Université Paris XII
IUT de Fontainebleau
Département informatique
FI-2
2005-2006

PARTIEL 2

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

Les deux parties sont indépendantes mais elles doivent être présentés dans l'ordre (prévoir deux pages par exercice).

Le but est de réaliser un serveur et un client permettant de transmettre des fichiers texte sur le réseau informatique.

Exercice 1.- (**Serveur**)

Dans le répertoire de l'application serveur se trouve un sous-répertoire de nom "`images`*" qui contient un fichier "*`liste.txt`*" (donnant la liste des fichiers transmissibles, un fichier par ligne) ainsi que les fichiers transmissibles par le serveur.*

Écrire une application Java serveur (sur le port 8888) qui reçoit une ligne de caractères d'un client :

- "`QUIT`" pour terminer la connexion.
- "`LIST`" pour envoyer la liste des fichiers transmissibles.
- tout autre ligne de caractères est considérée comme nom de fichier à transmettre au client.

[*Pour simplifier, on ne vérifiera pas que le nom du fichier demandé par le client apparaît bien dans la liste (d'où risque d'exception et de serveur très peu stable). On pourra améliorer ce point à la maison.*

Lors de la transmission d'un fichier via *le réseau, il faut décider d'une valeur signal pour indiquer la fin de celui-ci. On choisira la convention d'une ligne ne contenant qu'un seul caractère, un point '.'.*]

Exercice 2.- (**Client graphique**)

Écrire l'application Java client qui affiche un cadre fermant dans lequel apparaît (voir figure 1) :

- une étiquette "`Serveur`", un champ de texte permettant de spécifier l'adresse IP du serveur (pré-rempli avec "`localhost`"), une étiquette "`Port`", un champ de texte permettant de spécifier le port (pré-rempli avec `8888`)[1], un bouton (intitulé "connect") (qui permet de se connecter lorsque les deux champs de texte précédentes ont été renseignées) ;

[1] Le client pourra donc être utilisé avec un autre serveur que celui décrit à l'exercice 1.

- un bouton “`Liste`” (qui permet de demander la liste après s’être connecté), une zone de texte dans laquelle sera reçue la liste ;

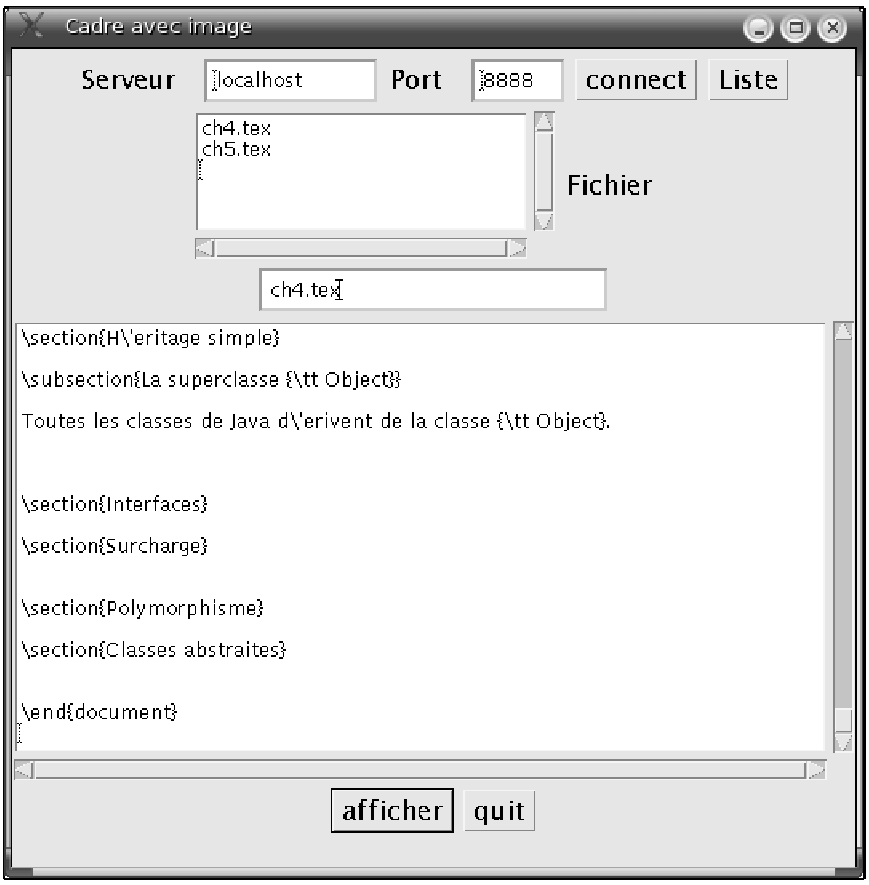

Figure 1: Aspect du client

- une étiquette “`Fichier`”, un champ de texte permettant de spécifier le fichier à recevoir, un bouton “`afficher`” (pour demander à rapatrier ce texte et à l’afficher), une zone de texte pour afficher ce fichier ;
- et enfin un bouton “`quit`” pour clore la connexion.

[*Comme d’habitude en Java, on ne cherchera pas à bien placer les widgets les uns par rapport aux autres.*]

Documentation

Zone de texte

La classe `TextField` est un composant qui permet de saisir une ligne de texte. La classe :

`TextArea`

du paquetage `java.awt`, permet de saisir plusieurs lignes.

Le constructeur le plus complet est :

```
public TextArea(String s, int rows, int columns, int scrollbars)
```

qui spécifie le texte `s` à placer lors de l'initialisation, le nombre de ligne et de colonnes et la présence ou non d'ascenseurs. Pour ce dernier argument, on peut utiliser l'une des constantes statiques `SCROLLBARS_BOTH`, `SCROLLBARS_HORIZONTAL_ONLY`, `SCROLLBARS_VERTICAL_ONLY` ou `SCROLLBARS_NONE` de cette classe.

On peut utiliser les deux méthodes :

`String getText()`
`setText(String)`
`append(String)`

(d'une sur-classe) de cette classe pour récupérer, positionner et ajouter du texte.