ESIEE AMIENS
JAVA I3

Projet guidé JAVA
(partie 1 – projet préliminaire)

Nicolas DAILLY

Contexte des mini-projets

Dans le cadre des cours de JAVA, nous vous proposons la réalisation de mini projets guidés, qui vont vous permettre de vous familiariser avec le langage et de découvrir les possibilités offertes par certaines classes standard du langage.

Ces mini-projets s'enchaînent de façon à ce que vous puissiez ré-exploiter les connaissances acquises et les classes développées au fil de différents mini-projets. C'est pourquoi, avant de commencer un nouveau mini-projet, vous devez avoir validé le mini-projet précédent.

Projet préliminaire

Ce premier projet (préliminaire) va vous permettre d'étudier les entrées sorties en JAVA. Avec ce projet, vous allez découvrir comment écrire un texte à l'écran, lire un texte saisi au clavier, puis écrire et lire dans des répertoires et fichiers.

Développement en JAVA

Java est un langage et un ensemble d'outils de développement, développé initialement par Sun Microystems, et rachetés par Oracle. Le code produit en java est un pseudo-code, interprété par une machine virtuelle, et donc exécutable par toute plate forme embarquant une machine virtuelle Java.

Java est librement téléchargeable sur le site d'Oracle. Toutes les fonctions sont documentées dans la Javadoc http://docs.oracle.com/javase/7/docs/api/ (que vous devez apprendre à utiliser et que vous devez garder à portée de main tout au long de ces Tps). Mettez le lien en marque page dans votre navigateur.

Pour développer en Java :

- Créez vos classes, et enregistrez les dans des fichiers portant le nom de vos classes, avec l'extension « .java »
- Compilez votre programme avec la commande « javac ». Cela génére des fichiers « .class », qui sont le pseudo-code interprétés par la machine virtuelle.
- Exécutez votre programme en utilisant la commande « java ».

Pour vérifier votre installation :

- En ligne de commande, tapez « java -version »
  - Vous verrez la version de java installée sur vos machines
- En ligne de commande, tapez « javac -version »
  - Vous verrez la version du compilateur utilisée
- Créez une classe « HelloWorld » qui vous permet d'afficher le texte « HelloWorld » à l'écran (compilez et exécutez la).

Notes :

- Pour faciliter la réalisation de ce TP, veuillez respecter le nom des classes qui vous sont imposées.
- La qualité de votre code (respect des convention d'écriture Java, indentation du code) sera prise en compte dans l'évaluation.

 # Conventions d'écriture JAVA :

- Les constantes sont écrites en majuscule (exemple : static final CONSTANTE=12)
- Les variables et attributs sont écrits en minuscules. Le premier mot commence par une minuscule. Si leur nom est composé de plusieurs mots, les mots suivants commencent par une majuscule. (exemple : variableAPlusieursMots)
- Les classes commencent sont écrites en minuscule, mais chaque mot de la classe commence par une majuscule. (exemple ClasseAPlusieursMots)

 # Lecture de la Javadoc

La Javadoc se présente sous forme d'une page à trois fenêtres :

- Les packages dans la fenêtre en haut à gauche
- Les classes dans la fenêtre en bas à gauche
- Le détail de la classe dans la fenêtre de droite

Le détail de la classe comporte :

- Le nom du package auquel la classe appartient (à importer si autre que java.lang)
- L'arborescence d'héritage (la classe peut dérive forcément de la classe Object)
- Un descriptif texte sur la classe (comporte parfois des exemples d'utilisation)
- La liste des attributs de la classe (Field Summary)
  - Chaque attribut est précédé de son type
- La liste des méthodes de la classe (Method Summary)

    - Chaque méthode est précédé de son type de retour et suivi du type d'objet qu'elle prend en argument
- En cliquant sur le nom des attributs et méthodes, on accède à leur descriptif détaillé.

Etape 1 : Entrées/Sorties Ecran

Pour les besoins de ce TP, vous allez devoir manipuler à la fois les entrées/sorties écran, les entrées/sorties fichier et les entrées/sorties réseau.

Entrée/sortie Ecran

Analyse : Pour afficher un texte à l'écran, vous devez utiliser la commande :

```
System.out.println("text")
```

Recherchez dans la Javadoc quel est le type de « System.out ». Est-ce une méthode ou un attribut de la classe ? Quelle est la différence entre « System.out » et « System.err » ?

Pour lire un texte tapé par l'utilisateur à l'écran, vous devrez utiliser l'objet « System.in ».

- Quel est le type de cet objet ?
- De quelles classe dérive t'il ?
- Y a t'il une fonction pour lire directement une ligne tapée à l'écran ?

Pour lire une ligne, vous pouvez utiliser la classe BufferedReader (méthode « readLine() »). Pour construire un objet BufferedReader à partir de l'objet « System.in », vous aurez besoin de créer un objet de type « InputStreamReader ».

A faire :

- Créez une classe IOCommandes (IO=Input/Output)
- Définir un attribut privé lectureEcran de type « BufferedReader » et un attribut privé ecritureEcran de type « PrintStream »
- Créer un constructeur qui initialise lectureEcran et ecritureEcran
- Créez une méthode de type « public void ecrireEcran(String texte) » qui prend un texte en argument et l'affiche à l'écran.
- Créer une fonction de type « public String lireEcran() » qui renvoie le texte tapé par l'utilisateur à l'écran.

- Créez une classe « Principale », qui ne contient qu'une fonction « main », crée un objet IOCommandes, et l'utilise pour lire un texte à l'écran et en faire un écho à l'écran.
- Adaptez ensuite ce programme de façon à ce que le programme demande la saisie d'un texte tant que l'utilisateur ne tape pas « quit ».

Rappel : En ligne de commande, pour vous déplacer dans les répertoires, tapez cd suivi du nom du répertoire, « cd .. » pour remonter dans l'arborescence, ls pour lister les fichiers du répertoire.

Pour compiler votre programme, il suffit de taper, en ligne de commande :

```
javac Principale.java
```

S'il n'y a pas d'erreurs, cela va produire un fichier IOCommandes.class et un fichier Principale.class. Pour lancer votre programme, tapez en ligne de commande :

```
java Principale
```

#Récupération d'erreurs

Certaines actions peuvent générer des erreurs. Une erreur classique et « IndexOfBoundException » quand vous essayez de lire dans un tableau un index qui n'existe pas. Lire et écrire dans un Socket qui n'est pas ouvert ou qui s'est fermé prématurément peut conduire également à des erreurs d'entrées/sorties « IOException ».

Quand une fonction peut déclencher une exception, il faut le déclarer en ajoutant le mot clef « throws », suivi du type de l'exception, à la suite de la déclaration de la fonction.

Il est également possible de récupérer l'exception, pour la traiter. On entoure alors le code susceptible de déclancher l'exception à l'aide d'un bloc try {<code>} catch(<typeDException> e) {<code de traitement>}. Ne jamais laisser le bloc de traitement vide. Quand on ne sait pas quel traitement faire, mettre un code pour tracer l'exception : e.printStackTrace() ;

Etape 2 : Entrée/sortie Réseau

La classe « Socket » permet de se connecter à un serveur distant.

Etudiez la documentation java de la classe socket pour trouver les fonctions équivalentes aux objets « System.in » et « System.out ».

Modifiez la classe IOCommandes :

- Affectez lui un attribut privé de type Socket
- Modifiez le constructeur pour qu'il prenne un Socket en argument
- Créez une méthode de type « public void ecrireReseau(String texte) » qui prend un

texte en argument et l'écrit sur le réseau.

- Créez une fonction de type « public String lireReseau() » qui renvoie une ligne envoyé par le serveur sur le réseau.

Modifiez votre classe « Principale » pour qu'elle se connecte au serveur sur le port 5999. Sur ce port tourne un service qui fait un echo de ce qu'on lui envoie.

Implémentez alors le programme suivant : jusqu'à ce que l'utilisateur tape la commande « quit »

- lisez ce que l'utilisateur tape à tape à l'écran
- envoyez le contenu au serveur
- lisez la réponse du serveur
- affichez la réponse du serveur à l'écran

Vous avez développé votre premier client. Passé cette étape, vous avez tous les outils pour développer votre client chat.

Etape 3 : Entrée/sortie Repertoire/Fichier

Sur le même modèle que précédemment, développez des méthodes permettant

- D'ouvrir un fichier et d'écrire dedans

- D'ouvrir un fichier et de lire son contenu (ligne à ligne)

Pour cela, vous étudiez la Javadoc des classes File et FileReader.

Développez un petit programme qui permet de lister les fichier d'un répertoire, de naviguer dans ces répertoires et d'afficher le contenu texte d'un fichier.

Exemple :

```
Répertoire courant : /toto/
Contenu :
.
..
rep. titi
rep. tata
file. toto.txt

Tapez votre commande (view <fichier> ou enter <rep>) :
```