

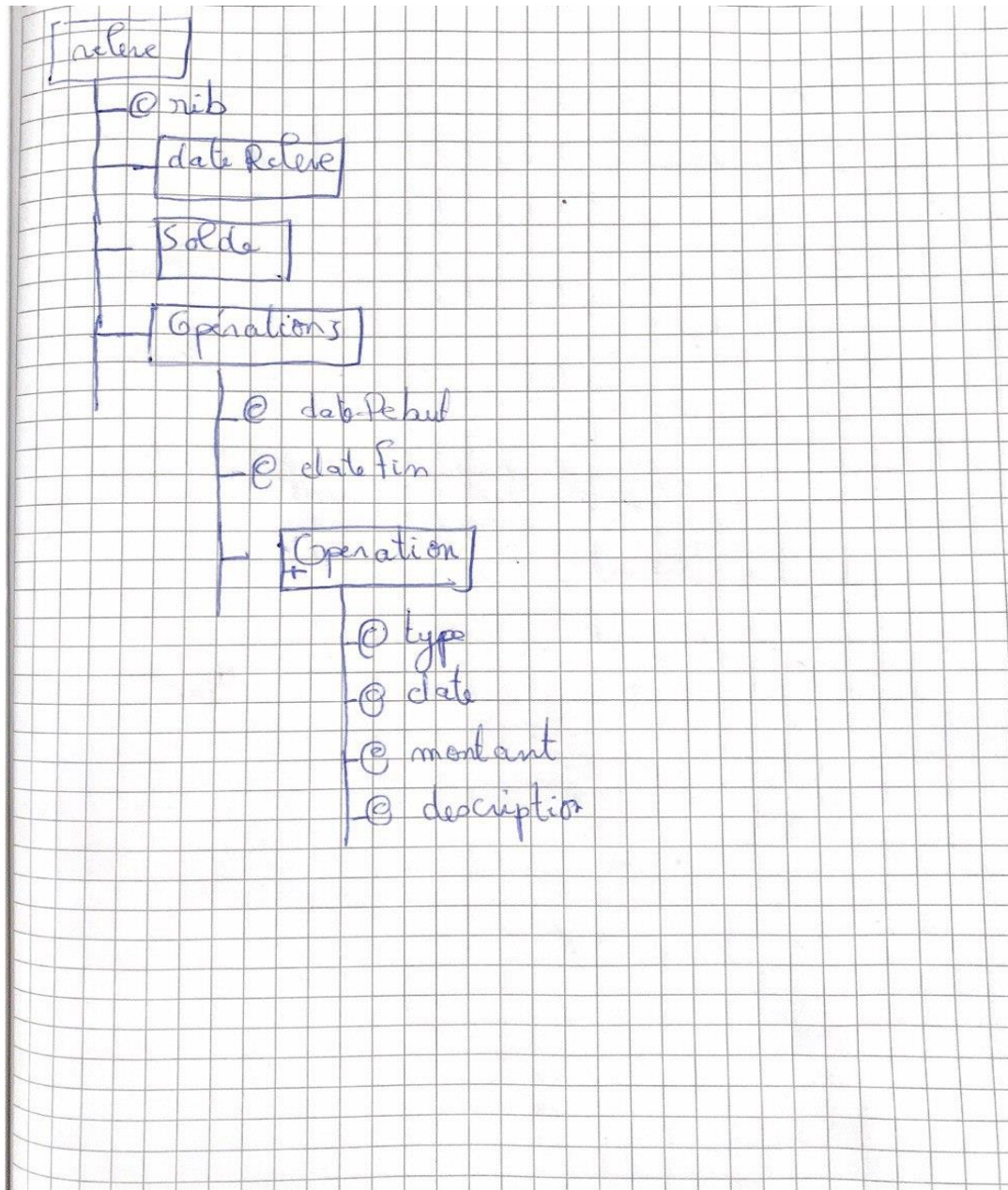
Rapport Projet Technologie XML et Web Services

Introduction

Dans ce projet, on souhaite créer une application qui permet de gérer les relevés de comptes bancaires.

A) Partie Technologie XML

1) Structure graphique de l'arbre XML



2) DTD

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!ELEMENT releve (dateReleve, solde, operations) >
3 <!ELEMENT dateReleve (#PCDATA) >
4 <!ELEMENT solde (#PCDATA) >
5 <!ELEMENT operations (operation+) >
6 <!ELEMENT operation (#PCDATA) >
7
8 <!-->
9 <!-->
10
11 <!-->
12 <!-->
13 <!-->
14
15 <!-->
16 <!-->
17 <!-->
18 <!-->
19 <!-->
```

- Document XML validé par la DTD

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE releve SYSTEM "file:/C:/Users/Baba%20Prince/Desktop/Examen%20Technologie%20XML%20-%20Web%20Services/Te
3 <releve rib="011112222333344445555666">
4 <dateReleve>2021-11-10</dateReleve>
5 <solde>14500</solde>
6 <operations dateDebut="2021-01-01" dateFin="2021-01-30">
7 <operation type="CREDIT" date="2021-01-01" montant="9000" description="Vers Espèce"></operation>
8 </operations>
9 </releve>
```

3) XSD

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
3 <xsd:element name="releve">
4 <xsd:complexType>
5 <xsd:sequence>
6 <xsd:element name="dateReleve" type="xsd:date"/></xsd:element>
7 <xsd:element name="solde" type="xsd:double"/></xsd:element>
8 <xsd:element name="operations" type="OPS"/></xsd:element>
9 </xsd:sequence>
10 <xsd:attribute name="rib" type="xsd:string" use="required"/></xsd:attribute>
11 </xsd:complexType>
12 </xsd:element>
13
14 <xsd:complexType name="OPS">
15 <xsd:sequence>
16 <xsd:element name="operation" type="OP" maxOccurs="unbounded"/></xsd:element>
17 </xsd:sequence>
18 <xsd:attribute name="dateDebut" type="xsd:date" use="required"/></xsd:attribute>
19 <xsd:attribute name="dateFin" type="xsd:date" use="required"/></xsd:attribute>
20 </xsd:complexType>
21
22 <xsd:complexType name="OP">
23 <xsd:attribute name="type" use="required">
```

```
23 <xsd:attribute name="type" use="required">
24 <xsd:simpleType>
25 <xsd:restriction base="xsd:string">
26 <xsd:enumeration value="credit"/></xsd:enumeration>
27 <xsd:enumeration value="debit"/></xsd:enumeration>
28 </xsd:restriction>
29 </xsd:simpleType>
30 </xsd:attribute>
31 <xsd:attribute name="date" type="xsd:date" use="required"/></xsd:attribute>
32 <xsd:attribute name="montant" type="xsd:double" use="required"/></xsd:attribute>
33 <xsd:attribute name="description" use="required">
34 <xsd:simpleType>
35 <xsd:restriction base="xsd:string">
36 <xsd:enumeration value="vers espèce"/></xsd:enumeration>
37 <xsd:enumeration value="chèque guichet"/></xsd:enumeration>
38 <xsd:enumeration value="prélèvement assurance"/></xsd:enumeration>
39 <xsd:enumeration value="virement"/></xsd:enumeration>
40 </xsd:restriction>
41 </xsd:simpleType>
42 </xsd:attribute>
43 </xsd:complexType>
44 </xsd:schema>
45
```

- Document XML validé par la XSD

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <releve xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:noNamespaceSchemaLocation="file:/C:/Users/Baba%20Prince/Desktop/Examen%20Technologie%20XML%20-%20Web%20Services/Technologie%2
4   rib="01111222333344445555666">
5   <dateReleve>2021-11-10</dateReleve>
6   <solde>14500</solde>
7   <operations dateDebut="2021-01-01" dateFin="2021-01-30">
8     <operation type="credit" date="2021-01-01" montant="9000" description="vers espèce"/>
9     <operation type="credit" date="2021-02-02" montant="5000" description="vers espèce"/>
10    <operation type="debit" date="2021-03-02" montant="7000" description="chèque guichet"/>
11  </operations>
12 </releve>

```

4) XSL 1

Cette feuille de style permet d'afficher les toutes les données de ce document XML au format HTML en affichant le total des opérations de débit et le total des opérations de crédit.

```

7 <xsl:template match="/">
8   <html>
9     <head>
10      <title>Relevé</title>
11    </head>
12    <body>
13      <h1>Relevés</h1>
14      <xsl:for-each select="releve">
15        <p>Date relevé : <xsl:value-of select="dateReleve"/></p>
16        <p>RIB : <xsl:value-of select="@rib"/></p>
17        <p>Solde <xsl:value-of select="solde"/></p>
18        <table border="2">
19          <caption>Somme des Opérations</caption>
20          <thead>
21            <th>Type d'opération</th>
22            <th>Date</th>
23            <th>Montant</th>
24            <th>Description</th>
25          </thead>
26          <tbody>
27            <xsl:for-each select="operations">
28              <xsl:for-each select="operation">
29                <tr>
30                  <td><xsl:value-of select="@type"/></td>
31                  <td><xsl:value-of select="@date"/></td>
32                  <td><xsl:value-of select="@montant"/></td>
33                  <td><xsl:value-of select="@description"/></td>
34                </tr>
35              </xsl:for-each>
36              <tr>
37                <td colspan="2">Total des opérations de crédit</td>
38                <td colspan="2" align="center"><xsl:value-of select="sum(operation[@type='credit']/@montant)"/></td>
39              </tr>
40              <tr>
41                <td colspan="2">Total des opérations de débit</td>
42                <td colspan="2" align="center"><xsl:value-of select="sum(operation[@type='debit']/@montant)"/></td>
43              </tr>
44            </xsl:for-each>
45          </tbody>
46        </table>
47      </xsl:for-each>
48    </body>
49  </template>

```

- Résultat de la feuille de style 1

Relevés

Date relevé : 2021-11-10

RIB : 011112222333344445555666

Solde 14500

Somme des Opérations

Type d'opération	Date	Montant	Description
credit	2021-01-01	9000	vers espèce
credit	2021-02-02	5000	vers espèce
debit	2021-03-02	7000	chèque guichet
Total des operations de crédit		14000	
Total des operations de débit		7000	

5) XSL 2

Cette feuille de style permet d'afficher au format HTML les opérations de type CREDIT d'un relevé bancaire.

```
6 <xsl:template match="/">
7   <html>
8     <head>
9       <title>Relevé</title>
10    </head>
11    <body>
12      <h1>Relevés</h1>
13      <xsl:for-each select="releve">
14        <p>Date relevé : <xsl:value-of select="dateReleve"></xsl:value-of> </p>
15        <p>RIB : <xsl:value-of select="@rib"></xsl:value-of> </p>
16        <p>Solde <xsl:value-of select="solde"></xsl:value-of> </p>
17        <table border="2">
18          <caption>Somme des Opérations</caption>
19          <thead>
20            <th>Type d'opération</th>
21            <th>Date</th>
22            <th>Montant</th>
23            <th>Description</th>
24          </thead>
25          <tbody>
26            <xsl:for-each select="operations">
27              <xsl:for-each select="operation[@type='credit']">
28                <tr>
29                  <td><xsl:value-of select="@type"></xsl:value-of> </td>
30                  <td><xsl:value-of select="@date"></xsl:value-of> </td>
31                  <td><xsl:value-of select="@montant"></xsl:value-of> </td>
32                  <td><xsl:value-of select="@description"></xsl:value-of> </td>
33                </tr>
34              </xsl:for-each>
35              <tr>
36                <td colspan="2">Total des opérations de crédit</td>
37                <td colspan="2" align="center"><xsl:value-of select="sum(operation[@type='credit']/@montant)"></xsl:value-of> </td>
38              </tr>
39            </xsl:for-each>
40          </tbody>
41        </table>
42      </xsl:for-each>
43    </body>
44  </html>
```


- Resultat de la feuille de style 2

Relevés

Date relevé : 2021-11-10

RIB : 011112222333344445555666

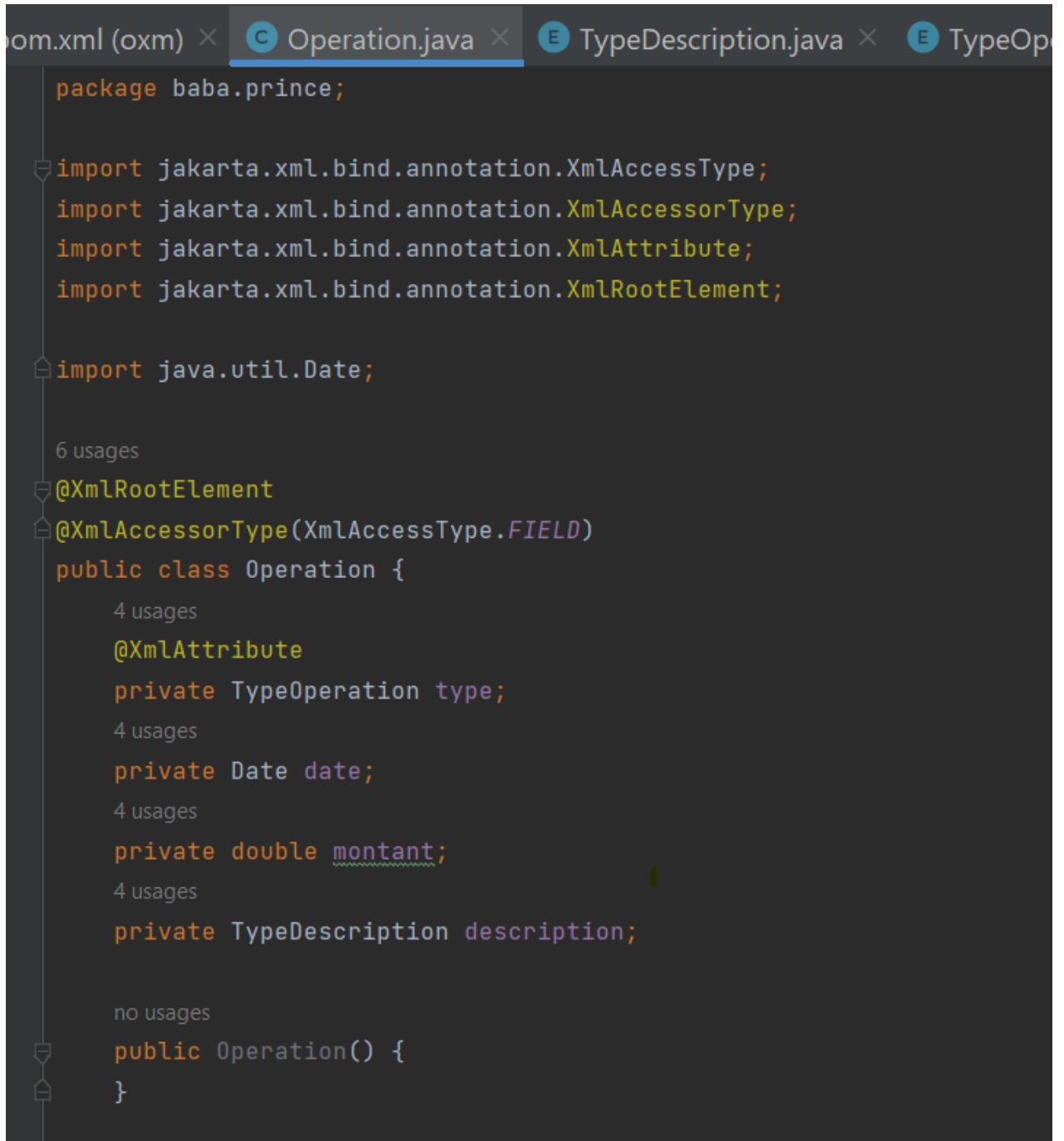
Solde 14500

Somme des Opérations

Type d'opération	Date	Montant	Description
credit	2021-01-01	9000	vers espèce
credit	2021-02-02	5000	vers espèce
Total des operations de crédit		14000	

B) Partie Mapping Objet XML avec Jax Binding

1) Classe Operation



```
om.xml (oxm) × C Operation.java × E TypeDescription.java × E TypeOp

package baba.prince;

import jakarta.xml.bind.annotation.XmlAccessType;
import jakarta.xml.bind.annotation.XmlAccessorType;
import jakarta.xml.bind.annotation.XmlAttribute;
import jakarta.xml.bind.annotation.XmlRootElement;

import java.util.Date;

6 usages
@XmlRootElement
@XmlAccessorType(XmlAccessType.FIELD)
public class Operation {

    4 usages
    @XmlAttribute
    private TypeOperation type;
    4 usages
    private Date date;
    4 usages
    private double montant;
    4 usages
    private TypeDescription description;

    no usages
    public Operation() {
    }
}
```



```
m.xml (oxm) × C Operation.java × E TypeDescription.java × E TypeOperation.java × C Releve.java × C  
4 usages  
public Operation(TypeOperation type, Date date, double montant, TypeDescription description) {  
    this.type = type;  
    this.date = date;  
    this.montant = montant;  
    this.description = description;  
}  
  
no usages  
public TypeOperation getType() {  
    return type;  
}  
  
no usages  
public void setType(TypeOperation type) {  
    this.type = type;  
}  
  
no usages  
public Date getDate() {  
    return date;  
}  
  
no usages  
public void setDate(Date date) {  
    this.date = date;  
}
```

```
src / main / java / base / price / Operation / testing
pom.xml (oxm) × C Operation.java × E TypeDescription.java × E TypeOpe

public double getMontant() {
    return montant;
}

no usages
public void setMontant(double montant) {
    this.montant = montant;
}

no usages
public TypeDescription getDescription() {
    return description;
}

no usages
public void setDescription(TypeDescription description) {
    this.description = description;
}

@Override
public String toString() {
    return "Operation{" +
        "type=" + type +
        ", date=" + date +
        ", montant=" + montant +
        ", description=" + description +
        '}';
}
}
```

2) Classe Relève

```
pom.xml (oxm) × C Operation.java × E TypeDescription.java ×
package baba.prince;

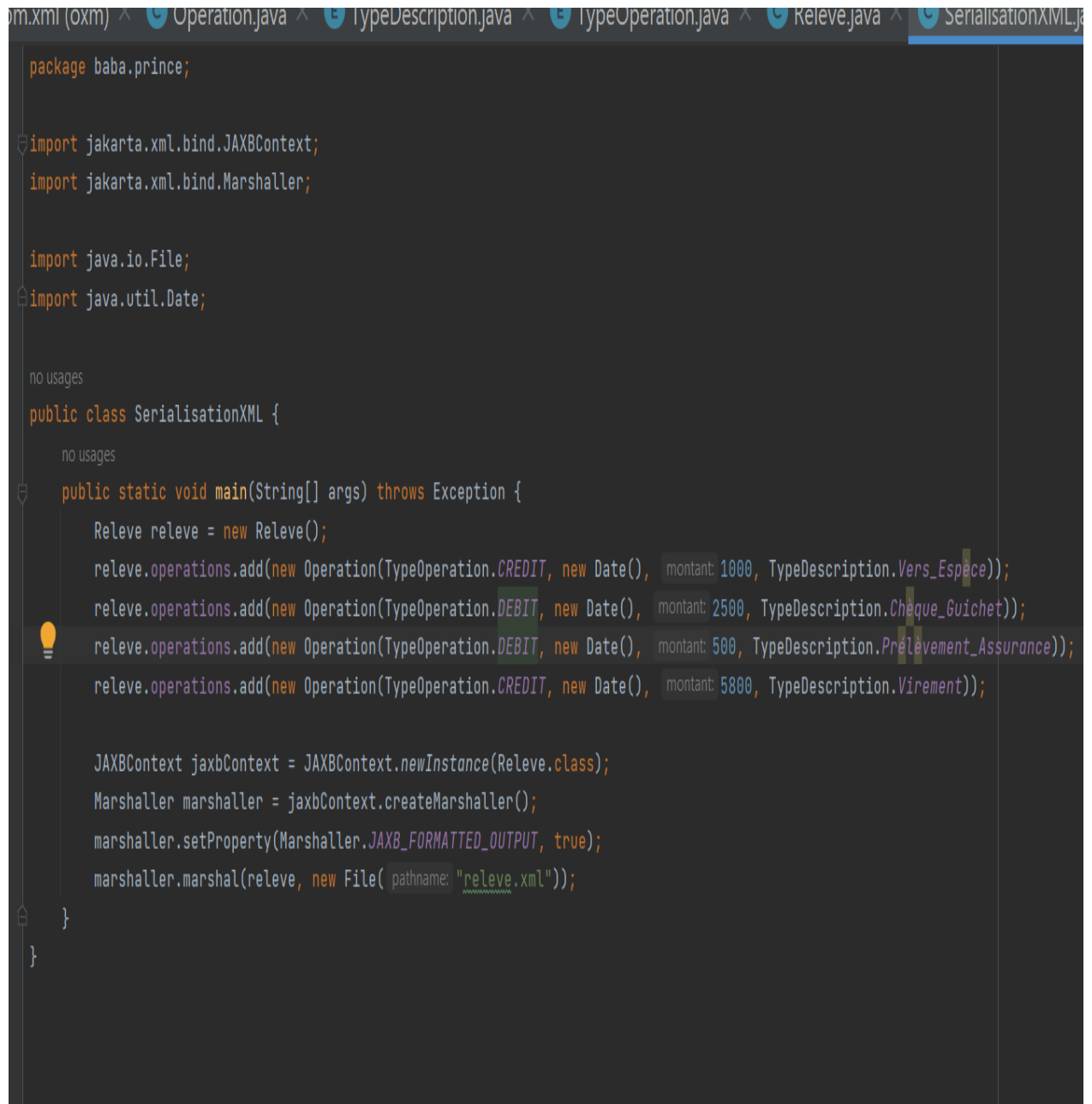
import jakarta.xml.bind.annotation.XmlElement;
import jakarta.xml.bind.annotation.XmlRootElement;

import java.util.ArrayList;
import java.util.List;

7 usages
@XmlRootElement
public class Relève {
    5 usages
    @XmlElement(name = "operation")
    public List<Operation> operations = new ArrayList<>();
}
```

3) Classe Serialisation

Cette classe permet de créer un objet « Releve » contenant une liste d'opérations et de sérialiser ces données dans un fichier XML



```
package baba.prince;

import jakarta.xml.bind.JAXBContext;
import jakarta.xml.bind.Marshaller;

import java.io.File;
import java.util.Date;

no usages
public class SerialisationXML {
    no usages
    public static void main(String[] args) throws Exception {
        Releve releve = new Releve();
        releve.operations.add(new Operation(TypeOperation.CREDIT, new Date(), montant: 1000, TypeDescription.Vers_Espece));
        releve.operations.add(new Operation(TypeOperation.DEBIT, new Date(), montant: 2500, TypeDescription.Cheque_Guichet));
        releve.operations.add(new Operation(TypeOperation.DEBIT, new Date(), montant: 500, TypeDescription.Prélèvement_Assurance));
        releve.operations.add(new Operation(TypeOperation.CREDIT, new Date(), montant: 5800, TypeDescription.Virement));

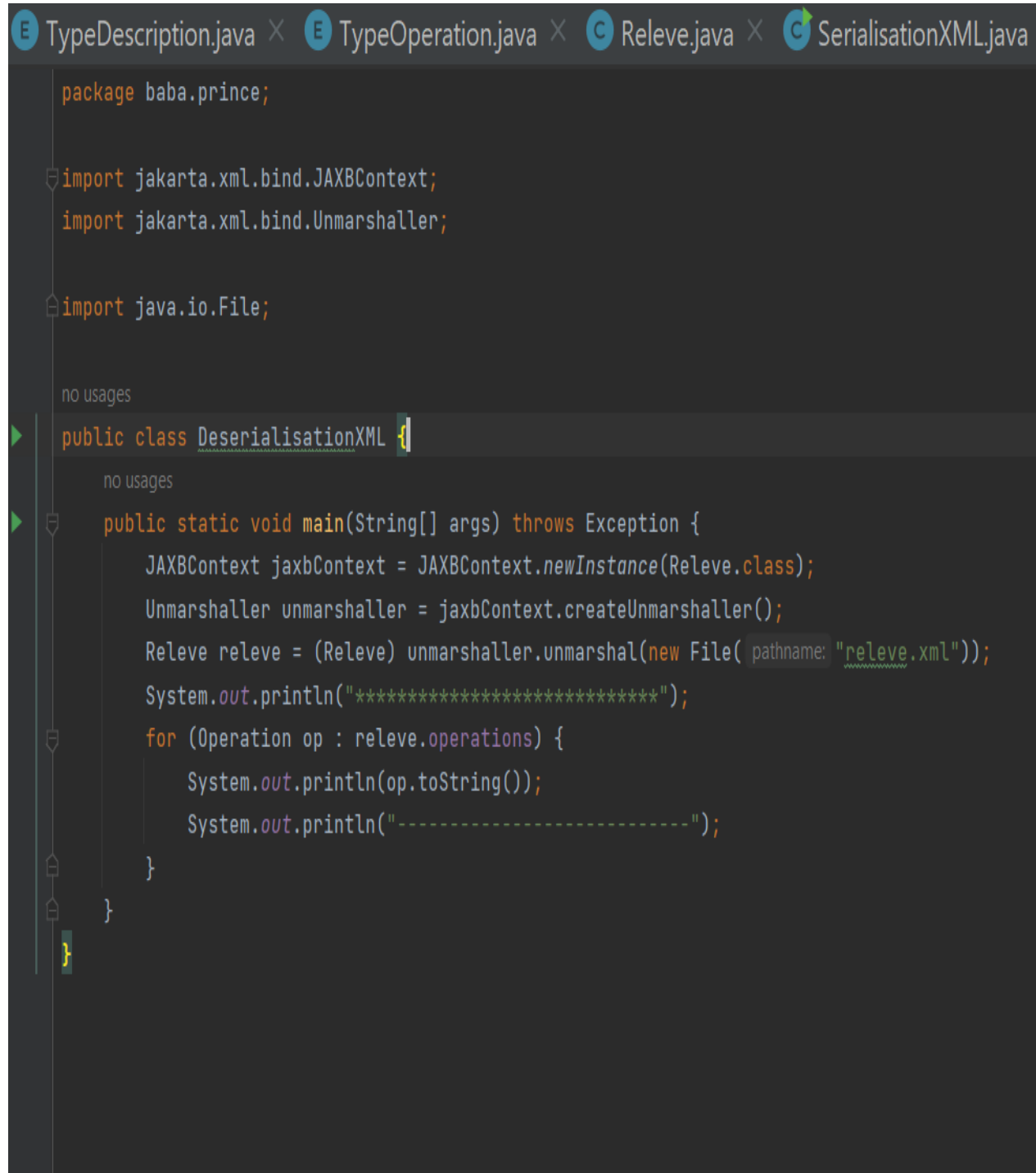
        JAXBContext jaxbContext = JAXBContext.newInstance(Releve.class);
        Marshaller marshaller = jaxbContext.createMarshaller();
        marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
        marshaller.marshal(releve, new File( pathname: "releve.xml"));
    }
}
```

- Fichier XML contenant les données de la sérialisation

```
option.java × E TypeOperation.java × C Releve.java × C Se
|<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
|<releve>
|   <operation type="CREDIT">
|       <date>2022-12-28T20:10:37.299+01:00</date>
|       <montant>1000.0</montant>
|       <description>Vers_Espèce</description>
|   </operation>
|   <operation type="DEBIT">
|       <date>2022-12-28T20:10:37.300+01:00</date>
|       <montant>2500.0</montant>
|       <description>Chèque_Guichet</description>
|   </operation>
|   <operation type="DEBIT">
|       <date>2022-12-28T20:10:37.300+01:00</date>
|       <montant>500.0</montant>
|       <description>Prélèvement_Assurance</description>
|   </operation>
|   <operation type="CREDIT">
|       <date>2022-12-28T20:10:37.300+01:00</date>
|       <montant>5800.0</montant>
|       <description>Virement</description>
|   </operation>
|</releve>
```

4) Classe Deserialisation

Cette classe permet de lire et d'afficher les données du relevé du fichier XML généré précédemment.



```
package baba.prince;

import jakarta.xml.bind.JAXBContext;
import jakarta.xml.bind.Unmarshaller;

import java.io.File;

no usages

public class DeserialisationXML {

no usages

    public static void main(String[] args) throws Exception {
        JAXBContext jaxbContext = JAXBContext.newInstance(Releve.class);
        Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();
        Releve releve = (Releve) unmarshaller.unmarshal(new File("releve.xml"));
        System.out.println("*****");
        for (Operation op : releve.operations) {
            System.out.println(op.toString());
            System.out.println("-----");
        }
    }
}
```

- **Résultat de la désérialisation**

```
DeserialisationXML x
"C:\Program Files\Java\jdk-19\bin\java.exe" ...
*****
Operation{type=CREDIT, date=Wed Dec 28 20:10:37 WEST 2022, montant=1000.0, description=Vers_Espèce}
-----
Operation{type=DEBIT, date=Wed Dec 28 20:10:37 WEST 2022, montant=2500.0, description=Chèque_Guichet}
-----
Operation{type=DEBIT, date=Wed Dec 28 20:10:37 WEST 2022, montant=500.0, description=Prélèvement_Assurance}
-----
Operation{type=CREDIT, date=Wed Dec 28 20:10:37 WEST 2022, montant=5800.0, description=Virement}
-----
Process finished with exit code 0
|
```

5) Classe GenerateXMLShcema

Cette classe permet de générer le schema XML representant la structure d'un releve.

```
ption.java x E TypeOperation.java x C Releve.java x C SerialisationXML.java x
package baba.prince;

import jakarta.xml.bind.JAXBContext;
import jakarta.xml.bind.SchemaOutputResolver;

import javax.xml.transform.Result;
import javax.xml.transform.stream.StreamResult;
import java.io.File;
import java.io.IOException;

no usages
public class GenerateXMLSchema {
    no usages
    public static void main(String[] args) throws Exception {
        JAXBContext jaxbContext = JAXBContext.newInstance(Releve.class);
        jaxbContext.generateSchema(new SchemaOutputResolver() {
            no usages
            @Override
            public Result createOutput(String s, String s1) throws IOException {
                File file = new File( pathname: "releve.xsd");
                StreamResult streamResult =new StreamResult(file);
                streamResult.setSystemId(file.getName());
                return streamResult;
            }
        });
    }
}
```


- Schéma XML représentant la structure d'un relevé

```
TypeOperation.java × Releve.java × SerialisationXML.java × releve.xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="operation" type="operation"/>

  <xs:element name="releve" type="releve"/>

  <xs:complexType name="releve">
    <xs:sequence>
      <xs:element ref="operation" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="operation">
    <xs:sequence>
      <xs:element name="date" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="montant" type="xs:double"/>
      <xs:element name="description" type="typeDescription" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="type" type="typeOperation"/>
  </xs:complexType>

  <xs:simpleType name="typeOperation">
    <xs:restriction base="xs:string">
      <xs:enumeration value="CREDIT"/>
      <xs:enumeration value="DEBIT"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

C) Partie C Web Services

1) Web Services

- Classe Operation

```
package baba.prince;

import jakarta.xml.bind.annotation.XmlAccessType;
import jakarta.xml.bind.annotation.XmlAccessorType;
import jakarta.xml.bind.annotation.XmlAttribute;
import jakarta.xml.bind.annotation.XmlRootElement;

import java.util.Date;

7 usages
public class Operation {
    4 usages
    private TypeOperation type;
    4 usages
    private Date date;
    4 usages
    private double montant;
    4 usages
    private TypeDescription description;

    no usages
    public Operation() {
    }

    4 usages
    public Operation(TypeOperation type, Date date, double montant, TypeDescription description) {
        this.type = type;
        this.date = date;
        this.montant = montant;
        this.description = description;
    }
}
```

```
public TypeOperation getType() {  
    return type;  
}
```

no usages

```
public void setType(TypeOperation type) {  
    this.type = type;  
}
```

no usages

```
public Date getDate() {  
    return date;  
}
```

no usages

```
public void setDate(Date date) {  
    this.date = date;  
}
```

no usages

```
public double getMontant() {  
    return montant;  
}
```

no usages

```
public void setMontant(double montant) {  
    this.montant = montant;  
}
```

no usages

```
public TypeDescription getDescription() {  
    return description;  
}
```

no usages

```
public void setDescription(TypeDescription description) {  
    this.description = description;  
}
```

@Override

```
public String toString() {  
    return "Operation{" +  
        "type=" + type +  
        ", date=" + date +  
        ", montant=" + montant +  
        ", description=" + description +  
        '}';  
}
```

- Classe Releve

```
package baba.prince;

import jakarta.jws.WebService;
import jakarta.xml.bind.annotation.XmlElement;
import jakarta.xml.bind.annotation.XmlRootElement;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;

17 usages
public class Releve {
    3 usages
    private long rib;
    6 usages
    private List<Operation> operations = new ArrayList<>();
    3 usages
    private Date dateReleve;
    3 usages
    private double solde;

    10 usages
    public Releve(long rib, Date dateReleve, double solde) {
        this.rib = rib;
        operations.add(new Operation(TypeOperation.CREDIT, new Date(), montant: Math.random()*9800, TypeDescription.Vers_Espace));
        operations.add(new Operation(TypeOperation.DEBIT, new Date(), montant: Math.random()*5600, TypeDescription.Virement));
        operations.add(new Operation(TypeOperation.CREDIT, new Date(), montant: Math.random()*3500, TypeDescription.Cheque_Guichet));
        operations.add(new Operation(TypeOperation.DEBIT, new Date(), montant: Math.random()*100, TypeDescription.Prelevement_Assurance));
        this.dateReleve = dateReleve;
        this.solde = solde;
    }
}
```

```
xml (ws) × Operation.java × TypeOperation.java × TypeDescription.java × Kc
public long getRib() {
    return rib;
}

no usages
public void setRib(long rib) {
    this.rib = rib;
}

no usages
public List<Operation> getOperations() {
    return operations;
}

no usages
public void setOperations(List<Operation> operations) {
    this.operations = operations;
}

1 usage
public Date getDateReleve() {
    return dateReleve;
}

no usages
public void setDateReleve(Date dateReleve) {
    this.dateReleve = dateReleve;
}

1 usage
public double getSolde() {
    return solde;
}
```

- **Classe ReleveService**

Cette classe un Web services basé sur JaxWS qui permet de consulter un relevé

```
package baba.prince;

import jakarta.jws.WebMethod;
import jakarta.jws.WebParam;
import jakarta.jws.WebService;

import java.util.Date;
import java.util.List;

2 usages
@WebService(serviceName = "ReleveWS")
public class ReleveService {
    no usages
    @WebMethod
    public Releve getReleve(@WebParam(name = "rib") long rib) {
        return new Releve(rib, new Date(), solde: Math.random()*43000);
    }

    no usages
    @WebMethod
    public List<Releve> listReleve() {
        return List.of(
            new Releve(rib: 1, new Date(), solde: Math.random()*2000),
            new Releve(rib: 2, new Date(), solde: Math.random()*87000),
            new Releve(rib: 3, new Date(), solde: Math.random()*2400),
            new Releve(rib: 4, new Date(), solde: Math.random()*9400),
            new Releve(rib: 5, new Date(), solde: Math.random()*8600),
            new Releve(rib: 6, new Date(), solde: Math.random()*4800)
        );
    }
}
```


2) Serveur JaxWS

```
package server;

import baba.prince.ReleveService;
import jakarta.xml.ws.Endpoint;

no usages
public class ServerJWS {
    no usages
    public static void main(String[] args) {
        Endpoint.publish( address: "http://0.0.0.0:9191/", new ReleveService());
        System.out.println("Web Service deployé sur http://0.0.0.0:9191/");
    }
}
```

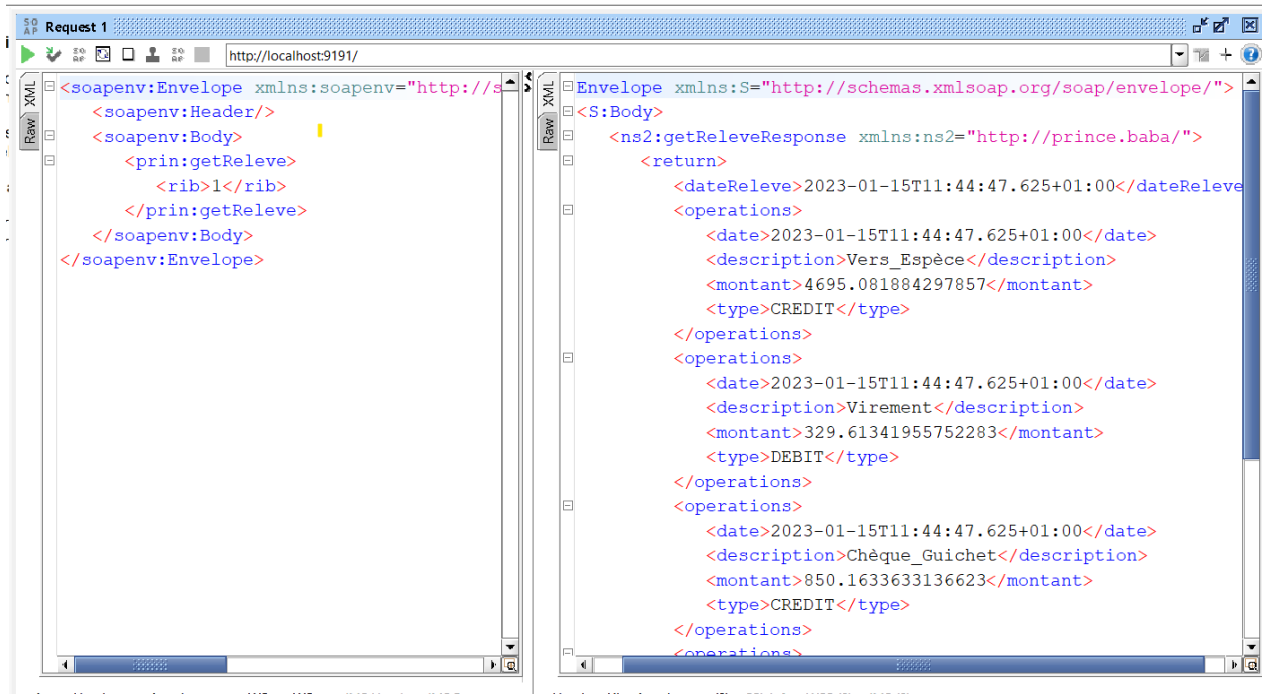
3) WSDL

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

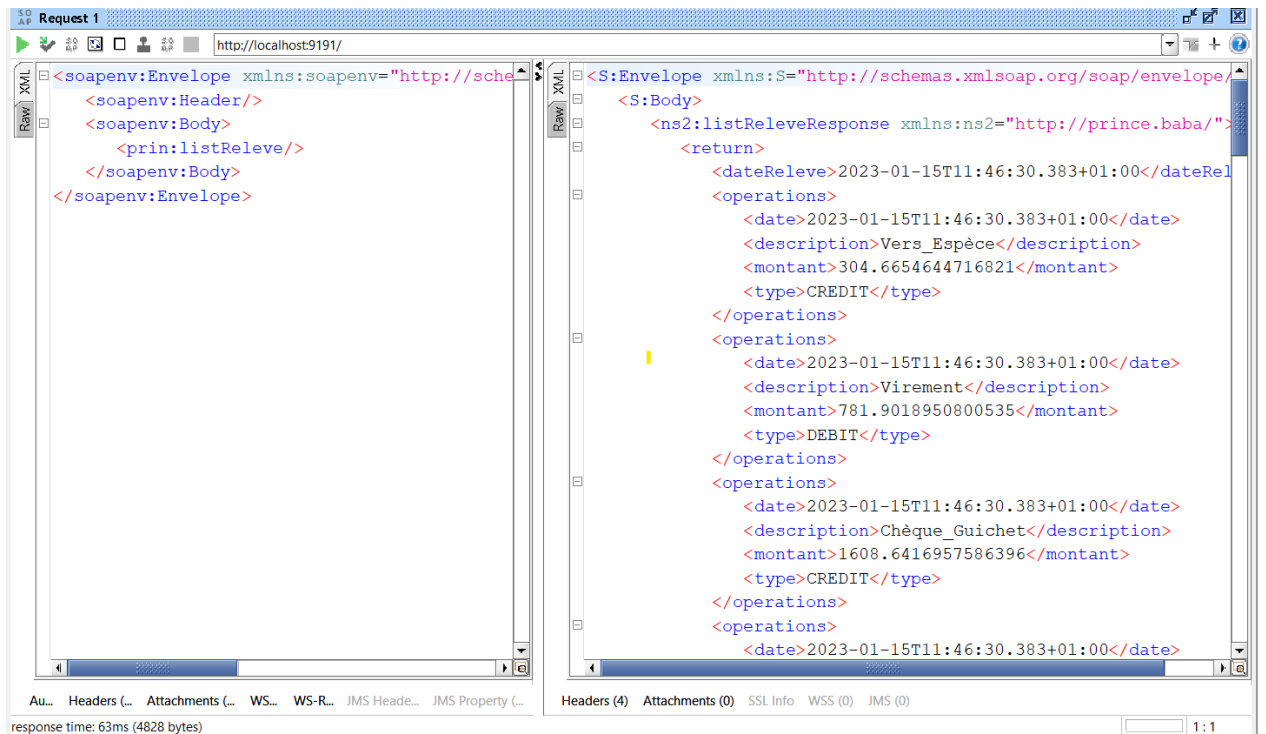
```
<!-- Published by XML-WS Runtime (https://github.com/eclipse-ee4j/metro-jax-ws). Runtime's version is XML-WS Runtime 4.0.0 git-revision#129f787. -->
<!-- Generated by XML-WS Runtime (https://github.com/eclipse-ee4j/metro-jax-ws). Runtime's version is XML-WS Runtime 4.0.0 git-revision#129f787. -->
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-2004-01-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/soap/"
    xmlns:tns="http://prince.baba/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://prince.baba/" name="ReleveWS">
    <types>
        <xsd:schema>
            <xsd:import namespace="http://prince.baba/" schemaLocation="http://localhost:9191/?xsd-1"/>
        </xsd:schema>
    </types>
    <message name="getReleve">
        <part name="parameters" element="tns:getReleve"/>
    </message>
    <message name="getReleveResponse">
        <part name="parameters" element="tns:getReleveResponse"/>
    </message>
    <message name="listReleve">
        <part name="parameters" element="tns:listReleve"/>
    </message>
    <message name="listReleveResponse">
        <part name="parameters" element="tns:listReleveResponse"/>
    </message>
    <portType name="ReleveService">
        <operation name="getReleve">
            <input wsam:Action="http://prince.baba/ReleveService/getReleveRequest" message="tns:getReleve"/>
            <output wsam:Action="http://prince.baba/ReleveService/getReleveResponse" message="tns:getReleveResponse"/>
        </operation>
        <operation name="listReleve">
            <input wsam:Action="http://prince.baba/ReleveService/listReleveRequest" message="tns:listReleve"/>
            <output wsam:Action="http://prince.baba/ReleveService/listReleveResponse" message="tns:listReleveResponse"/>
        </operation>
    </portType>
    <binding name="ReleveServicePortBinding" type="tns:ReleveService">
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
        <operation name="getReleve">
            <soap:operation soapAction="">
                <input>
                    <soap:body use="literal"/>
                </input>
            </soap:operation>
        </operation>
    </binding>
</definitions>
```

4) Test des méthodes avec SoapUI

- Method getReleve



- Method listReleve



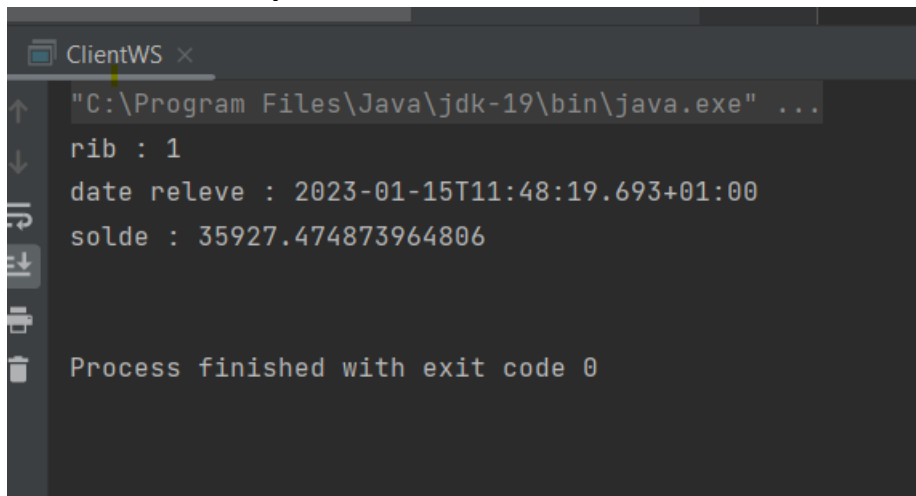
5) Client SOAP Java

- Classe ClientWS

```
import proxy.Operation;
import proxy.Releve;
import proxy.ReleveService;
import proxy.ReleveWS;

no usages
public class ClientWS {
    no usages
    public static void main(String[] args) {
        ReleveService stub = new ReleveWS().getReleveServicePort();
        System.out.println("rib : " + stub.getReleve(rib: 1).getRib());
        System.out.println("date releve : " + stub.getReleve(rib: 1).getDateReleve());
        System.out.println("solde : " + stub.getReleve(rib: 1).getSolde());
        System.out.println("");
    }
}
```

- Resultat de la requête



```
ClientWS x
"C:\Program Files\Java\jdk-19\bin\java.exe" ...
rib : 1
date releve : 2023-01-15T11:48:19.693+01:00
solde : 35927.474873964806

Process finished with exit code 0
```