**Université Paris Est Créteil**
**IUT de Fontainebleau**
**Département informatique**
**IS3**
**Algorithmique Programmation**
**2009–2010**

# 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.- (**Indexation automatique**)

*L'indexation des documents (c'est-à-dire terminer celui-ci par une liste de mots ou d'expressions présents dans le document dans l'ordre alphabétique avec les numéros de page [ou de ligne pour les poèmes] dans lequels ils apparaissent) est une opération usuelle. Nous allons initier une aide automatisée à l'indexation par ordinateur. Disons pour cela qu'un* texte brut *ne contient pas le caractère '*' et qu'un* texte annoté *contient un nombre pair de '*' par ligne.*

Écrire une application Java qui demande le nom d'un fichier texte, contenant un texte annoté. Ce programme créera alors deux fichiers (texte) : un fichier 'texte.txt' constitué du reprenant le contenu du fichier entré dont on a enlevé tous les caractères '*' et un fichier 'index.txt' qui donne la liste, (pas nécessairement dans l'ordre alphabétique pour commencer) de chaque chaîne de caractères entre '*' dans le texte entré, suivi de ' : ' et du numéro de la ligne dans laquelle elle se trouve.

*L'étape suivante consistant à trier les chaînes de caractères de l'index et à regrouper les lignes contenant la même chaîne de caractères ne sera pas traitée ici.*

Exercice 2.- (**Dessinons un camenbert**)

*Les ressources du budget de l'université sont les subventions accordées par le ministère, les ressources propres et la reprise d'une partie des réserves (par exemple respectivement 179,2 MEuros, 24,4 MEuros et 5,6 MEuros pour 2010).*

Écrire une applet Java qui demande les montants de ces trois ressources en millions d'euros et affiche un camenbert avec des zones respectives proportionnelles.

*On utilisera trois étiquettes et trois champs de texte pour la saisie des données. La partie du camenbert représentant les subventions sera en bleu, celle des ressources propres en blanc et celle des reporises sur réserve en rouge.*

Exercice 3.- (**Ajustement fin de la couleur de fond**)

Écrire une application Java qui affiche un cadre (fermant) dans lequel apparaissent trois étiquettes et trois ascenceurs horizontaux contrôlant les composantes (rouge, vert et bleu) de la couleur de fond du cadre.

# Documentation

## 1 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.

## 2 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
```

On peut également définir plus finement une couleur grâce au constructeur :

`Color(int pRed, int pGreen, int pBlue)`

où `pRed`, `pGreen` et `pBlue` sont compris entre 0 et 255.

On peut modifier la couleur de fond d'un composant grâce à la méthode :

`setBackground(Color)`

## 3 Dessin d'un secteur angulaire coloré

La méthode :

`setColor(Color)`

de la classe `Graphics` permet de changer la couleur de que l'on est en train de dessiner (jusqu'au prochain changement de couleur).

La méthode :

`fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)`

dessine le secteur angulaire dans la couleur en cours contenu dans le disque inscrit dans le carré défini par les coordonnées $(x, y)$ de son coin supérieur gauche, de largeur `width`, de hauteur `height`, d'angle de départ `startAngle` et d'un angle de `arcAngle`. Les quatre premiers paramètres représentent des pixels, les deux derniers des degrés (compris entre 0 et 360).

Si `height` est différent de `width`, on définit un rectangle et non plus un carré et donc un secteur d'ellipse.

# 4 Ascenseur

Un objet de la classe de widgets :

```
Scrollbar
```

du paquetage `java.awt` permet à l'utilisateur de choisir une valeur dans une gamme de valeurs entières. Le widget présente des traits de graduation et un curseur (que l'on déplace en appuyant la souris sur l'un des deux flèches situées aux extrémités.

Les cinq paramètres du constructeur :

```
Scrollbar(int orientation, int initial, int increment, int min,
int max)
```

sont l'orientation (on peut utiliser utiliser les constantes `Scrollbar.HORIZONTAL` et `Scrollbar.VERTICAL`), la valeur initiale, le pas de déplacement, la valeur minimale possible et la valeur maximale.

La méthode :

```
int getValue()
```

de cette classe permet de récupérer la valeur choisie par l'utilisateur.

Les objets de la classe `Scrollbar` générent des événements de la classe :

```
AdjustmentEvent
```

du paquetage `java.awt.event` quand l'utilisateur déplace le curseur de l'ascenseur. Pour répondre à ces changements, on doit implémenter l'interface :

```
AdjustmentListener
```

en surchargeant sa méthode :

```
public void adjustmentValueChanged(AdjustmentEvent e)
```

et mettre les sources d'événement en écoute grâce à la méthode :

```
addAdjustmentListener(AdjustmentListener)
```

de `Component`.