Université Paris-Est Créteil
IUT de Sénart-Fontainebleau
Département informatique
IS3
Algorithmique Programmation
2011–2012

Partiel
Durée : 2 heures

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.- (**Intersection de rectangles**)

Écrire une applet Java qui prend les coordonnées (x_1, y_1), (x_2, y_2) des sommets en bas à gauche et en haut à droite d'un rectangle de côtés parallèles aux bords de la fenêtre, les coordonnées (x_3, y_3), (x_4, y_4) d'un second rectangle, qui affiche ces rectangles ainsi que la plaque rectangulaires délimitée par l'intersection de ces rectangles.

Pour simplifier, on initialisera ces valeurs à 5, 100, 250, 500, 50, 50, 600, 400.

Chez vous et pour le prochain TP, améliorer l'applet précédente de façon à obtenir ces coordonnées de la part de l'utilisateur.

Exercice 2.- (**Cases à cocher**)

Écrire une application Java qui affiche un cadre (fermant) de titre « Cadre avec cases à cocher », à fond rose, dans lequel apparaissent trois cases à cocher de noms « rouge », « vert » et « bleu » tel que le fond prend l'une des huit couleurs possibles suivant les cases à cocher sélectionnées.

Ceci permet de voir la couleur obtenue par combinaison simple des trois couleurs primaires d'écran.

Exercice 3.- (**Mots utilisés dans un texte**)

Il arrive souvent que l'on ait besoin de donner la liste ordonnée des mots utilisés dans un texte, sans indiquer les doublons (on ne veut pas savoir qu'on a utilisé 22 fois le mot « le »). Une façon de faire est d'utiliser un **arbre binaire de recherche** (ABR) *dont les éléments sont des chaînes de caractères.*

Un tel ABR est une structure de données dynamiques dont les nœuds possèdent trois attributs : la valeur qui est une chaîne de caractères, le fils gauche et le fils droit qui sont des nœuds. Les méthodes sont les constructeurs et les accesseurs dont on a besoin.

- 1^o) Écrire la classe Java `Node` correspondante.

Un ABR est un arbre binaire dont le seul attribut est la racine. Ses méthodes sont :

- *un constructeur par défaut créant un arbre vide ;*

- *un constructeur créant le sous-arbre déterminé par un nœud de l'arbre ;*

- *une méthode* `insert(String)` *permettant d'insérer (éventuellement) un nouvel élément : à la racine s'il s'agit du premier élément, nul part s'il s'agit de la même valeur que la racine, dans le fils gauche de la racine si l'élément à insérer est strictement plus petit lexicographiquement que la valeur de la racine, dans le fils droit de la racine si l'élément à insérer est strictement plus grand lexicographiquement que la valeur de la racine.*

- *une méthode* `affiche()`. *La façon d'insérer rend cette méthode facile : il suffit, récursivement, d'afficher le sous-arbre gauche, la valeur de la racine puis le sous-arbre droit pour obtenir les mots dans l'ordre.*

- 2^o) Écrire cette classe Java `Tree`.

- 3^o) Écrire un programme de test insérant, un à un, les mots de la phrase « le chat du petit garçon ». On doit obtenir :

```
chat
du
garçon
le
petit
```

[*Bien entendu ce programme de test devra être amélioré, à la maison et pour le prochain TP, en demandant un fichier texte, que l'on découpera en mots (*tokens *en linguistique) en utilisant la classe* `Tokenizer`.]

Documentation

1 Les rectangles

Rappelons que la classe `Graphics` possède les méthodes :

```
drawRectangle(int x, int y, int width, int height);
```

et :

```
fillRect(int x, int y, int width, int height);
```

permettant de dessiner un rectangle et une plaque rectangulaire.

Par ailleurs le paquetage `awt` contient une classe `Rectangle`, avec les attributs publics `x`, `y`, `width` et `height`, ainsi que le constructeur :

```
Rectangle(int x, int y, int width, int height);
```

et la méthode :

```
Rectangle intersection( Rectangle);
```

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 aussi définir l'une des 16 777 216 « vraies » couleurs grâce au constructeur :

  ```
  Couleur(int r, int g, int b);
  ```

  les variables prenant une valeur comprise entre 0 et 255.

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

  ```
  setBackground(Color)
  ```

3 Cases à cocher

Une **case à cocher** prend deux valeurs (*true* ou `false`) suivant qu'elle est sélectionnée ou non. Le paquetage `awt` comprend la classe `Checkbox`, une instance de cette classe apparaissant comme un petit carré pécédé d'un nom, avec une croix dans le carré lorsqu'elle est sélectionnée (en cliquant dessus avec la souris). Son constructeur le plus courant est :

```
Checkbox( String);
```

permettant de lui attribuer un nom. La méthode :

```
public boolean getState();
```

permettant d'en connaître l'état.

Pour pouvoir détecter le changement d'état d'une boîte à cocher, il faut implémenter l'interface `ItemListener` en surchargeant sa méthode :

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

4 Les chaînes de caractères

Rappelons que la méthode de `String` :

```
public int compareTo(String );
```

renvoie un entier strictement négatif si le mot objet est strictement plus petit (dans l'ordre lexicographique) que le mot argument, nul s'ils sont égaux et strictement positif dans le cas de strictement plus grand.