IUP GMI 2 - Java et autres technologies du Web
Tout document autorisé

Michelle Joab - Pierre Pompidor
18 mai 2004

Rédigez les deux parties sur deux copies séparées

Partie I (Pierre Pompidor) : décrivez le résultat d'une transformation XSL

Remarque :
- la transformation XSL invoquée par un script PHP (non indiqué) génère un fichier SVG qui sera affiché dans le navigateur via une page HTML (non indiquée)
- rien n'est à savoir sur SVG (tout est expliqué dans le code XSL)
- ne paraphrasez pas le code, mais faites un dessin d'écran et donner des explications sur les expressions XPath

Le document XML

```
<?xml version='1.0' encoding='ISO-8859-1' ?>

<magasin>

<rayons>
   <BandesDessinées couleur='red' />
   <Romans couleur='pink' />
   <Informatique couleur='yellow' />
   <Caisse couleur='green' />
   <travée couleur='indigo' />
</rayons>

<produits>
   <BandesDessinées produits='BandesDessinées.xml'>
      <produit nom='La ballade de la mer salée' ref='1201' prix='15' />
      <produit nom='La Maison dorée de Samarkand' ref='1202' prix='15' />
      <produit nom='Mû' ref='1203' prix='20' />
   </BandesDessinées>
   <Romans produits='Romans.xml'>
      <produit nom='Comme un collégien' ref='3030' prix='12' />
      <produit nom='La Taupe' ref='3031' prix='14' />
      <produit nom='Les gens de Smiley' ref='3032' prix='12' />
   </Romans>
</produits>

<plan>
   <travée colonne='3' début='0' fin='2' />
   <BandesDessinées ligne='0' début='0' fin='2' />
   <travée ligne='1' début='0' fin='3' />
   <Romans ligne='2' début='0' fin='2' />
   <travée ligne='3' début='0' fin='3' />
   <Informatique colonne='4' début='0' fin='4' />
   <Caisse ligne='4' début='0' fin='3' />
</plan>

</magasin>
```

La feuille de style XSL

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" >

<xsl:output method="xml" indent="yes" encoding="ISO-8859-1" />
<xsl:strip-space elements="*" />

<xsl:template match="magasin">
   <svg xmlns="http://www.w3.org/2000/svg" width="800" height="600">
      <xsl:apply-templates select="plan/*"/>
   </svg>
</xsl:template>


<xsl:template match="plan/*[@colonne]" xmlns="http://www.w3.org/2000/svg">
    <xsl:variable name="balise" select="name()" />
    <xsl:variable name="hauteur" select="(@fin - @début + 1) * 50" />
    <xsl:variable name="abscisse" select="@colonne * 50" />
    <xsl:variable name="ordonnee" select="@début * 50 + 50" />

    <!-- la balise SVG rect dessine un rectangle
         x="abscisse coin supérieur gauche"
         y="ordonnée coin supérieur gauche"
         width="largeur"
         height="hauteur"
         style="fill:couleur du rectangle" -->
    <rect>
       <xsl:attribute name="id">     <xsl:value-of select="name()"/>     </xsl:attribute>
       <xsl:attribute name="x">      <xsl:value-of select="$abscisse" /> </xsl:attribute>
       <xsl:attribute name="y">      <xsl:value-of select="$ordonnee" /> </xsl:attribute>
       <xsl:attribute name="height"> <xsl:value-of select="$hauteur" />  </xsl:attribute>
       <xsl:attribute name="width">  <xsl:value-of select="50" />        </xsl:attribute>
  <xsl:attribute name="style">fill:<xsl:value-of select="/magasin/rayons/*[name() = $balise]/@couleur" />;
       </xsl:attribute>
    </rect>

    <xsl:variable name="longueur" select="string-length(name())" />
    <xsl:if test="$hauteur &gt; 50">
       <!-- la balise SVG text écrit du texte
            le sous-attribut writing-mode a pour valeur tb quand le texte est écrit verticalement
       -->
       <text xmlns="http://www.w3.org/2000/svg" style="fill:black;writing-mode:tb;">
          <xsl:attribute name="x"> <xsl:value-of select="$abscisse + 35" /> </xsl:attribute>
          <xsl:attribute name="y"> <xsl:value-of select="$ordonnee + 45" /> </xsl:attribute>
          <xsl:value-of select="name()" />
         (<xsl:value-of select="count(//*[name() = $balise]/*)" />)
       </text>
    </xsl:if>

</xsl:template>


<xsl:template match="plan/*[@ligne]" xmlns="http://www.w3.org/2000/svg">
    <xsl:variable name="balise" select="name()" />
    <xsl:variable name="largeur" select="(@fin - @début + 1) * 50" />
    <xsl:variable name="abscisse" select="@début * 50" />
    <xsl:variable name="ordonnee" select="@ligne * 50 + 50" />
```

```
    <rect>
       <xsl:attribute name="id">     <xsl:value-of select="name()"/></xsl:attribute>
       <xsl:attribute name="x">      <xsl:value-of select="$abscisse" /> </xsl:attribute>
       <xsl:attribute name="y">      <xsl:value-of select="$ordonnee" /> </xsl:attribute>
       <xsl:attribute name="width">  <xsl:value-of select="$largeur" /> </xsl:attribute>
       <xsl:attribute name="height"> <xsl:value-of select="50" /> </xsl:attribute>
       <xsl:attribute name="style">fill:<xsl:value-of select="/magasin/rayons/*[name() = $balise]/@couleur" /
    </rect>

    <xsl:variable name="longueur" select="string-length(name())" />
    <xsl:if test="$largeur &gt; 50">
       <text xmlns="http://www.w3.org/2000/svg" style="fill:black;">
          <xsl:attribute name="x"> <xsl:value-of select="$abscisse + 15" /> </xsl:attribute>
          <xsl:attribute name="y"> <xsl:value-of select="$ordonnee + 30" /> </xsl:attribute>
          <xsl:value-of select="name()" />
         (<xsl:value-of select="count(//*[name() = $balise]/*)" />)
       </text>
    </xsl:if>

</xsl:template>

</xsl:stylesheet>
```

Partie II (Michelle Joab) : Java

On donne le code du programme Java fourni en annexe.

Question 1

Faire un schéma précisant la structure des composants utilisés (Jpanel) dans la fenêtre principale (JFrame).

Question 2

Quels sont les composants du programmeăavec lesquels l'utilisateur peut interagir ? Quel est le mode de contrôle choisiă ?

Question 3

Faire un schéma précisant les grandes lignes de l'interface utilisateur du programme.

Annexe : programme Phases.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.net.URL;

public class Phases extends JPanel {
    final static int NUM_IMAGES = 8;
    final static int START_INDEX = 3;

    ImageIcon[] images = new ImageIcon[NUM_IMAGES];
    JPanel mainPanel, selectPanel, displayPanel;
    JComboBox phaseChoices = null;
    JLabel phaseIconLabel = null;

    public Phases() {
        selectPanel = new JPanel();
```

```
        displayPanel = new JPanel();
        addWidgets();
        mainPanel = new JPanel();
        mainPanel.setLayout(new BoxLayout(mainPanel, BoxLayout.PAGE_AXIS));
        mainPanel.add(selectPanel);
        mainPanel.add(displayPanel);
        this.add(mainPanel);
    }

    private void addWidgets() {
        for (int i = 0; i < NUM_IMAGES; i++) {
            images[i] = createImageIcon("image" + i + ".jpg");
        }
        phaseIconLabel = new JLabel();

        String[] phases = { "New", "Waxing Crescent", "First Quarter",
                            "Waxing Gibbous", "Full", "Waning Gibbous",
                            "Third Quarter", "Waning Crescent" };
        phaseChoices = new JComboBox(phases);
        phaseChoices.setSelectedIndex(START_INDEX);
        phaseIconLabel.setIcon(images[START_INDEX]);
        phaseIconLabel.setText("");
        displayPanel.add(phaseIconLabel);
        selectPanel.add(phaseChoices);
        phaseChoices.addActionListener(new MonAction());
    }

    protected static ImageIcon createImageIcon(String path) {
        java.net.URL imageURL = LunarPhases.class.getResource(path);
        if (imageURL == null) {
            System.err.println("Resource not found: "+ path);
            return null;
        } else {
            return new ImageIcon(imageURL);
        }
    }

    private static void createAndShowGUI() {
        Phases phases = new Phases();
        JFrame PhasesFrame = new JFrame("Phases");
        PhasesFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        PhasesFrame.setContentPane(phases);
        PhasesFrame.pack();
        PhasesFrame.setVisible(true);
    }

    public static void main(String[] args) {
        createAndShowGUI();
    }

    public class MonAction implements ActionListener {

    public void actionPerformed(ActionEvent event) {
        if ("comboBoxChanged".equals(event.getActionCommand())) {
            phaseIconLabel.setIcon(images[phaseChoices.getSelectedIndex()]);
        }
    }
};
}
```