

Cours n⁸ Gestion des documents



Sommaire

1. Gestion des documents textuels

- 1. Interfaces de modélisation
- 2. Visualisation et Edition

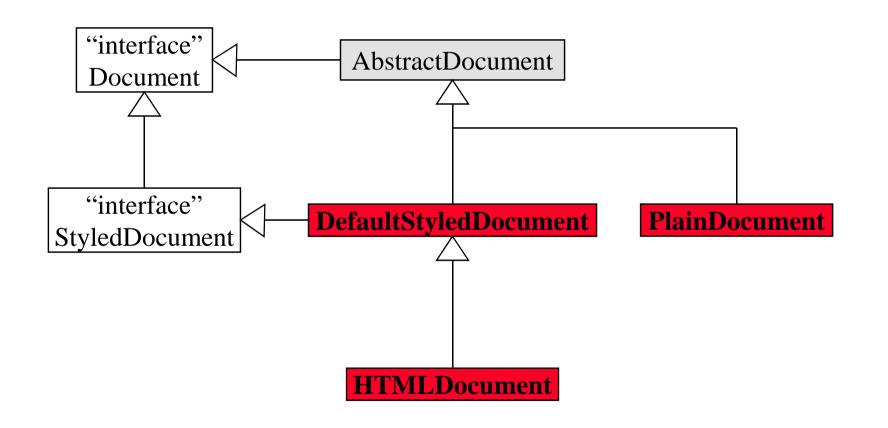
2. Analyse de documents XML

- 1. Simple API for XML (SAX)
- 2. Document Object Mode (DOM)

1. GESTION DES DOCUMENTS TEXTUELS

Modélisation

Trois classes concrètes



1.1 INTERFACES DE MODELISATION

Document

Spécification du contrat que toutes les types de document doivent implémenter

```
Texte du document
```

String **getText** (int début, int taille) void **getText** (int début, int taille, Segment cc)

Nombre de caractères du document int getLength ()

Ajout d'un texte cc à l'indice i du document void insertString(int i, String cc, AttributeSet a)

Suppression d'une partie du texte à partir de l'indice i du document void remove(int i, int taille)

Positionnement d'une marque au ième caractère Position createPosition (int i)

Ajout d'un écouteur d'événement modification de document void addDocumentListener (DocumentListener dl)

Interface Position

Indice correspondant à la position d'une marque

int getOffset()

StyledDocument - Typographie

Création, Gestion et Utilisation de styles de typographie

Création des styles

Applications des méthodes de la classe StyleConstants

```
static void setBackground(Style s, Color c)
                                                      // couleur de fond
static void setForeground(Style s, Color c)
                                                       // couleur du caractère
static void setFontFamily(Style s, String cc)
                                                      // police
static void setFontSize(Style's, int corps)
                                                       // corps
static void setBold(Style s, boolean b)
                                                      // gras
static void setItalic(Style s, boolean b)
                                                      // italique
static void setSubscript(Style s, float x)
                                                      // indice
static void setSuperscript(Style's, float'x)
                                                      // exposant
static void setAlignment(Style s, int alig)
                                                      // alignement (gauche, droit, ..)
static void setFirstLineIndent(Style s, float x)
                                                       // Indentation
static void setLeftIndent(Style`s, float x)
static void setRightIndent(Style's, float'x)
static void setSpaceAbove(Style s, float x)
static void setSpaceBelow(Style s, float x)
static void setLineSpacing(Style s, float x)
                                                       // taille des lignes
```

1.1 INTERFACES DE MODELISATION

StyledDocument - Styles

Gestion des styles

Ajout d'un style s dans la hiérarchie h des styles précédemment définis Style addStyle(Style s, Style h)

Recherche du style s Style getStyle(String cc)

Suppression du style s void removeStyle(String cc)

Recherche du style du paragraphe contenant le caractère d'indice i Style getLogicalStyle(int i)

Utilisation des styles

Application ou ajout du style s à une partie du texte void setCharacterAttributes(int i, int taille, Style s, boolean app)
Application du style s au paragraphe contenant le caractère d'indice i void setLogicalStyle(int pos, Style s)

Introduction

Cinq classes concrètes "interface" Accessible JTextComponent "interface" Scrollable **JTextField JEditorPane JTextArea JPasswordField JTextPane**

Classe JTextComponent

Classe abstraite, ancêtre de tous les classes de visualisation de texte

Définition de plus de 100 méthodes

```
Lien avec un document
```

void setDocument (Document doc)

Gestion du sélecteur (caret)

void setCaretPosition(int i) int getCaretPosition()

Sélection d'une zone de texte

void select(int début, int fin) void selectAll()

Lecture et écriture d'un fichier texte

Méthodes interactives d'édition (possibilité de redéfinition)

void copy() void cut() void paste()

Ecouteurs

void addCaretListener(CaretListener listener)

Classe JTextArea

Composant de visualisation de documents textuels simples

Compatibilité avec les documents de type PlainDocument, Vue d'une partie du document (implémentation de l'interface Scrollable), Contrôle des coupures de fin de ligne

Constructeurs

Constructeur vide

JTextArea()

Construction à partir du document doc avec l lignes, c colonnes, et le titre t

JTextArea(Document doc, String t, int I, int c)

JTextArea(Document doc)

JTextArea(String t, int I, int c)

JTextArea(String t)

JTextArea(int I, int c)

Gestion des fin de ligne et des tabulations

void setLineWrap(boolean wrap)

void setWrapStyleWord(boolean word)

void setTabSize(int taille)

JTextArea - Programme

```
PlainDocument pdoc = new PlainDocument();
String NomFichier = "livres/avare.txt";
pdoc.putProperty("Titre" , NomFichier);
                                                                            ≗ L'avare
JTextArea ta = new JTextArea(pdoc);
                                                                            VALERE
                                                                            Hé quoi ? charmante Elise, vous devenez mélancolique, après
                                                                            les obligeantes assurances que vous avez eu la bonté de me donne
try {
                                                                            votre foi ? Je vous vois soupirer, hélas ! au milieu de ma joie. Est-ce :
                                                                           du regret, dites-moi, de m'avoir fait heureux, et vous repentez-vous de
ta.read(new FileReader(NomFichier), null);
                                                                            cet engagement où mes feux ont pu vous contraindre ?
                                                                           ELISE
catch (IOException e) {}
                                                                           Non, Valère, je ne puis pas me repentir de tout ce que je fais
                                                                           pour vous. Je mV sens entraîner par une trop douce puissance, et je
                                                                           n'ai pas même la force de souhaiter que les choses ne fussent pas.
                                                                           Mais, a vous dire vrai, le succès me donne de l'inquiétude, et je
JFrame fen = new JFrame("L'avare");
                                                                            crains fort de vous aimer un peu plus que je ne devrais.
fen.setSize(400, 500);
                                                                            VALERE
Container jp = fen.getContentPane();
                                                                            Hé ! que pouvez-vous craindre, Elise, dans les bontés que
                                                                            vous avez pour moi?
ip.add(ta);
jp.add(new JScrollPane(ta));
                                                                            ELISE
                                                                            Hélas! cent choses à la fois : l'emportement d'un père, les
fen.setVisible(true);
                                                                            reproches d'une famille, les censures du monde ; mais plus que tou
                                                                            Valère, le changement de votre coeur, et cette froideur criminelle don
                                                                            ceux de votre sexe payent le plus souvent les témoignages trop arde
                                                                            d'une innocente amour.
                                                                            VALERE
                                                                            Ah! ne me faites pas ce tort de juger de moi par les autres.
```

Classe JEditorPane

Composant de visualisation de documents textuels enrichis

Lecture des formats texte pur, RTF et HTML, Initialisation à partir d'une URL Gestion des hyperliens

Constructeurs

Constructeur vide

JEditorPane()

Construction à partir de la chaîne cc de format f (txt, rft, html)

JEditorPane(String f, String cc)

Construction à partir de l'URL u

JEditorPane(URL u)

Construction à partir d'une chaîne de caractère correspondant à une URL

JEditorPane(String url)

Classe JTextPane

Composant de visualisation de documents textuels enrichies

Classe dérivée de JEditorPane, Compatibilité avec les documents de type DefaultStyledDocument, Attachement d'images et de composants à des parties du texte

Constructeurs

Constructeur vide

JTextPane()

Construction à partir du document doc

JTextPane(StyledDocument doc)

Liens avec les documents

void setDocument(Document doc)

void setStyledDocument(StyledDocument doc)

Classe JTextPane – Programme (1/2)

```
String NomFichier = "livres/avare.txt";
JTextPane tp = new JTextPane();
// lecture du document
try {
tp.read(new FileReader(NomFichier), null); }
catch (IOException e) {}
String cc = null;
try {cc = dsdoc.getText(0, dsdoc.getLength()); }
catch (BadLocationException e) {}
// définition du style Personnage
StyledDocument dsdoc = tp.getStyledDocument();
dsdoc.putProperty("Titre" , "L'avare");
Style s = dsdoc.addStyle("Personnage" , null);
StyleConstants.setForeground(s, Color.PINK);
StyleConstants.setFontSize(s, 16);
StyleConstants.setBold(s, true);
```

Classe JTextPane – Programme (2/2)

```
//enrichissement du texte
BreakIterator bi1 = BreakIterator.getWordInstance();
bil.setText(cc);
int debut = bil.first();
for (int fin = bil.next(); fin != BreakIterator.DONE; debut = fin, fin
= bi1.next()) {
      (cc.substring(debut, fin).matches("[A-Z]*"))
         dsdoc.setCharacterAttributes(debut, fin-debut, s, true);
                                                                   ≗ L'avare
                                                                    A FLECHE, valet de Cléante.
//affichage
                                                                   DAME CLAUDE, servante d'Harpagon,
                                                                   BRINDAVOINE, laquais d'Harpagon.
JFrame fen = new JFrame("L'avare");
                                                                    .A MERLUCHE, laquais d'Harpagon.
                                                                    LE COMMISSAIRE et son clerc.
fen.setSize(400, 500);
                                                                   La scène est à Paris.
Container jp = fen.getContentPane();
ip.add(tp);
ip.add(new JScrollPane(tp));
fen.setVisible(true);
                                                                   Hé quoi ? charmante Elise, vous devenez mélancolique, après
                                                                   les obligeantes assurances que vous avez eu la bonté de me-
                                                                   votre foi ? Je vous vois soupirer, hélas ! au milieu de ma joie
                                                                   du regret, dites-moi, de m'avoir fait heureux, et vous repentez-vous
                                                                   cet engagement où mes feux ont pu vous contraindre?
```

JAXP (Java API for XML processing)

Librairie Java pour un traitement standardisé des données au format XML (java.sun.com/xml/jaxp)

Version 1.2.4

Disponible dans le kit de développement standard (depuis la version 1.4),

Packages javax.xml.parsers, javax.xml.transform

Indépendance aux outils d'analyse des données XML (ajouts par plug-in)

Support d'outils d'analyse : CRIMSON, SAX et DOM,

Support d'outils de transformation XSLT : XALAN.

Caractéristiques (1)

Nécessité d'une API Java de référence pour le parcours d'arborescence XML

1997 – projet YAXPAPI (Yet Another XML Parser API) en référence au programme YACC (Yet Another Compiler-Compiler) de Peter Murray-Rust

1998 – SAX Version 1.0 (Peter Murray-Rust, Tim Bray et David Megginson)

2000 – SAX Version 2.0 (www.saxproject.org)

Langages supportées

Pascal, C++ (Arabica), Perl (module), Python

Librairie Microsoft (MS XML-3.0) accessible en VB, C et C++

Caractéristiques (2)

Ensemble de points d'entrées dans la procédure de parcours d'arbre (détecteur d'événements SAX)

Début du document (startDocument)

Fin du document (endDocument)

Balise de début (startElement)

Balise de fin (endElement)

Contenu (characters)

Erreur d'analyse (error)

Pas de création d'objets pendant le parcours

Peu de mémoire utilisée
Pas de mémorisation de l'arborescence

Implémentation en Java (1)

Librairie JAXP

+

Package org.xml.java (pas de support de Sun)

Classes (2), interfaces (11) et exceptions (2) pour la création et le lancement d'un analyseur SAX.

Package org.xml.sax.helpers (pas de support de Sun)

Classes (10) pour la gestion des points d'entrées (classe DefaultHandler) de l'analyseur SAX.

Implémentation en Java (2) – Classe DefaultHandler

Traitement des événements « début du document » et « fin du document »

StartDocument() EndDocument()

Traitement de l'événement « début de balise »

StartElement(String namespaceURI, String sName, String qName, Attributes attrs)

namespaceURI : espace de noms

sName : nom de la balise (en cas d'espace de noms)

qName : nom de la balise (en cas d'absence d'espace de noms)

attrs: attribut de la balise

Implémentation en Java (3) – Classe DefaultHandler

Traitement de l'événement « fin de balise »

StartElement(String namespaceURI, String sName, String qName)

Traitement de l'événement « contenu entre deux balises »

characters(char buf[], int offset, int len)

buf : ensemble du texte analysé

offset : emplacement du 1er caractère après la balise de début

len : taille du contenu inter-balises

Construction d'un objet SAX

```
public class ParcoursArbreSax extends DefaultHandler {
public ParcoursArbreSax(InputStream isXml) {
super();
// création d'une instance d'API SAX
SAXParserFactory factory = SAXParserFactory.newInstance();
factory.setValidating(true);
//création d'un objet de parcours
SAXParser saxParser = null;
try {saxParser = factory.newSAXParser();}
catch (SAXException e) {}
catch (ParserConfigurationException e) {}
// parcours utilisant le gestionnaire d'événement de this
try {saxParser.parse(isXml, this);}
catch (IOException e) {}
catch (SAXException e) {}
```

Lecture des balises et des attributs

```
public void startElement(String namespaceURI, String sName,
  String gName, Attributes attrs) throws SAXException {
String eName = sName;
if ("".equals(eName)) eName = qName;
System.out.print("balise de début: ");
System.out.print(eName);
if (attrs != null) {
     // attributs
     for (int i = 0; i < attrs.getLength(); i++) {
           String aName = attrs.getLocalName(i);
           if ("".equals(aName)) aName = attrs.getQName(i);
     // nom
           System.out.print(" " + aName + "=\"");
     // valeur
           System.out.print(attrs.getValue(i)+"\"");
System.out.println("");
```

Lecture des données

```
public void endElement(String namespaceURI, String sName,
  String qName) throws SAXException {
String eName = sName;
if ("".equals(eName)) eName = qName;
System.out.println("balise de fin: "+eName);
public void characters(char buf[], int offset, int len)
throws SAXException
String s = new String(buf, offset, len);
String st = s.trim();
if (st.length() > 0)
     System.out.println(st);
public void startDocument() throws SAXException{
System.out.println("début du document:");}
public void endDocument() throws SAXException{
System.out.println("fin du document:"); }}
```

Extractions d'un fichier XML

```
import java.io.*;
/ * *
 * @author montagie
public class testParcoursArbreSAX {
public static void main(String[] args) {
     File fxml = new File("xml/article.xml");
     FileInputStream isxml = null;
     try {
           isxml = new FileInputStream(fxml);
     catch (IOException e) {}
ParcoursArbreSax ges = new ParcoursArbreSax(isxml);
```

Exemple de fichier XML

```
<?xml version='1.0' encoding="UTF-8"?>
<article lang="fr">
<section>
<para> <phrase>Deux nouveaux foyers de grippe aviaire ont été
détectés dans des élevages de canards de Dordogne et des
Landes, lundi 14 décembre.</phrase>
<phrase>Le ministère de l'agriculture recense à ce jour un
total de « 15 foyers d'influenza aviaire » dans cinq
départements du Sud-Ouest.</phrase> </para>
<para>
<phrase>Un cas de grippe de type H5N1 a été détecté et
confirmé lundi dans un élevage de 1 070 canards à Bosset
(Dordogne), près de Bergerac, indique le ministère sur son
site Internet.</phrase>
<phrase>C'est dans cette même commune qu'avait été détecté,
le 7 décembre, un premier foyer dans un élevage de 630
canards.</phrase> </para>
<para> <phrase>C'est aussi en Dordogne qu'avait été identifié
le tout premier cas de cette épizootie, le 24 novembre, dans
la basse-cour d'un particulier à Biras.</phrase> ...
```

testParcoursArbreSAX.java

2.1 SIMPLE API FOR XML (SAX)

Exécution et résultat

```
balise de fin: phrase
balise de début: phrase
Le ministère de l'agriculture recense à ce jour un total de «
15 foyers d'influenza aviaire » dans cinq départements du Sud-
Ouest.
balise de fin: phrase
balise de fin: para
balise de début: para
balise de début: phrase
Un cas de grippe de type H5N1 a été détecté et confirmé lundi
dans un élevage de 1 070 canards à Bosset (Dordogne), près de
Bergerac, indique le ministère sur son site Internet.
balise de fin: phrase
balise de début: phrase
C'est dans cette même commune qu'avait été détecté, le 7
décembre, un premier foyer dans un élevage de 630 canards.
balise de fin: phrase
balise de fin: para
balise de début: para
balise de début: phrase ...
```

Caractéristiques (1)

Nécessité d'une plate-forme de référence pour la gestion d'arborescence XML

Analyse et construction d'une arborescence de « nœuds » en mémoire,

Navigation, Modification, Sauvegarde partielle ou totale

Définition par l'Object Management Group (OMG)

1998 – Version 1.0

2000 - Version 2.0

2004 - Version 3.0 (<u>www.w3.org/DOM/</u>)

2015 - Version 4.0 (www.w3.org/TR/2015/REC-dom-20151119/)

Langages supportées

Pascal, C++ (Apache), Java, Perl (module), Python, Lingo (Director/Flash)

Caractéristiques (2) - Noeud

Elément constitutif de l'arborescence

Possibilité d'avoir des fils et un père

Typologie des nœuds

nœud document ou nœud racine (DOCUMENT_NODE) nœud balise (ELEMENT_NODE) nœud attribut (ATTRIBUTE_NODE) nœud texte (TEXT_NODE) nœud commentaire (COMMENT_NODE) nœud entité (ENTITY_NODE)

Création à chaque nœud d'un ou plusieurs objets

Beaucoup de mémoire utilisée Inadaptée pour les grands documents

Implémentation en Java

Librairie JAXP

+

Package org.w3c.dom (pas de support de Oracle)

interfaces (17) et une exception définissant le squelette des classes DOM.

Implémentation en Java (2) – Interface Node

Propriétés du nœud

String getNodeName() String getNamespaceURI()

NamedNodeMap getAttributes() liste des nœuds attributs d'un nœud balise

Parcours dans l'arborescence des nœuds-balises

Node Jist getChildNodes()
Node getFirstChild()
Node getLastChild()
Node getNextSibling()
Node getPreviousSibling()
Iiste des nœuds-fils
premier des nœud-fils
dernier des nœud-fils
prochain nœud-frère
précédent nœud-frère

Node **getParentNode**() nœud père

Modification de l'arborescence

Node removeChild(Node oldChild) suppression d'un nœud

Node appendChild(Node newChild) ajout d'un nœud

Node replaceChild(Node newChild, Node oldChild) remplacement d'un nœud

void setNodeValue(String nodeValue) modification de la valeur d'un noeud

Construction d'un objet DOM

```
public class GestionArbreDom
Document document = null;
public GestionArbreDom(InputStream isXml) {
// création d'une instance d'API DOM
DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
// création d'un objet de construction
DocumentBuilder builder = null;
try {builder = factory.newDocumentBuilder();}
catch (ParserConfigurationException e) {}
// parcours et construction de l'arbre
try {document = builder.parse(isXml);}
catch (IOException e) {}
catch (SAXException e){}
```

Parcours de l'arbre XML

```
/** Parcours complet de l'arbre */
public void ParcoursArbre() {
if (document != null) ParcoursArbre(document);
/** Exploration récursive d'un noeud de l'arbre
 * @param node noeud */
private void ParcoursArbre(Node node) {
int type = node.getNodeType();
switch (type){
case Node.DOCUMENT_NODE:{
System.out.println("début du document:");
ParcoursArbre(((Document)node).getDocumentElement());
System.out.println("fin du document:");
break;
case Node.TEXT NODE:{
String st = node.getNodeValue().trim();
if (st.length() > 0) System.out.println(st); break; }
```

Lecture des balises, des attributs et des données

```
case Node.ELEMENT_NODE:{
System.out.print("balise de début: " + node.getNodeName());
// attributs
NamedNodeMap attrs = node.getAttributes();
for (int i = 0; i < attrs.getLength(); i++){</pre>
     Node attr = attrs.item(i);
     // nom
     System.out.print(" " + attr.getNodeName() + "=\"");
     // valeur
     System.out.print(attr.getNodeValue()+"\"");
} System.out.println();
NodeList children = node.getChildNodes();
if (children != null)
     int len = children.getLength();
     for (int i = 0; i < len; i++)
     ParcoursArbre(children.item(i));
System.out.println("balise de fin: " + node.getNodeName());
break;
```

Extractions à partir d'un fichier XML

```
import java.io.*;
/ * *
 * @author montacie
public class testGestionArbreDom {
public static void main(String[] args) {
     File fxml = new File("xml/article.xml");
     FileInputStream isxml = null;
     try {isxml = new FileInputStream(fxml);}
     catch (IOException e) {}
     GestionArbreDom ges = new GestionArbreDom(isxml);
     qes.ParcoursArbre();
```