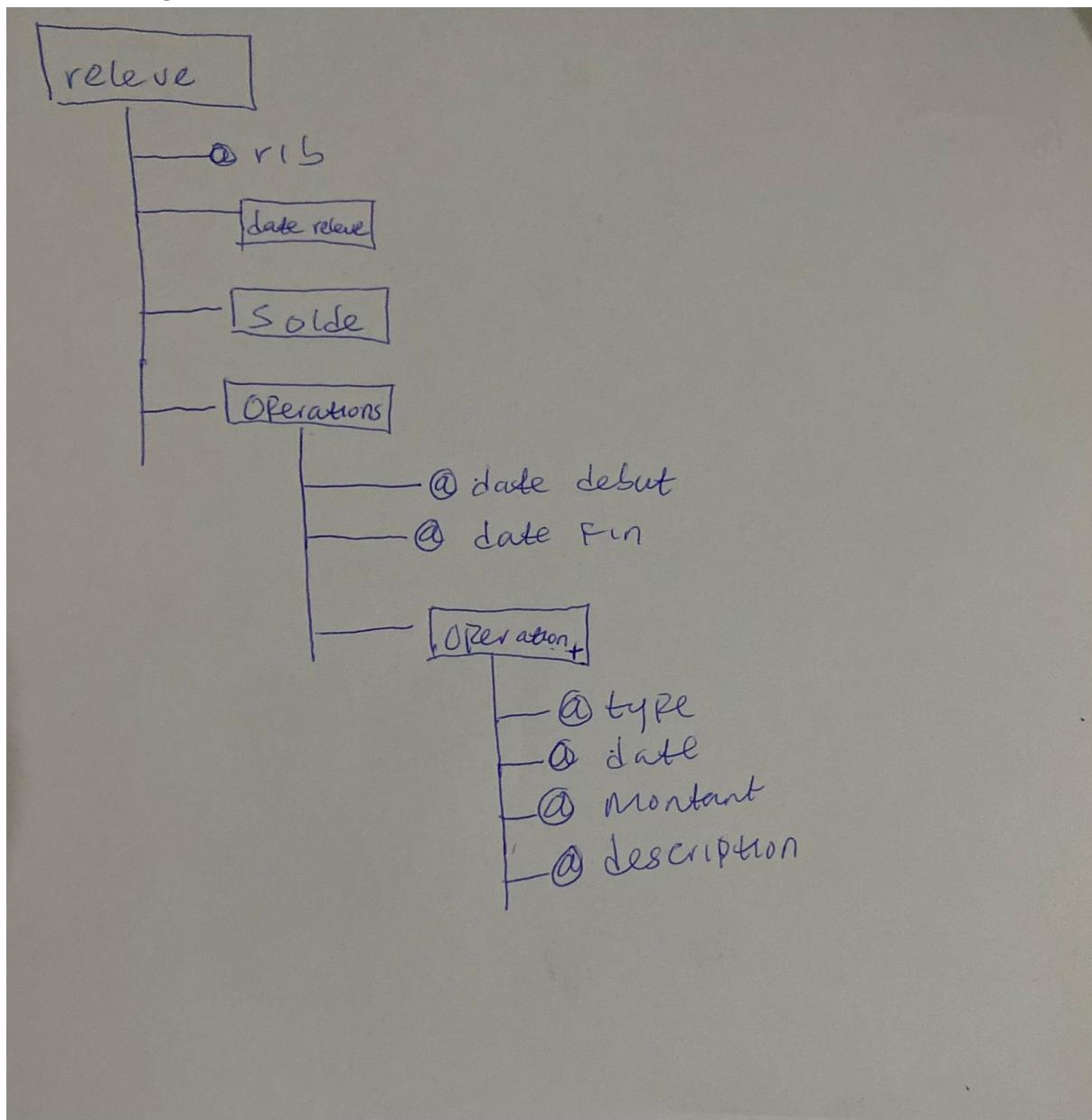


Technologie XML et Web Services

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  <!ATTLIST releve
9      |       rib NMTOKEN #REQUIRED>
0
1  <!ATTLIST operations
2      |       dateDebut CDATA #REQUIRED
3      |       dateFin CDATA #REQUIRED>
4
5  <!ATTLIST operation
6      |       type CDATA #REQUIRED
7      |       date CDATA #REQUIRED
8      |       montant NMTOKEN #REQUIRED
9      |       description CDATA #REQUIRED>
0

```

You, seconds ago • Uncommitted changes

Document XML validé par la DTD

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE releve SYSTEM "file:/C:/Users/philip/Desktop/Examen%20Technologie%20XML%20-%20Web%20Services/Technologie%20XML/codes/releve.dtd">
| <releve rib="#01112222333444555666">
|   <dateReleve>2021-11-10</dateReleve>
|   <solde>14500</solde>
|   <operations dateDebut="2021-01-01" dateFin="2021-01-30">
|     <operation type="CREDIT" date="2021-01-01" montant="9000" description="Vers Espèce"></operation>
|   </operations>
| </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>
0          <xsd:attribute name="rib" type="xsd:string" use="required"></xsd:attribute>
1      </xsd:complexType>
2  </xsd:element>
3
4  <xsd:complexType name="OPS">
5      <xsd:sequence>
6          <xsd:element name="operation" type="OP" maxOccurs="unbounded"></xsd:element>
7      </xsd:sequence>
8      <xsd:attribute name="dateDebut" type="xsd:date" use="required"></xsd:attribute>
9      <xsd:attribute name="dateFin" type="xsd:date" use="required"></xsd:attribute>
0  </xsd:complexType>
1

```

Document XML validé par la XSD

```

<?xml version="1.0" encoding="UTF-8"?>
<releve xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="file:/C:/Users/philip/Desktop/Examen%20Technologie%20XML%20-%20Web%20Services/Technologie%20XML/codes/releve.xsd">
| <rib>#01112222333444555666</rib>
| <dateReleve>2021-11-10</dateReleve>
| <solde>14500</solde>
| <operations dateDebut="2021-01-01" dateFin="2021-01-30">
|   <operation type="credit" date="2021-01-01" montant="9000" description="vers espèce"/>
|   <operation type="credit" date="2021-02-02" montant="5000" description="vers espèce"/>
|   <operation type="debit" date="2021-03-02" montant="7000" description="chèque guichet"/>
| </operations>
| </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.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0">
  <xsl:template match="/">
    <html>
      <head>
        <title>Relevé</title>
      </head>
      <body>
        <h1>Relevé</h1>
        <xsl:for-each select="releve">
          <p>Date relevé : <xsl:value-of select="dateReleve"/></xsl:value-of> </p>
          <p>RIB : <xsl:value-of select="@rib"/></xsl:value-of> </p>
          <p>Solde <xsl:value-of select="@solde"/></xsl:value-of> </p>
          <table border="2">
            <caption>Somme des Opérations</caption>
            <thead>
              <th>Type d'opération</th>
              <th>Date</th>
              <th>Montant</th>
              <th>Description</th>
            </thead>
            <tbody>
              <xsl:for-each select="operations">
                <xsl:for-each select="operation">
                  <tr>
                    <td> <xsl:value-of select="@type"/></xsl:value-of> </td>
                    <td> <xsl:value-of select="@date"/></xsl:value-of> </td>
                    <td> <xsl:value-of select="@montant"/></xsl:value-of> </td>
                    <td> <xsl:value-of select="@description"/></xsl:value-of> </td>
                  </tr>
                </xsl:for-each>
                <tr>
                  <td colspan="2">Total des operations de crédit</td>
                  <td colspan="2" align="center"> <xsl:value-of select="sum(operation[@type='credit']/@montant)" /></xsl:value-of> </td>
                </tr>
                <tr>
                  <td colspan="2">Total des operations de débit</td>
                  <td colspan="2" align="center"> <xsl:value-of select="sum(operation[@type='debit']/@montant)" /></xsl:value-of> </td>
                </tr>
              </xsl:for-each>
            </tbody>
          </table>
        </xsl:for-each>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

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.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3      xmlns:xs="http://www.w3.org/2001/XMLSchema"
4      exclude-result-prefixes="xs"
5      version="2.0">
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>
38                                 </tr>
39                             </xsl:for-each>
40                         <xsl:for-each select="operations">
41                             <xsl:for-each select="operation[@type='credit']">
42                                 <tr>
43                                     <td> <xsl:value-of select="@type"/></xsl:value-of> </td>
44                                     <td> <xsl:value-of select="@date"/></xsl:value-of> </td>
45                                     <td> <xsl:value-of select="@montant"/></xsl:value-of> </td>
46                                     <td> <xsl:value-of select="@description"/></xsl:value-of> </td>
47                                 </tr>
48                             </xsl:for-each>
49                         </tbody>
50                     </table>
51                 </xsl:for-each>
52             </body>
53         </html>
54     </xsl:template>
55 </xsl:stylesheet>

```

- **Réultat 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

```
1 package philip.shedrack;
2
3 import jakarta.xml.bind.annotation.XmlAccessType;
4 import jakarta.xml.bind.annotation.XmlAccessorType;
5 import jakarta.xml.bind.annotation.XmlAttribute;
6 import jakarta.xml.bind.annotation.XmlRootElement;
7
8 import java.util.Date;
9
10 @XmlRootElement
11 @XmlAccessorType(XmlAccessType.FIELD)
12 public class Operation {
13     @XmlAttribute
14     private TypeOperation type;
15     private Date date;
16     private double montant;
17     private TypeDescription description;
18
19     public Operation() {
20     }
21
22     public Operation(TypeOperation type, Date date, double montant, TypeDescription description) {
23         this.type = type;
24         this.date = date;
25         this.montant = montant;
26         this.description = description;
27     }
28
29
30     public TypeOperation getType() {
31         return type;
32     }
33
34     public void setType(TypeOperation type) {
35         this.type = type;
36     }
37
38     public Date getDate() {
39         return date;
40     }
41
42     public void setDate(Date date) {
43         this.date = date;
44     }
45
46     public double getMontant() {
47         return montant;
48     }
49
50     public void setMontant(double montant) {
51         this.montant = montant;
52     }
53
54     public TypeDescription getDescription() {
55         return description;
56     }
57
58     public void setDescription(TypeDescription description) {
59         this.description = description;
60     }
61
62     @Override
63     public String toString() {
64         return "Operation{" +
65                 "type=" + type +
66                 ", date=" + date +
67                 ", montant=" + montant +
68                 ", description=" + description +
69                 '}';
70     }
71 }
72 }
```

2) Classe Relevé

```
Mapping objet XML > oxm > src > main > java > philip > shedrack > Relevé.java > {} philip.shedrack
1 package philip.shedrack;
2
3 import jakarta.xml.bind.annotation.XmlElement;
4 import jakarta.xml.bind.annotation.XmlRootElement;
5
6 import java.util.ArrayList;
7 import java.util.List;
8
9 @XmlRootElement
10 public class Relevé {
11     @XmlElement(name = "opération")
12     public List<Opération> opérations = new ArrayList<>();
13 }
14
```

3) Classe Serialisation

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

```
Mapping objet XML > oxm > src > main > java > philip > shedrack > SerialisationXML.java > ...
1 package philip.shedrack;
2
3 import jakarta.xml.bind.JAXBContext;
4 import jakarta.xml.bind.Marshaller;
5
6 import java.io.File;
7 import java.util.Date;
8
9 public class SerialisationXML {
10     Run | Debug
11     public static void main(String[] args) throws Exception {
12         Relevé relevé = new Relevé();
13         relevé.opérations.add(new Opération(TypeOperation.CREDIT, new Date(), 1000, TypeDescription.Vers_Espèce));
14         relevé.opérations.add(new Opération(TypeOperation.DEBIT, new Date(), 2500, TypeDescription.Chèque_Guichet));
15         relevé.opérations.add(new Opération(TypeOperation.DEBIT, new Date(), 500, TypeDescription.Prélèvement_Assurance));
16         relevé.opérations.add(new Opération(TypeOperation.CREDIT, new Date(), 5800, TypeDescription.Virement));
17
18         JAXBContext jaxbContext = JAXBContext.newInstance(Relevé.class);
19         Marshaller marshaller = jaxbContext.createMarshaller();
20         marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
21         marshaller.marshal(relevé, new File("relevé.xml"));
22     }
23 }
```

- Fichier XML contenant les données de la sérialisation

```
Mapping objet XML > oxm > reeve.xml
You, seconds ago | 2 authors (Baba Prince and others)
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <reeve>
3      <operation type="CREDIT">
4          <date>2022-12-28T20:10:37.299+01:00</date>
5          <montant>1000.0</montant>
6          <description>Vers_Espèce</description>
7      </operation>
8      <operation type="DEBIT">
9          <date>2022-12-28T20:10:37.300+01:00</date>
10         <montant>2500.0</montant>
11         <description>Chèque_Guichet</description>
12     </operation>
13     <operation type="DEBIT">
14         <date>2022-12-28T20:10:37.300+01:00</date>
15         <montant>500.0</montant>
16         <description>Prélèvement_Assurance</description>
17     </operation>
18     <operation type="CREDIT">
19         <date>2022-12-28T20:10:37.300+01:00</date>
20         <montant>5800.0</montant>
21         <description>Virement</description>
22     </operation>
23 </reeve>
```

4) Classe Deserialisation

```
Mapping objet XML > oxm > src > main > java > philip > shedrack > DeserialisationXML.java > ...
1 package philip.shedrack;
2
3 import jakarta.xml.bind.JAXBContext;
4 import jakarta.xml.bind.Unmarshaller;
5
6 import java.io.File;
7
8 public class DeserialisationXML {
9     Run | Debug
10    public static void main(String[] args) throws Exception {
11        JAXBContext jaxbContext = JAXBContext.newInstance(Reeve.class);
12        Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();
13        Reeve reeve = (Reeve) unmarshaller.unmarshal(new File("reeve.xml"));
14        System.out.println("*****");
15        for (Operation op : reeve.operations) {
16            System.out.println(op.toString());
17            System.out.println("-----");
18        }
19    }
20 }
```

5) Classe GenerateXMLShcema

Cette classe permet de générer le schema XML représentant la structure d'un relevé.

```
Mapping objet XML > oxm > src > main > java > philip > shedrack > ❸ GenerateXMLSchema.java > ...
1 package philip.shedrack;
2
3 import jakarta.xml.bind.JAXBContext;
4 import jakarta.xml.bind.SchemaOutputResolver;
5
6 import javax.xml.transform.Result;
7 import javax.xml.transform.stream.StreamResult;
8 import java.io.File;
9 import java.io.IOException;
10
11 public class GenerateXMLSchema {
12     Run | Debug
13     public static void main(String[] args) throws Exception {
14         JAXBContext jaxbContext = JAXBContext.newInstance(Releve.class);
15         jaxbContext.generateSchema(new SchemaOutputResolver() {
16             @Override
17             public Result createOutput(String s, String s1) throws IOException {
18                 File file = new File("releve.xsd");
19                 StreamResult streamResult = new StreamResult(file);
20                 streamResult.setSystemId(file.getName());
21                 return streamResult;
22             }
23         });
24     }
25 }
26 |
```

- Schéma XML représentant la structure d'un relevé

```
Mapping objet XML > oxm > ❹ releve.xsd
You, seconds ago | 2 authors (Baba Prince and others)
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <xss: schema version="1.0" xmlns:xss="http://www.w3.org/2001/XMLSchema">
3
4     <xss:element name="operation" type="operation"/>
5
6     <xss:element name="releve" type="releve"/>
7
8     <xss:complexType name="releve">
9         <xss:sequence>
10            <xss:element ref="operation" minOccurs="0" maxOccurs="unbounded"/>
11        </xss:sequence>
12    </xss:complexType>
13
14    <xss:complexType name="operation">
15        <xss:sequence>
16            <xss:element name="date" type="xs:dateTime" minOccurs="0"/>
17            <xss:element name="montant" type="xs:double"/>
18            <xss:element name="description" type="typeDescription" minOccurs="0"/>
19        </xss:sequence>
20        <xss:attribute name="type" type="typeOperation"/>
21    </xss:complexType>
22
23    <xss:simpleType name="typeOperation">
24        <xss:restriction base="xs:string">
25            <xss:enumeration value="CREDIT"/>
26            <xss:enumeration value="DEBIT"/>
27        </xss:restriction>
28    </xss:simpleType>
29
30    <xss:simpleType name="typeDescription">
31        <xss:restriction base="xs:string">
32            <xss:enumeration value="Vers_Esp&#232;ce"/>
33            <xss:enumeration value="Ch&#232;que_Guichet"/>
34            <xss:enumeration value="Pr&#233;l&#232;vement_Assurance"/>
35            <xss:enumeration value="Virement"/>
36        </xss:restriction>
37    </xss:simpleType>
38 </xss: schema>
39 |
```

C) Partie C Web Services

1) Web Services

• Classe Operation

```
Web Services > ws > src > main > java > philip > shedrack > Operation.java > Operation > montant
1  package philip.shedrack;
2
3  import jakarta.xml.bind.annotation.XmlAccessType;
4  import jakarta.xml.bind.annotation.XmlAccessorType;
5  import jakarta.xml.bind.annotation.XmlAttribute;
6  import jakarta.xml.bind.annotation.XmlRootElement;
7
8  import java.util.Date;
9
10 public class Operation {
11     private TypeOperation type;
12     private Date date;
13     private double montant;
14     private TypeDescription description;
15
16     public Operation() {
17     }
18
19     public Operation(TypeOperation type, Date date, double montant, TypeDescription description) {
20         this.type = type;
21         this.date = date;
22         this.montant = montant;
23         this.description = description;
24     }
25
26
27     public TypeOperation getType() {
28         return type;
29     }
30
31     public void setType(TypeOperation type) {
32         this.type = type;
33     }
34
35     public Date getDate() {
36         return date;
37     }
38
39     public void setDate(Date date) {
40         this.date = date;
41     }
42
43 }
```

```
Web Services > ws > src > main > java > philip > shedrack > Operation.java > Operation > montant
34
35     public Date getDate() {
36         return date;
37     }
38
39     public void setDate(Date date) {
40         this.date = date;
41     }
42
43     public double getMontant() {
44         return montant;
45     }
46
47     public void setMontant(double montant) {
48         this.montant = montant;
49     }
50
51     public TypeDescription getDescription() {
52         return description;
53     }
54
55     public void setDescription(TypeDescription description) {
56         this.description = description;
57     }
58
59     @Override
60     public String toString() {
61         return "Operation{" +
62             "type=" + type +
63             ", date=" + date +
64             ", montant=" + montant +
65             ", description=" + description +
66             '}';
67     }
68 }
```

Classe Releve

```
Web Services > ws > src > main > java > philip > shedrack > Releve.java > {} philip.shedrack
 1 package philip.shedrack;
 2
 3 import jakarta.jws.WebService;
 4 import jakarta.xml.bind.annotation.XmlElement;
 5 import jakarta.xml.bind.annotation.XmlRootElement;
 6
 7 import java.util.ArrayList;
 8 import java.util.Date;
 9 import java.util.List;
10
11 public class Releve {
12     private long rib;
13     private List<Operation> operations = new ArrayList<>();
14     private Date dateReleve;
15     private double solde;
16
17     public Releve(long rib, Date dateReleve, double solde) {
18         this.rib = rib;
19         operations.add(new Operation(TypeOperation.CREDIT, new Date(), Math.random()*9800, TypeDescription.Vers_Espèce));
20         operations.add(new Operation(TypeOperation.DEBIT, new Date(), Math.random()*5600, TypeDescription.Virement));
21         operations.add(new Operation(TypeOperation.CREDIT, new Date(), Math.random()*3500, TypeDescription.Chèque_Guichet));
22         operations.add(new Operation(TypeOperation.DEBIT, new Date(), Math.random()*100, TypeDescription.Prélèvement_Assurance));
23         this.dateReleve = dateReleve;
24         this.solde = solde;
25     }
26
27     public long getRib() {
28         return rib;
29     }
30
31     public void setRib(long rib) {
32         this.rib = rib;
33     }
34
35     public List<Operation> getOperations() {
36         return operations;
37     }
38
39     public void setOperations(List<Operation> operations) {
40         this.operations = operations;
41     }
42
43     public Date getDateReleve() {
44         return dateReleve;
45     }
46
47     public void setDateReleve(Date dateReleve) {
48         this.dateReleve = dateReleve;
49     }
50
51     public double getSolde() {
52         return solde;
53     }
54
55     public void setSolde(double solde) {
56         this.solde = solde;
57     }
58 }
59 }
```

• Classe RelevéService

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

```
Web Services > ws > src > main > java > philip > shedrack > RelevéService.java > ...
1 package philip.shedrack;
2
3 import jakarta.jws.WebMethod;
4 import jakarta.jws.WebParam;
5 import jakarta.jws.WebService;
6
7 import java.util.Date;
8 import java.util.List;
9
10 @WebService(serviceName = "RelevéWS")
11 public class RelevéService {
12     @WebMethod
13     public Relevé getRelevé(@WebParam(name = "rib") long rib) {
14         return new Relevé(rib, new Date(), Math.random()*43000);
15     }
16
17     @WebMethod
18     public List<Relevé> listRelevé() {
19         return List.of(
20             new Relevé(1, new Date(), Math.random()*2000),
21             new Relevé(2, new Date(), Math.random()*87000),
22             new Relevé(3, new Date(), Math.random()*2400),
23             new Relevé(4, new Date(), Math.random()*9400),
24             new Relevé(5, new Date(), Math.random()*8600),
25             new Relevé(6, new Date(), Math.random()*4800)
26         );
27     }
28 }
29
30 }
```

2) Serveur JaxWS

```
Web Services > ws > src > main > java > server > ServerJWS.java > ...
You, seconds ago | 2 authors (Baba Prince and others)
1 package server;
2
3 import jakarta.xml.ws.Endpoint;
4 import philip.shedrack.RelevéService;
5
Baba Prince, 12 hours ago | 1 author (Baba Prince)
6 public class ServerJWS {
    Run | Debug
7     public static void main(String[] args) {
8         Endpoint.publish("http://0.0.0.0:9191/", new RelevéService());
9         System.out.println("Web Service déployé sur http://0.0.0.0:9191/");
10    }
11 }
12 |
```

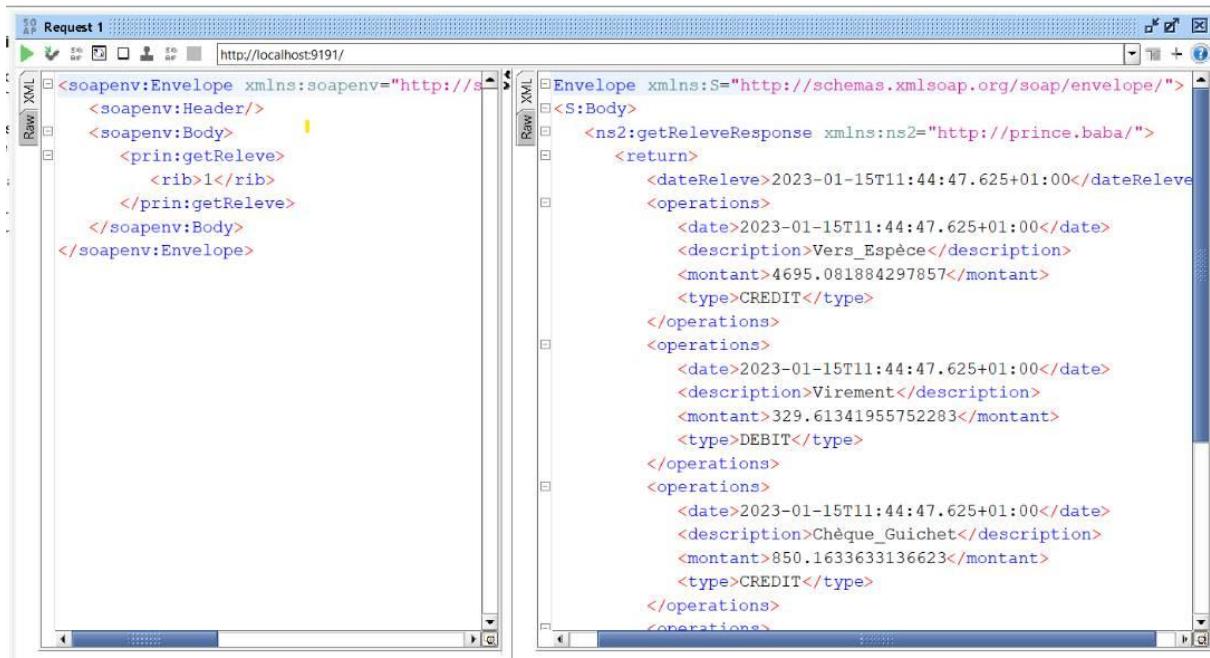
3) WSDL

```
Web Services > Relevé-soapui-project.xml
  You, seconds ago | 2 authors (Baba Prince and others)
1  <?xml version="1.0" encoding="UTF-8"?>
2  <con:soapui-project id="b0ace8b2-82ef-4a48-ba54-d7c69d0d099c" activeEnvironment="Default" name="Relevé" resourceRoot="" soapui-version="5.7.0"
3  <!--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#129f78
4  <definitions targetNamespace="http://philip.shedrack/" name="releve" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssec" wsu:Id="urn:uuid:1337423e-423e-423e-423e-423e423e423e">
5    <types>
6      <xsd:schema>
7        | <xsd:import namespace="http://philip.shedrack/" schemaLocation="http://localhost:9191/?xsd=1" />
8      </xsd:schema>
9    </types>
10   <message name="listReleve">
11     <part name="parameters" element="tns:listReleve"/>
12   </message>
13   <message name="listReleveResponse">
14     <part name="parameters" element="tns:listReleveResponse"/>
15   </message>
16   <message name="getReleve">
17     <part name="parameters" element="tns:getReleve"/>
18   </message>
19   <message name="getReleveResponse">
20     <part name="parameters" element="tns:getReleveResponse"/>
21   </message>
22   <portType name="ReleveService">
23     <operation name="listReleve">
24       <input wsam:Action="http://philip.shedrack/ReleveService/listReleveRequest" message="tns:listReleve"/>
25       <output wsam:Action="http://philip.shedrack/ReleveService/listReleveResponse" message="tns:listReleveResponse"/>
26     </operation>
27     <operation name="getReleve">
28       <input wsam:Action="http://philip.shedrack/ReleveService/getReleveRequest" message="tns:getReleve"/>
29       <output wsam:Action="http://philip.shedrack/ReleveService/getReleveResponse" message="tns:getReleveResponse"/>
30     </operation>
31   </portType>
32   <binding name="ReleveServicePortBinding" type="tns:ReleveService">
33     <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document">
34       <operation name="listReleve">
35         <soap:operation soapAction="" />
36         <input>
37           <soap:body use="literal"/>
38         </input>
39         <output>
40           <soap:body use="literal"/>

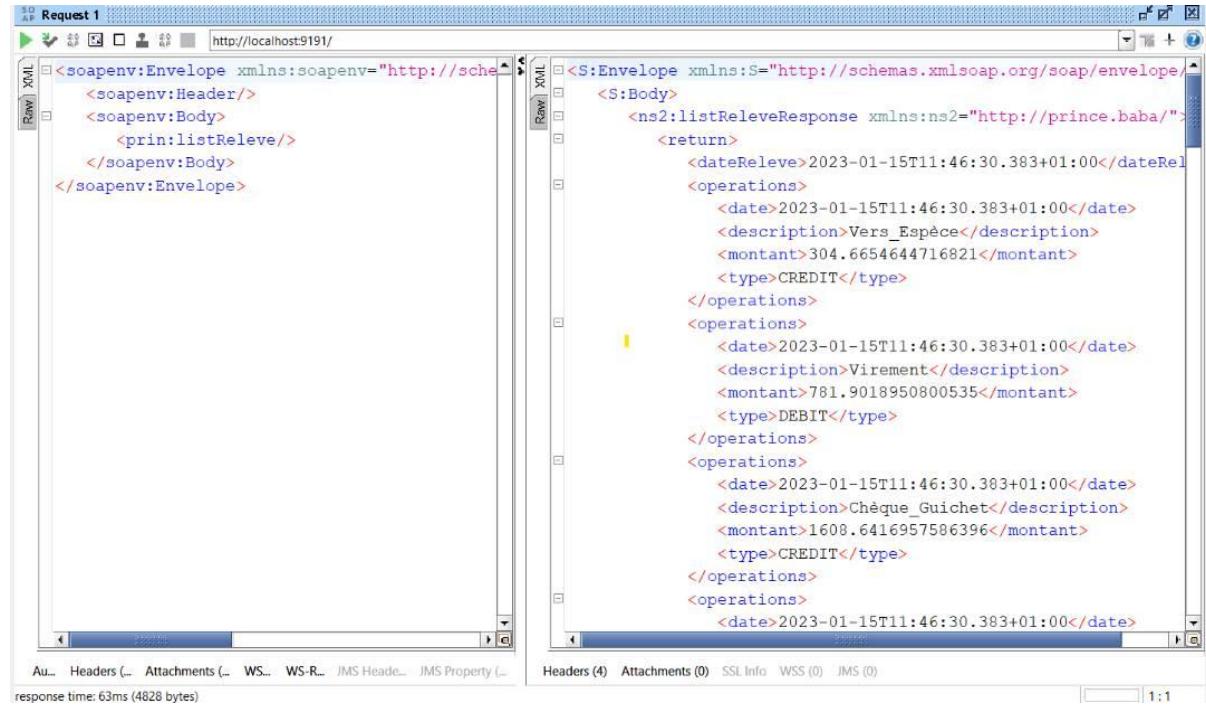
```

4) Test des méthodes avec SoapUI

- Method getReleve



• Method listReleve



5) Client SOAP

Java • Classe ClientWS

```
Web Services > client_ws > src > main > java > ClientWS.java > ...
Baba Prince, 12 hours ago | 1 author (Baba Prince)
1 import proxy.Operation;
2 import proxy.Releve;
3 import proxy.ReleveService;
4 import proxy.ReleveWS;
5
Baba Prince, 12 hours ago | 1 author (Baba Prince)
6 public class ClientWS {
    Run | Debug
7     public static void main(String[] args) {
8         ReleveService stub = new ReleveWS().getReleveServicePort();
9         System.out.println("rib : " + stub.getReleve(1).getRib());
10        System.out.println("date releve : " + stub.getReleve(1).getDateReleve());
11        System.out.println("solde : " + stub.getReleve(1).getSolde());
12        System.out.println("");
13    }
14 }
15
```

- **Réultat de la requête**

```
ClientWS ✘
"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
```