

20 juin 2007

**Contrôle d'informatique no 4**

Durée : 1 heure 45'

Nom : .....

Groupe : 

Prénom : .....

Barème sur 100 points

No	1	2	3
Total points	10 points	48 points	42 points

**Remarque générale :** Toutes les questions qui suivent se réfèrent au langage de programmation Java (à partir du JDK 5.0) et les réponses doivent être rédigées à l'encre et d'une manière propre sur ces feuilles agrafées.

**Sujet no 1.**

Indiquer si les affirmations suivantes sont vraies ou fausses (par VRAI ou FAUX à l'endroit prévu à cet effet) :

- a) Le programmeur peut imposer qu'un objet container (de premier niveau ou intermédiaire) ne dispose d'aucun gestionnaire de mise en forme (Layout Manager).

.....

- b) Le gestionnaire de disposition **BorderLayout** ne peut être utilisé que pour la mise en forme des conteneurs de premier niveau.

.....

- c) Tout fichier HTML contient au moins une balise (tag) **<APPLET>**.

.....

- d) Deux applets lancées par la même page HTML peuvent communiquer entre elles.

.....

**Remarque :** chaque réponse correcte vaut +2.5 points, chaque réponse fausse vaut -1.5 points et chaque question sans réponse vaut 0 point.

## Sujet no 2.

Soit le fichier FenPrincipale.java contenant le code source suivant :

```
package cms_ctr4;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class FenPrincipale extends JFrame
    implements ActionListener, KeyListener
{
    JPanel jpanCenter;
    private JLabel jlab1, jlab2;
    private JButton jbtTab[];
    String tabPointsCard[] = {"North", "South", "West",
                             "East", "Center"};
    String tabMessages[] = {"De bello Gallico !", "Ave Caesar !",
                            "Raje, reze, reye !", "Whaw fwypw aop !"};

    int n=0;

    FenPrincipale( )
    {
        this.setTitle("Caesar");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setResizable(false);

        jlab1 = new JLabel("La source");
        jlab1.setPreferredSize(new Dimension(250,50));
        jlab1.setHorizontalAlignment(SwingConstants.CENTER);
        jlab1.setBackground(Color.YELLOW);
        jlab1.setFont(new Font("Arial", Font.BOLD, 24));
        this.add(jlab1, BorderLayout.NORTH);

        jlab2 = new JLabel("La destination");
        jlab2.setPreferredSize(new Dimension(250,50));
        jlab2.setHorizontalAlignment(SwingConstants.CENTER);
        jlab2.setFont(new Font("Arial", Font.BOLD, 24));
        this.add(jlab2, BorderLayout.SOUTH);

        jpanCenter = new JPanel();
        jpanCenter.setLayout(new BorderLayout());
        jbtTab = new JButton[5];
        for(int i = 0; i < 5; i++)
        {
            jbtTab[i] = new JButton("Bouton_" + i);
            jbtTab[i].setPreferredSize(new Dimension(150,50));
            jbtTab[i].setFont(new Font("Arial",Font.BOLD,24));
            jbtTab[i].addActionListener(this);
            jbtTab[i].addKeyListener(this);
            jpanCenter.add(jbtTab[i], tabPointsCard[i]);
        }
        this.add(jpanCenter, "Center");

        this.pack();
        this.setVisible(true);
    }
    //fin du constructeur FenPrincipale
}
```

```

public void actionPerformed(ActionEvent e)
{
    JButton jbtTemp = (JButton)(e.getSource());
    if(jbtTemp == jbtTab[0])
    {
        if(n==3)    n=0;
        else        n++;
        jlab1.setText(tabMessages[n]);
    } else if (jbtTemp == jbtTab[1])
    {
        if(n==0)    n=3;
        else        n--;
        jlab1.setText(tabMessages[n]);
    } else if (jbtTemp == jbtTab[2])
    {
        jlab2.setText(coder(jlab1.getText(), 4));
    } else if (jbtTemp == jbtTab[3])
    {
        jlab2.setText(transformer(jlab1.getText()));
    } else if (jbtTemp == jbtTab[4])
    {
        jlab1.setText("La source");
        jlab2.setText("La destination");
    }
} //fin de la méthode actionPerformed

public void keyPressed(KeyEvent e)
{
    if(e.getKeyCode() == KeyEvent.VK_ENTER)
    {
        jlab2.setText(coder(jlab1.getText(), 4));
    } else if (e.getKeyChar() == '-')
    {
        if(n==0)    n=3;
        else        n--;
        jlab1.setText(tabMessages[n]);
    }
} //fin de la méthode keyPressed

public void keyReleased(KeyEvent e) { }
public void keyTyped(KeyEvent e) { }

private String coder(String arg, int pas)
{
    if(pas < 1 || pas >= 26)    pas = 4;
    int longueur = arg.length();
    String res = "";
    char car;
    for(int i=0; i<longueur; i++)
    {
        car = arg.charAt(i);
        if(car>='a' && car<='z')
        {
            if(car + pas <= 'z')
                res = res + (char)(car + pas);
            else
                res = res + (char)(car + pas - 26);
        } else if (car >= 'A' && car <= 'Z')
        {
            if(car + pas <= 'Z')
                res = res + (char)(car+pas);
            else
                res = res + (char)(car + pas - 26);
        } else
        {
            res = res + car;
        }
    }
    return res;
} //fin de la méthode coder

```

```

private String transformer(String arg)
{
    String res = "";
    for(int i=arg.length()-1; i>=0; i--)
    {
        res = res + arg.charAt(i);
    }
    return res;
} //fin de la méthode transformer
} //fin de la classe FenPrincipale

```

Suite à l'instanciation de la classe FenPrincipale qui vient d'être présentée, l'utilisateur dispose d'une interface graphique (GUI) comme celle montrée dans la figure ci-dessous :



## 2.1 Compléter les deux phrases suivantes :

- a) Une fois l'interface graphique affichée, le fait d'appuyer la touche "**Enter**" du clavier a le même effet que de cliquer avec le pointeur de la souris sur le bouton dont le texte est .....
- b) Une fois l'interface graphique affichée, le fait d'appuyer la touche "-" du clavier a le même effet que de cliquer avec le pointeur de la souris sur le bouton dont le texte est .....

**Rappel** : Selon le code Unicode, les **26** lettres latines (sans accent ni cédille) se succède dans l'ordre suivant (qui est le même pour les majuscules comme pour les minuscules) :

**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**

**2.2** Par la suite, "**l'étiquette nord**" désigne l'étiquette (**JLabel**) placée dans la région Nord de la fenêtre graphique, au dessus du bouton **Bouton\_0**, et qui affiche initialement le texte **La source**, tandis que par "**l'étiquette sud**" on désigne l'étiquette (**JLabel**) placée dans la région Sud de la fenêtre graphique, au dessous du bouton **Bouton\_1**, et qui affiche initialement le texte **La destination**.

Indiquer quels seront les textes finaux affichés dans **l'étiquette nord** et dans **l'étiquette sud** de la fenêtre graphique "Caesar" suite à chaque nouvelle instanciation de la classe FenPrincipale si, après l'affichage de l'interface graphique, l'utilisateur clique sur le ou les boutons suivants :

a) Bouton\_0

**l'étiquette nord** .....

**l'étiquette sud** .....

b) Bouton\_1

**l'étiquette nord** .....

**l'étiquette sud** .....

c) Bouton\_0 → Bouton\_2

**l'étiquette nord** .....

**l'étiquette sud** .....

d) Bouton\_0 → Bouton\_3

**l'étiquette nord** .....

**l'étiquette sud** .....

e) Bouton\_1 → Bouton\_2

**l'étiquette nord** .....

**l'étiquette sud** .....

f) Bouton\_0 → Bouton\_0 → Bouton\_2

**l'étiquette nord** .....

**l'étiquette sud** .....

g) Bouton\_0 → Bouton\_1

**l'étiquette nord** .....

**l'étiquette sud** .....

h) Bouton\_0 → Bouton\_1 → Bouton\_2 → Bouton\_3

**l'étiquette nord** .....

**l'étiquette sud** .....

i) Bouton\_0 → Bouton\_3 → Bouton\_4

**l'étiquette nord** .....

**l'étiquette sud** .....

j) Bouton\_0 → Bouton\_3 → Bouton\_4 → Bouton\_0 → Bouton\_0 → Bouton\_2

**l'étiquette nord** .....

**l'étiquette sud** .....

### Sujet no 3.

Le but de cet exercice est d'écrire une application Java autonome qui traite les informations contenues dans un **fichier texte** (au format XML) qui correspond à la base de données d'une petite bibliothèque abritant un nombre quelconque de livres.

Dans le **fichier texte** appelé "*source.xml*" ont été enregistrées, à l'aide des balises adéquates, des informations pertinentes pour chaque livre de la bibliothèque, par exemple : son code ISBN, la langue de l'ouvrage, le sujet, les noms et les prénoms des auteurs, le fait qu'il s'agisse éventuellement d'une traduction, le titre du livre, la maison d'édition ou l'année de sa parution.

L'application Java qui sera réalisée doit :

- lire le **fichier texte** appelé "*source.xml*" qui est stocké dans le dossier **BaseDonnees** de la partition **C:** ;
- déterminer le nombre de livres référencés et extraire le titre et l'année de parution de chaque livre ;
- écrire dans un **fichier texte** appelé "*synthese.txt*" (qui sera créé pour l'occasion sur la partition **C:** et dans le même dossier **BaseDonnees**) ;
  - o sur la première ligne, le nombre de livres de la bibliothèque ;
  - o sur des lignes distinctes, le titre et l'année de parution de chaque livre, ainsi que le nombre d'années passées depuis son parution (jusqu'en 2007).

On peut consulter plus loin l'exemple d'un fichier texte "*source.xml*" et du fichier texte "*synthese.txt*" obtenu suite à l'exécution de l'application Java.

Le projet à concevoir est formé d'une seule classe publique appelée **CP\_Ctr4\_3**, placée dans un package nommé **cms\_ctr4**.

La classe **CP\_Ctr4\_3** contient la méthode **main** qui réalise la succession des opérations principales suivantes :

- on crée une connexion **en lecture** vers le fichier texte "*source.xml*" déjà existant dans le dossier **BaseDonnees** de la partition **C:** ;
- on fait une première lecture de ce fichier et on détermine le nombre de livres qu'il référence, en comptant, par exemple, le nombre de balises **LIVRE** (ouvrantes ou fermantes) ;
- on ferme la connexion entre le programme Java et le fichier texte "*source.xml*" ;

- si le nombre de livres référencés est zéro, on affiche un message approprié dans la fenêtre console (par exemple : ***Il n'y a pas de livres dans la BD !***) et on arrête le programme (par un appel à la méthode ***exit*** avec un code de retour ***-1***) ;
- autrement, le programme continue son exécution comme indiqué par la suite ;
- on crée une nouvelle connexion **en lecture** vers le même fichier texte "***source.xml***" ;
- on crée une connexion **en écriture** vers un fichier texte "***synthese.txt***" (qui sera créé dans le dossier ***BaseDonnees*** de la partition ***C:***) ;
- on écrit la première ligne du fichier "***synthese.txt***" en précisant le nombre de livres dans la bibliothèque ;
- on lit le fichier texte "***source.xml***" ligne après ligne et, au fur et à mesure qu'on identifie les informations recherchées, à savoir le titre (correspondant à la balise **TITRE**) et la date de parution (correspondant à la balise **DATEPUB**), on écrit dans le fichier texte "***synthese.txt***" le titre et l'année de parution de chaque livre, ainsi que le nombre d'années passées depuis son parution (jusqu'en 2007) ;
- finalement, on ferme les connexions entre le programme Java et les fichiers textes "***synthese.txt***" et "***source.xml***".

#### **Indications :**

- on ne traite que les lignes non vides du fichier "***source.xml***" ;
- après chaque lecture d'une ligne non vide du fichier "***source.xml***", on élimine d'abord les éventuels espaces ou tabulations du début ou de la fin de la ligne (par un appel à la méthode **trim**) ;
- puis, afin de trouver des informations pertinentes sur la ligne, on peut utiliser la classe ***StringTokenizer*** en précisant des séparateurs adéquats dans la chaîne de caractères passée comme deuxième argument au constructeur de cette classe ;
- le programme doit fonctionner correctement même si les balises sont écrites avec des lettres minuscules dans le fichier texte "***source.xml***" ;
- pour calculer le nombre d'années passées de la parution d'un livre jusqu'en 2007, il faut faire d'abord une transformation (un parsing) de la chaîne de caractères contenant la date de parution du livre en nombre réel ;
- dans le fichier texte "***synthese.txt***", les titres des livres doivent apparaître entre des guillemets.



Exemple d'un fichier "*source.xml*" :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BIBLIO SUJET="XML">
  <LIVRE ISBN="9782212090819" LANG="fr" SUJET="applications">
    <AUTEUR>
      <PRENOM>Jean-Christophe</PRENOM>
      <NOM>Bernadac</NOM>
    </AUTEUR>
    <AUTEUR>
      <PRENOM>François</PRENOM>
      <NOM>Knab</NOM>
    </AUTEUR>
    <TITRE>Construire une application XML</TITRE>
    <MAISON_EDITION>
      <NOM>Eyrolles</NOM>
      <LIEU>Paris</LIEU>
    </MAISON_EDITION>
    <DATEPUB>1999</DATEPUB>
  </LIVRE>
  <LIVRE ISBN="9782212090529" LANG="fr" SUJET="général">
    <AUTEUR>
      <PRENOM>Alain</PRENOM>
      <NOM>Michard</NOM>
    </AUTEUR>
    <TITRE>XML, Langage et Applications</TITRE>
    <MAISON_EDITION>
      <NOM>Eyrolles</NOM>
      <LIEU>Paris</LIEU>
    </MAISON_EDITION>
    <DATEPUB>1998</DATEPUB>
  </LIVRE>
  <LIVRE ISBN="9782840825685" LANG="fr" SUJET="applications">
    <AUTEUR>
      <PRENOM>William J.</PRENOM>
      <NOM>Pardi</NOM>
    </AUTEUR>
    <TRADUCTEUR PREFIX="adapté de l'anglais par">
      <PRENOM>James</PRENOM>
      <NOM>Guerin</NOM>
    </TRADUCTEUR>
    <TITRE>XML en Action</TITRE>
    <MAISON_EDITION>
      <NOM>Microsoft Press</NOM>
      <LIEU>Paris</LIEU>
    </MAISON_EDITION>
    <DATEPUB>1999</DATEPUB>
  </LIVRE>
</BIBLIO>
```

Exemple du fichier texte "***synthese.txt***" obtenu à partir du fichier texte "***source.xml***" suite à l'exécution de l'application autonome :

Il y a 3 livres dans la bibliothèque :

"Construire une application XML" publié en 1999, il y a 8 ans.

"XML, Langage et Applications" publié en 1998, il y a 9 ans.

"XML en Action" publié en 1999, il y a 8 ans.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

This image shows a full page of a handwriting practice worksheet. It consists of approximately 28 horizontal rows. Each row is defined by two parallel dotted lines, creating a series of uniform gaps for letter height. The lines are evenly spaced across the entire page, providing a guide for consistent letter formation. There is no text or other markings on the page.

[illegible]