EPF – Lausanne *COURS DE MATHEMATIQUES SPECIALES*

28 mai 2008

Contrôle d'informatique no 4

Durée : 1 heure 45'

Nom : …… Groupe : ☐

Prénom : ……

Barème sur 100 points

No	1	2	3
Total points	34 points	34 points	32 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.

On présente ci-dessous le contenu du fichier CP_Ctr4_1.java et on demande d'écrire les résultats affichés à l'écran suite à l'exécution de l'application autonome Java correspondante.

```
package cms_ctr4;

public class CP_Ctr4_1
{    public static void main(String[] args)
     {    String tab[] = {"12/3=4",  "var=234.0",
                          "123+1=124", "2*3=6" ,"12.3*2=24.6"};
          //Attention au "sens de parcours" !
          for(int i=tab.length-1; i>=0; i--)
          {    try
               {
                    tab[i] = verifier(tab[i], i);
                    System.out.println("Suite de try !");
               }
               catch(ToucheException e)
               {
                    System.out.println("La touche " + e.champ
                                       + " n'est pas valide !");
                    tab[i] = "Erreur de touche !";
               }
```

```
                catch(EgalException e)
                {
                     System.out.println(e.getMessage());
                }
                System.out.println("A la fin de la boucle " +
                                   (i+1) + ".");
                System.out.println("-------------------------");
           }

           System.out.println("-------------------------");

           for(int i=0; i<tab.length; i++)
           {    System.out.println(tab[i]);
                //Attention au "piège" !
                System.out.println("La boucle " + i + 1 + "!");
                System.out.println("*************************");
           }
           System.out.println("A la prochaine !");
           System.out.println("*************************");
      }//fin de la méthode main

      static String verifier(String arg1, int arg2)
                              throws ToucheException
      {    char car = arg1.charAt(arg2);
           try
           {    if(car >='0' && car <='9')
                {
                     throw new ToucheException(car);
                }else if(car == '+' || car == '-'
                          || car == '*' || car == '/')
                {
                     throw new OperationException(car);
                }else if(car == '=')
                {
                     throw new EgalException(car);
                }
           }
           catch(OperationException e)
           {
                return e.getMessage();
           }
           finally
           {
                System.out.println("Au revoir !");
           }
           System.out.println("La fin de verifier !");
           return arg1;

      }//fin de la méthode verifier

}//fin de la classe CP_Ctr4_1
```

```
class ToucheException extends Exception
{
     char champ;
     ToucheException(char arg)
     {
          super("Problème de type " + arg + " !");
          champ = arg;
     }
}//fin de la classe ToucheException


class OperationException extends Exception
{
     char champ;
     OperationException(char arg)
     {
          super("Problème de type " + arg + " !");
          System.out.println("Faites attention !");
          champ = arg;
     }
}//fin de la classe OperationException


class EgalException extends RuntimeException
{
     char champ;
     EgalException(char arg)
     {
          super("Problème de type " + arg + " !");
          champ = arg;
     }
}//fin de la classe EgalException
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Sujet no 2.

Soit le fichier CP_Ctr4_2.java contenant le code source suivant :

```
package cms_ctr4;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class CP_Ctr4_2
{
     public static void main(String[] args)
     {
          new VivaPoly();
     }//fin de la méthode main
}//fin de la classe principale CP_Ctr4_2

class VivaPoly extends JFrame implements ActionListener
{
     private JButton jbPers, jbAux, jbPP, jbMelange;
     private JPanel panRes;
     String tab_1[] = {"dansé", "chanté", "mangé", "fêté"};
     String tab_2[] = {"ont","avez","avons","a","as","ai"};
     String tab_3[] = {"J'","Tu","Elle","Nous","Vous","Ils"};
     int iUn=0, iDeux=4;

     public VivaPoly( )
     {    setTitle("Grande fête");
          setResizable(false);
          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

          jbPers = new JButton("Personne");
          jbPers.setPreferredSize(new Dimension(450,50));
          jbPers.addActionListener(this);
          jbAux = new JButton("Auxiliaire");
          jbAux.setPreferredSize(new Dimension(100,100));
          jbAux.addActionListener(this);
          jbPP = new JButton("PP");
          jbPP.setPreferredSize(new Dimension(100,100));
          jbPP.addActionListener(this);
          jbMelange = new JButton("Mélange");
          jbMelange.setPreferredSize(new Dimension(450,50));
          jbMelange.addActionListener(this);

          add(jbPers, "North");
          add(jbAux, "West");
          add(jbPP,"East");
          add(jbMelange, "South");
```

```
            panRes = new MonPaneau();
            add(panRes);
            pack();
            setVisible(true);
        }//fin du constructeur
        public void actionPerformed(ActionEvent arg)
        {
            if(arg.getSource() == jbPers)
            {
                //l'opératuer % est l'opérateur modulo
                iUn = (iUn+1) % 6;
                ((JButton)jbPers).setText( tab_3[iUn] );
                ((JButton)jbAux).setText( tab_2[5-iUn] );
            }else if(arg.getSource() == jbAux)
            {
                iUn = (iUn+1) % 6;
                ((JButton)jbPers).setText( tab_3[iUn] );
                ((JButton)jbAux).setText( tab_2[5-iUn] );
            }else if(arg.getSource() == jbPP)
            {
                iDeux = (iDeux+1) % 4;
                ((JButton)jbPP).setText( tab_1[iDeux] );
                repaint();
            }else
            {
                iUn = 11;
                iDeux = 22;
                jbMelange.setText("C'est parti !");
            }
        }//fin de la méthode actionPerformed

        class MonPaneau extends JPanel
        {
            String [] message = {"Bonne fête !", "Bon appétit !",
                                 "Bon chant !", "Bonne danse !"};
            private MonPaneau()
            {
                setPreferredSize(new Dimension(250,100));
            }

            @Override
            public void paint(Graphics g)
            {
                super.paint(g);
                g.setFont(new Font("Arial", Font.BOLD, 16));
                iDeux = iDeux % 4;
                g.drawString(message[iDeux], 70, 50);
            }
        }//fin de la classe MonPaneau
    }//fin de la classe VivaPoly
```

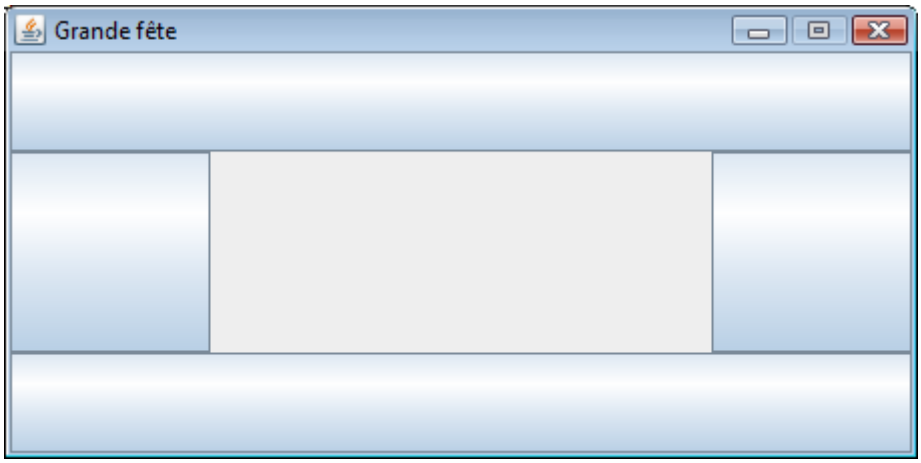
Suite à l'instanciation de la classe **VivaPoly** par la méthode ***main***, l'utilisateur dispose d'une interface graphique (GUI) comme celle montrée dans la figure ci-dessous :

Par la suite :

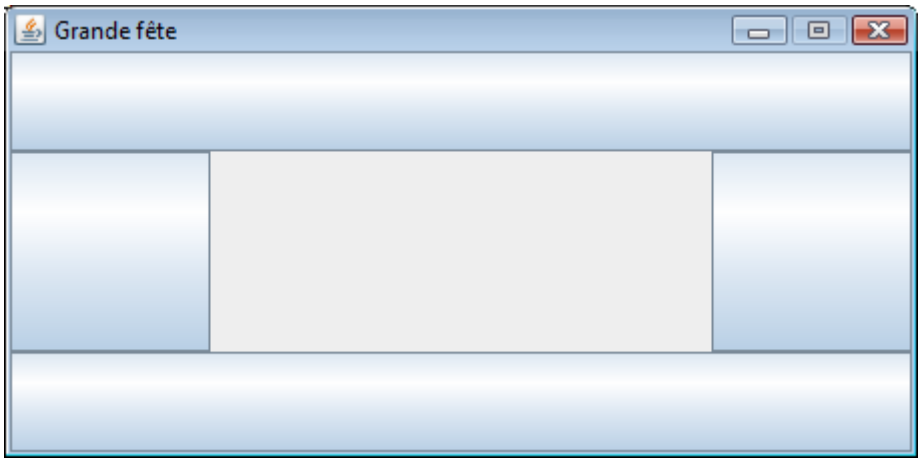
- le "**Bouton nord**" désigne le bouton (***JButton***) placé dans la région Nord de la fenêtre graphique et qui affiche initialement le texte **Personne** ;
- le "**Bouton ouest**" désigne le bouton (***JButton***) placé dans la région Ouest de la fenêtre graphique et qui affiche initialement le texte **Auxiliaire** ;
- le "**Bouton est**" désigne le bouton (***JButton***) placé dans la région Est de la fenêtre graphique et qui affiche initialement le texte **PP** ;
- le "**Bouton sud**" désigne le bouton (***JButton***) placé dans la région Sud de la fenêtre graphique et qui affiche initialement le texte **Mélange** ;
- la "**Zone centrale**" désigne le panneau (***JPanel***) placé dans la région centrale de la fenêtre graphique et qui affiche initialement le texte **Bonne fête !** .

Indiquer sur les copies incomplètes de l'interface graphique présentées ci-dessous quels seront les textes finaux affichés par les boutons et la zone centrale de la fenêtre graphique "Grande fête" suite à chaque nouvelle instanciation de la classe **VivaPoly** si, après l'affichage de l'interface graphique, l'utilisateur clique (dans l'ordre) sur le ou les boutons suivants :

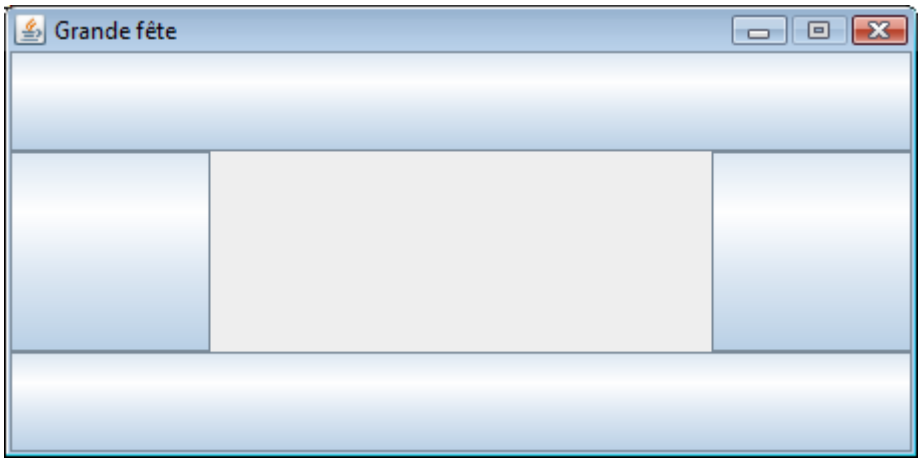
a) Bouton nord



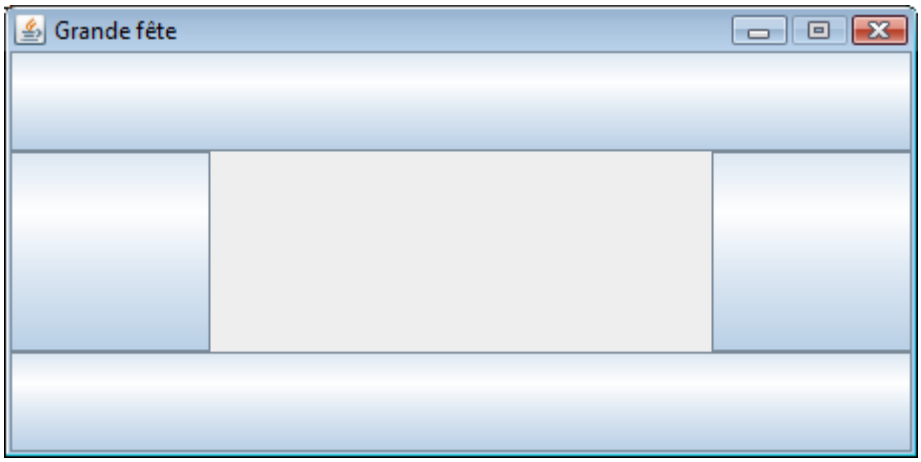
b) Bouton ouest → Bouton ouest



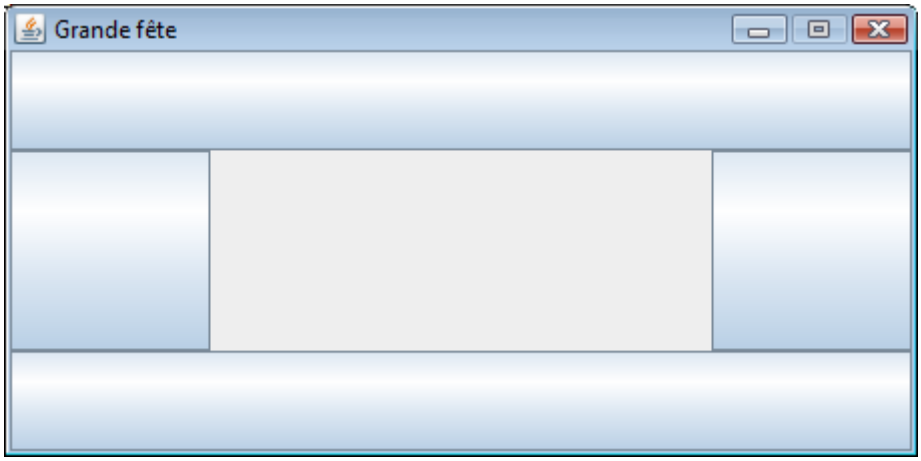
c) Bouton est → Bouton est → Bouton est



d) Bouton sud → Bouton est → Bouton ouest → Bouton nord



e) Bouton nord → Bouton est → Bouton sud → Bouton ouest



Sujet no 3.

Le but de cet exercice est d'écrire une application Java autonome qui crée, à partir d'un fichier texte **brut.txt** qui existe déjà dans le dossier **bd** situé sur la partition **f:**, un nouveau fichier texte appelé **donnes.csv** et qui sera placé dans le même dossier **bd**.

Dans le fichier initial **brut.txt**, on trouve des informations concernant les produits stockés dans un dépôt :

- ce fichier contient un nombre quelconque de lignes ;
- chaque ligne est associée à un produit et elle contient exactement trois données séparées par des **espaces**, à savoir :
    - le code entier d'un produit ;
    - une valeur entière qui représente la quantité du produit ;
    - une valeur réelle qui représente le prix unitaire du produit.

Le nouveau fichier **donnes.csv** doit respecter les consignes suivantes :

- il doit avoir le même nombre de lignes que le fichier **brut.txt** ;
- il faut que chacune de ses lignes contienne exactement 4 données séparées par des **points-virgules**, à savoir :
    - les trois premières données doivent être les mêmes que celles du fichier **brut.txt** ;
    - la dernière donnée doit être une nouvelle information représentant le prix total du produit correspondant obtenu en multipliant la quantité par le prix unitaire.

On peut consulter plus loin l'exemple d'un fichier texte ***brut.txt*** et du fichier texte ***donnes.csv*** obtenu suite à l'exécution de l'application Java.

Le projet à concevoir est formé de deux classes :

- une classe publique ***Auxiliaire*** qui est placée dans un package nommé ***cms_ctr4*** et qui contient une méthode publique et statique appelée ***traiter*** qui réalise le traitement précisé ci-dessus ;
- une classe "principale" publique ***CP_Ctr4_3*** qui est placée dans le même package ***cms_ctr4*** et qui contient la méthode ***main*** qui "se contente" d'appeler convenablement la méthode ***traiter***.

La méthode publique et statique ***traiter*** de la classe ***Auxiliaire*** doit :

- avoir deux arguments :
    - un premier argument de type ***BufferedReader*** qui permet de lire le contenu d'un fichier texte vers lequel il "pointe" (et qui, dans l'appel de la méthode, sera le fichier ***brut.txt***) ;
    - un deuxième argument de type ***String*** qui contient le nom complet (y compris le chemin d'accès absolu) d'un fichier texte qui sera créé par la méthode (et qui, suite à l'appel de la méthode, sera le fichier ***donnes.csv***) ;
- ne pas retourner de résultat ;
- déclarer une variable locale d'un type convenable et y stocker l'adresse d'un nouvel objet Java qui assure la création et l'ouverture en écriture d'un fichier texte correspondant à son deuxième argument ;
- effectuer le traitement indiqué auparavant.

La méthode publique et statique ***main*** de la classe "principale" ***CP_Ctr4_3*** doit :

- déclarer une variable de type ***BufferedReader*** et y stocker l'adresse d'un nouvel objet qui "pointe" vers le fichier ***brut.txt*** ;
- déclarer une variable de type ***String*** et y stocker le nom complet (y compris le chemin d'accès absolu) du fichier **donnees.csv** ;
- utiliser ces deux variables dans l'appel de la méthode statique ***traiter*** ;
- prévoir d'autres éventuelles instructions et clauses nécessaires pour le bon fonctionnement du projet.

Indications :

- on suppose que le contenu du fichier **brut.txt** est correct par rapport aux précisions données auparavant et que le programme a le droit de lire et d'écrire partout dans la partition **f:** ;
- afin de trouver les informations pertinentes sur les lignes du fichier **brut.txt**, on peut utiliser la classe ***StringTokenizer*** ;
- pour calculer le prix total d'un produit, il faut faire d'abord des transformations des chaînes de caractères en nombres réels ;
- il ne faut pas oublier d'importer les packages nécessaires, de traiter ou de déléguer les éventuelles exceptions et de fermer les canaux de communication ouverts après leur utilisation.

Exemple d'un fichier texte "***brut.txt***" :

1111 10 12.3
22222 100 1.5
33 40 6

Exemple du fichier texte "***donnees.csv***" obtenu à partir du fichier texte "***brut.txt***" suite à l'exécution de l'application autonome :

1111;10;12.3;123.0

22222;100;1.5;150.0

33;40;6;240.0

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....