Université Paris XII
IUT de Fontainebleau
Département informatique
FI-2
2003-2004

PARTIEL 1

Seuls sont autorisés, à titre de documents, les listings comprenant explicitement le nom de l'étudiant (et celui-ci seulement) et les notes manuscrites.

Exercice 1.- (**Grille**)

Écrire une applet Java faisant apparaître une grille de trois lignes et cinq colonnes dans laquelle apparaissent dix boutons étiquetés de 0 à 9, représentant les touches d'une calculette, ainsi qu'un champ de texte. Lorsqu'on appuie sur ces boutons, le nombre correspondant doit être affiché dans le champ de texte.

Si on appuie sur '1', '2' puis '3', le nombre 123 sera affiché dans le champ de texte.

Exercice 2.- (**Liste**)

Écrire une application Java faisant apparaître un cadre fermant dans lequel apparaissent deux listes précédées des étiquettes 'fond' et 'avant-plan'. La première liste comprendra les items 'rouge', 'vert' et 'bleu', la seconde les items 'orange', 'jaune' et 'magenta'. Lorsqu'on clique sur un item d'une des listes le fond ou l'avant-plan, suivant le choix de la liste, devra prendre la couleur indiquée.

Exercice 3.- (**Concaténé de mots**)

On a souvent besoin d'une structure de données pour représenter plusieurs mots. On peut utiliser pour cela un tableau de mots ou, mieux, les mots concaténés les uns derrière les autres, avec un indicateur pour repérer chaque fin de mot (ou chaque début, au choix).

La concaténation de plusieurs mots sera représentée par un tableau de caractères et un tableau, de taille le nombre de mots, indiquant l'index de chaque fin de mot dans le tableau de caractères. Par exemple dans le cas des trois mots 'tu', 'tue' et 'tutu', on aura :

$cat =$

t	u	t	u	e	t	u	t	u

et $fin[0] = 1$, $fin[1] = 4$, $fin[2] = 8$.

Écrire une application Java qui demande le nombre de mots puis chacun des mots. Elle doit alors afficher le concaténé des mots (les deux tableaux).

DOCUMENTATION

1 Rappels sur les champs de texte

Les méthodes :

```
String getText()
```

et :

```
void setText(String)
```

de la classe `TextField` permettent, respectivement, d'obtenir la chaîne de caractères écrite par l'utilisateur dans le champ de texte et de placer une chaîne de caractères dans un champ de texte.

2 Grille

Par défaut les composants placés dans un cadre le sont sur la première ligne (virtuelle) puis, lorsqu'il n'y a plus de place sur la seconde et ainsi de suite. Une façon de mettre un peu d'ordre est d'utiliser une grille définie dans la classe :

```
GridLayout
```

dont un constructeur est :

```
void GridLayout(int rows, int cols, int hor, int vert)
```

où `rows` est le nombre de lignes, `cols` le nombre de colonnes de la grille, `hor` et `vert` l'espacement (en pixels) respectivement horizontal et vertical entre deux composants.

La grille est placée dans un cadre grâce à la méthode :

```
void setLayout(LayoutManager)
```

`GridLayout` étant une classe dérivée de `LayoutManager`.

Les composant sont alors placés dans la grille, dans l'ordre chronologique d'ajout.

3 Les couleurs

Les couleurs sont les objets de la classe `Color` du paquetage `java.awt`.

Il y a un certain nombre de couleurs prédéfinies sous la forme de données *statiques* :

```
black blue cyan darkGray gray green
lightGray magenta orange pink red white yellow
```

de la classe `Color`.

On peut modifier la couleur de fond ou d'avant-plan d'un composant grâce aux méthodes :

```
setBackground(Color)
```

et :

```
setForeground(Color)
```

4 Liste

Une **liste** apparaît sous la forme d'un petite boîte dans laquelle il y a un certains nombres de mots écrits, un par ligne, appelés **items**. Un de ceux-ci est en surbrillance, celui qui est sélectionné. Lorsqu'on clique sur un de ces items à l'aide de la souris, celui-ci est sélectionné et on déclenche un événement d'item.

Les listes sont des objets de la classe :

```
List
```

du paquetage `java.awt`, dérivée de la classe `Component`.

On ajoute un item à une liste donnée grâce à la méthode :

```
add(String)
```

de la classe `List`.

Les items sont numérotés dans l'ordre dans lequel on les a ajoutés, en commençant par 0. La méthode :

```
int getSelectedIndex()
```

de la classe `List` permet de reprérer celui qui est actif.

Le traitement des événements d'item se fait grâce à l'interface :

```
ItemListener
```

dont la seule méthode est :

```
public void itemStateChanged(ItemEvent)
```

La mise en écoute d'événements d'items s'effectue grâce à la méthode :

```
void addItemListener(ItemListener)
```

de la classe `List`.

5 Compléments sur les chaînes de caractères

La méthode :

```
int length()
```

de la classe `String` permet d'obtenir la longueur d'une chaîne de caractères.

La méthode :

```
char charAt(int)
```

de la classe `String` permet d'obtenir le caractère d'index donné d'une chaîne de caractères. Les index commencent à 0.