CORMIER Esteban

LAUMONNIER Mathieu

Compte rendu projet JAVA

Sérialisation :

Nous avons choisi d'utiliser XML et le parseur DOM pour pouvoir extraire et manipuler les nœuds xml plus facilement. Après la lecture de la documentation en ligne il a fallu commencer par créer un fichier xml avec une architecture simple pour pouvoir accéder aux éléments.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<root>
  <element>
    <id>0</id>
    <nom>Cormier</nom>
    <numTel>645954638</numTel>
  </element>
</root>
```

Ici le nœud racine est « root » et ensuite on décompose le fichier en « element », ici les éléments sont des clients qui ont des champs comme un id, un nom et un numéro de téléphone.

Il a donc ensuite fallu trouver une façon de lire les champs un à un pour, à la fin, recomposer un client avec ses informations.

```java
public static ArrayList<Client> readXMLClient(){
  ArrayList<Client>  clients=new ArrayList<Client>();
  try{
    DocumentBuilderFactory docbf = DocumentBuilderFactory.newInstance();
    DocumentBuilder docb = docbf.newDocumentBuilder();
    Document doc = docb.parse(new File("dataClient.xml"));
    NodeList listeC = doc.getElementsByTagName("element"); //On récupère chaque noeud
    for (int i=0;i<listeC.getLength() ;i++ ) {
      if(listeC.item(i).getNodeType()==Node.ELEMENT_NODE){ //On récupère les noeuds qui sont des éléments
        Element node = (Element) listeC.item(i);
        clients.add(new Client(
        node.getElementsByTagName("nom").item(0).getTextContent(),
        Integer.parseInt(node.getElementsByTagName("numTel").item(0).getTextContent()),
        Integer.parseInt(node.getElementsByTagName("id").item(0).getTextContent())));
        System.out.println(clients.get(i));
      }
    }
  }catch(ParserConfigurationException e){
    e.printStackTrace();
  }catch(SAXException e){
    e.printStackTrace();
  }catch(IOException e){
    e.printStackTrace();
  }
  return clients;
}
```

La fonction ci-dessus va avoir pour rôle de lire le fichier dataClient.xml pour ensuite remplir une ArrayList de client. On va tout d'abord créer un document sur la base du fichier xml pour ensuite récupérer la liste des nœuds « element » et les parcourir 1 à 1. La création des objets Client sera faite

grâce aux champs textuels contenus dans les sous-champs qui seront intégrés dans l'ArrayList qui sera finalement renvoyée à la fin de la fonction.

```
public static void addClient(Client c){
  int last=getLastIDClient();
  try{
    DocumentBuilderFactory docbf = DocumentBuilderFactory.newInstance();
    DocumentBuilder docb = docbf.newDocumentBuilder();
    Document doc = docb.parse(new File("dataClient.xml"));
    Node racine = doc.getFirstChild(); //On récupère la racine
    Element client = doc.createElement("element"); // On crée un nouveau client
    Element nom = doc.createElement("nom"); //On crée une spécification du nouveau client
    nom.appendChild(doc.createTextNode(c.getNom())); // On ajoute a la spec sa valeur
    client.appendChild(nom); //On ajoute chaque champ au client
    Element numTel = doc.createElement("numTel");
    numTel.appendChild(doc.createTextNode(Integer.toString(c.getNumTel())));
    client.appendChild(numTel);
    Element id = doc.createElement("id");
    id.appendChild(doc.createTextNode(Integer.toString(last)));
    client.appendChild(id);
    racine.appendChild(client); // On ajoute le client a la racine
    DOMSource robot = new DOMSource(doc); // On donne le document xml comme source pour
    Transformer optimus = TransformerFactory.newInstance().newTransformer(); // On crée
    optimus.setOutputProperty(OutputKeys.INDENT,"yes"); //On spécifie que la modification
    StreamResult gentil = new StreamResult("dataClient.xml"); // On donne le fichier qui
    optimus.transform(robot, gentil);// On transforme la source avec les informations do
```

Pour ce qui est de l'écriture de dans le fichier on va encore construire un document qui a pour base le dataClient.xml. Puis on va créer les nœud pour intégrer au document les éléments. On commence par le nœud père « element » puis on ajoute à ce nœud des éléments qu'on définira comme fils. Dès que les nœuds auront tous été ajoutés au père on ajoute ce père au nœud racine. Il ne reste plus qu'a transformer notre document DOM en document xml.

Le même schéma sera reproduit pour les véhicules et les locations.

Les fonctions supplémentaires comme getLastIdClient sont utilisées pour récupérer le dernier id d'un client et donc calculer le prochain id du prochain nouveau client qui sera ajouté.

Interface :

Pour le menu principal nous utilisons 4 boutons pour accéder aux différents menus.

Menu Client : Nous utilisons une JList pour afficher les clients et 2 JTextFields pour afficher leur nom et numéro de téléphone. Il y a un bouton « Ajouter » qui ouvre une fenêtre pour créer un nouveau client en remplissant les JTextFields avec les informations du client. Il y a un Bouton « Supprimer » pour supprimer le client sélectionné dans la JList. Il y a un bouton « Ok » qui modifie le client sélectionné si un des JTextField a été modifié. Un bouton « Actualiser » afin de rafraichi la page afin d'observer les changements. Et le bouton retour qui permet de revenir au menu principal.

Menu Locations : Nous utilisons une JList pour chaque type de véhicule que nous affichons grâce à un CardLayout et à des boutons. Il y a des JTextFields pour afficher les informations du véhicule sélectionné et une JComboBox pour choisir le client. Le bouton « Ok » créer la location avec le client et le véhicule sélectionné.

Menu Restitutions : Nous utilisons une JList pour afficher toutes les locations, un JTextField pour y entrer le nombre de kilomètres parcourus et un JTextField pour afficher le prix à payer. Le bouton « Ok » calcul le prix et supprime la location choisie.

Menu Gestion : Nous utilisons une JList pour afficher tous les véhicules et des JTextFields pour afficher leurs informations. Le bouton « Ajouter » ouvre une fenêtre avec une JComboBox pour choisir quel type de véhicule on ajoute et des JTextFields pour les informations. Il y a aussi un bouton « Actualiser » et « Supprimer ».