clients

Préparation

Il faut générer le code nécessaire aux client à partir des serveurs respectifs

```
wsimport -p calculatriceWebService -keep http://localhost:8888/calculatriceWS?wsdl
```

```
wsimport -p calculatriceComplexesWebService -keep
                                    http://localhost:8880/calculatriceWSComplexe?wsdl
```

Puis écrire le code des clients, compiler et exécuter

ClientEntiersPositifs.java ClientComplexes.java

```java
import calculatriceWebService.CalculatriceWebService;


public class ClientEntiersPositifs {
  public static void main(String[] args) {
    try {
      CalculatriceWebService calculatrice = (new calculatriceWebService.CalculatriceWSImplService()).getCalculatriceWSImplPort();
      int operande1 = Integer.parseInt(args[args.length-3]),
          operande2 = Integer.parseInt(args[args.length-1]),
          resultat=-1;

      if(args[args.length-2].equals("+"))
        resultat = calculatrice.additionner(operande1, operande2);
      if(args[args.length-2].equals("-"))
        resultat = calculatrice.soustraire(operande1, operande2);
      if(args[args.length-2].equals("x"))
        resultat = calculatrice.multiplier(operande1, operande2);
      if(args[args.length-2].equals("/"))
        resultat = calculatrice.diviser(operande1, operande2);

      System.out.println(operande1 + args[args.length-2] + operande2 + " = " + resultat);
    } catch (Exception e) {
       System.out.println("Client exception: " +e);
    }
  }
}

```

ClientComplexes.java ClientEntiersPositifs.java

```java
import calculatriceComplexesWebService.CalculatriceComplexesWebService;
import calculatriceComplexesWebService.Complexe;


public class ClientComplexes {
  public static void main(String[] args) throws Exception  {
    try {
      Complexe c1=new Complexe(), c2=new Complexe(), un=new Complexe(), zero=new Complexe(), result;
      c1.setRe(2); c1.setIm(4);
      c2.setRe(1); c2.setIm(2);
      un.setRe(1); un.setIm(0);
      zero.setRe(0); zero.setIm(0);
      CalculatriceComplexesWebService calculatrice = (new calculatriceComplexesWebService.CalculatriceWSImplService()).getCalculatriceWSImplPort();

      result = calculatrice.additionner(c1,c2);
      System.out.println("(2+4i) + (1+2i) = " + result.getRe() + " " + result.getIm() + "i");

      result = calculatrice.soustraire(c1,c2);
      System.out.println("(2+4i) - (1+2i) = " + result.getRe() + " " + result.getIm() + "i");

      result = calculatrice.multiplier(c1,c2);
      System.out.println("(2+4i) x (1+2i) = " + result.getRe() + " " + result.getIm() + "i");

      result = calculatrice.diviser(c1,c2);
      System.out.println("(2+4i) / (1+2i) = " + result.getRe() + " " + result.getIm() + "i");

      result = calculatrice.diviser(c1,un);
      System.out.println("(2+4i) / 1 = " + result.getRe() + " " + result.getIm() + "i");

      result = calculatrice.diviser(c1,zero);
      System.out.println("(2+4i) / 0 = " + result.getRe() + " " + result.getIm() + "i");
    } catch (Exception e) {
       System.out.println("Client exception: " +e);
    }
  }
}

```