Premiers pas en Java

Septembre 2005

Objectif

Les objectifs de ce premier TP sont la mise en place de l'environnement de travail en java et l'étude des fondements du langage. C'est-à-dire d'une part la compréhension de la compilation et de l'exécution de code java, l'arborescente classique d'une application et la génération de la documentation et d'autre part la maitrise de la création de classes et d'objets.

En cas de besoin, une version des logiciels se trouve dans le répertoire `/home/profs/bruno/JAVASOFTS/`

1 Une première exécution

1.1 Mettre en place l'environnement

□ La première chose à faire est de fixer quelle distribution du jdk va être utilisée, pour cela fixer la valeur de la variable d'environnement JAVA_HOME pour indiquer son emplacement (`/usr/java/j2sdk1.4.2_05/`).
□ Ensuite, on ajoute le répertoire `$JAVA_HOME/bin` à la valeur de la variable `PATH` pour que les commandes de bases soient accessibles.
□ Vérifier que cela fonctione en exécutant les commandes `java -version` et `javac -help`

Il est impératif que **toutes** les documentations suivantes soient ouvertes dans votre navigateur web :

- la documentation sur les API :
  `http://java.sun.com/j2se/1.4.2/docs/api/index.html`
- la documentation sur les outils Java fournis par Sun :
  `http://java.sun.com/j2se/1.4.2/docs/tooldocs/tools.html#basic`

1.2 Compiler et exécuter une application

□ A la racine de votre compte ajouter un répertoire `SSI3` (ce répertoire sera appelé *répertoire de travail*). A l'intérieur de votre répertoire de travail ajouter le répertoire `TP` (appelé répertoire de *projet*). A l'inté
répertoires standards `src`, `build`, `doc`, `dis`
□ Dans le répertoire `src`, créer l'arboresce
`fr.univtln.`*`login`*`.tp.tp1`. A l'intérieur d
vous trouverez à l'adresse suivante `http`
`M1/SSI3/TP/TP1/PremierProgramme.java`
□ Ouvrir et modifier ce programme pour q
□ `A partir de votre répertoire de p`
mande suivante :

```
javac -sourcepath src -d build
src/fr/univtln/login/tp/tp1/Premier
```

□ Regarder dans le répertoire `build`. Pour
de projet :

```
java -classpath build
fr.univtln.login.tp.tp1.PremierProgr
```

1.3 Générer la documentation

1.3.1 Précision sur la forme

La commande `javadoc` produit de la docu
uliers insérés dans le code source des classe
paquetages, classes ou interfaces, variables

Les commentaires peuvent contenir du
forme de texte (¡I¿ italique, ¡B¿ caractère gr
inclure du code dans les commentaires. Des
le caractère @ (@author, @version, @para
juste avant ce qu'ils commentent.
□ Lire cette page : `http://java.sun.com/`
`html`.

De plus, il est rappelé que le langage J
tiques. Cette page présente les habitudes d

□ Lire cette page : `http://g.oswego.edu/dl/html/javaCodingStd.html`.

1.3.2 Génération

A partir de maintenant, **tous vos programmes seront commentés**.
□ Génerer la documentation depuis votre répertoire de projet avec la commande :
`javadoc fr.univtln.tp.tp1 src/fr/univtln/bruno/tp/tp1/PremierProgramme.java -d doc`

1.4 Générer et exécuter l'archive jar

□ Génerer l'archive de votre projet depuis votre répertoire de projet avec la commande :
`cd build; jar cvf ../dist/tp.jar fr; cd ..`
□ Exécuter le programme en ajoutant le jar au classpath et en indiquant la classe exécutable avec la commande suivante :
`java -classpath dist/tp.jar fr.univtln.`*login*`.tp.tp1.PremierProgramme Pierre`
Il est possible de rendre le jar " exécutable " en créant un fichier `manifest` qui indique en particulier quelle est la classe exécutable.
□ Copier le fichier `http://sis.univ-tln.fr/~bruno/Enseignement/M1/SSI3/TP/TP1/monManifest` dans le répertoire de projet et regarder son contenu.
Recréer le fichier `.jar` en précisant le manifeste à ajouter :
□ cd build; jar cvfm ../dist/tp.jar ../monManifest fr; cd ..
Puis exécuter directement l'archive :
□ `java -jar dist/tp.jar Pierre`

1.5 Automatiser le processus avec Apache Ant

Comme vous l'avez vu la compilation d'un projet complet en Java est une tâche fastidieuse. C'est pourquoi, il est conseillé d'utiliser un outil qui automatise ce travail : Ant (`http://ant.apache.org/`). Ant utilise un fichier de configuration qui indique les tâches à accomplir, vous trouverez un exemple de ce fichier à l'adresse `http://sis.univ-tln.fr/~bruno/Enseignement/M1/SSI3/TP/TP1/build.xml`.
□ Copier ce fichier dans votre répertoire de projet et regarder son contenu. Adapter son contenu pour qu'il indique des chemins corrects.
□ Créer une variable d'environnement `ANT_HOME` qui indique le répertoire d'installation de ant (`/usr/local/apache-ant-1.6.1/`). Ajouter `ANT_HOME/bin` dans le `PATH`.
Pour obtenir l'ensemble des tâches possibles avec un fichier de construction :
□ `ant -projecthelp build.xml`
Pour lancer l'exécution et si nécessaire compiler et générer l'archive :
□ `ant run`

1.6 Utilisation de l'environnem

Nous allons maintenant utiliser l'environne
`eclipse.org`). Comme " workspace " vous
que eclipse stocke ses informations de conf
□ Créer un nouveau projet à partir d'un fi
□ Modifier les propriétés du nouveau projet
indiquer la compatibilité avec Java5. Da
répertoires des sources et des binaires sont
□ Dans les propriétés du paquetage `JRE_LI`
□ Pour lancer une tâche ant, utiliser le me
□ Vous pouvez maintenant éditer vos prog

2 Les concepts de base d

2.1 Création et instanciation d

Créer une classe Java exécutable nommée `T`
`Test` devra afficher `Hello`.
Créer une classe `Personne` **sans cons**
ayant un nom, un prénom, un age et un sa
attributs. Vous vérifierez qu'un salaire ne
`p1` dans la classe `Test`, mettre à jour ses
`agé de 30 ans et gagne 2000e`).
Ajouter des constructeurs dans la classe
éventuellement l'age et la profession. Mod
ne puisse plus être modifiés après l'instanc
Ajouter un attribut qui indique l'année
la méthode que retourne l'age. L'année cou
`java.util.Calendar.getInstance().get`
Ajouter une méthode `comparerSalair`
retourne -1, 0 ou 1 selon que celle à un s
courante. Créer une deuxième personne `p`
plus. Afficher le résultat de la comparaison
Ajouter une méthode `comparerSalair`
qui retourne -1, 0 ou 1 selon de le salaire d
celui de la seconde.
Attention `comparerSalaire(Personn`
sont-elle des méthode de classe ou d
`totalDesSalaires` qui indique le total des
fier les méthodes nécessaires pour le salaire
Afficher le salaire total après la créatio

2.2 Les méthodes de base

2.2.1 toString()

Afficher directement l'objet `p1` dans la classe `Test`. Ajouter une méthode : `String toString()` à la classe personne qui retourne une chaine de caractère qui décrit la personne. Exécuter `Test`.

2.2.2 equals()

On souhaite considérer que deux personnes sont " égales " dans notre application, si elles sont nées la même année. Pour définir une relation d'égalite d'objets en Java on redéfinit la méthode `Boolean equals(Object)`. Vérifier que `p1` et `p2` sont égales.

2.3 Définition de classes locales

Pour représenter le fait qu'une Personne à un cerveau ajouter une définition de la Classe Cerveau **à l'intérieur** de la définition de la classe Personne. Ajouter un attribut cerveau à personne et instancier un cerveau à chaque personne lors de sa création. Compiler et exécuter `Test`. Regarder dans le répertoire `build` les classes qui ont été compilées. Essayer de créer une instance de Cerveau diretement dans la classe Test.

3 Pour finir...

Vous allez maintenant créer une classe `Entreprise` qui permet de réprésenter des entreprises qui comportent au maximun Personnes `MAX_EMPLOYES`.

Vous ajouterez les méthodes pour créer une entreprise, ajouter des employés et afficher le nombre d'employé, un employé précis et tous les employés (en utilisant le foreach de Java5).

Les entreprises doivent appartenir à l'une des trois catégories suivantes: {PUBLIQUE, PME, GRAND_GROUPE}. Modifier la classe `Entreprise` pour représenter cela en utilisant un type enuméré et ajouter trois méthodes `isPUBLIQUE()`, `isPME()`, `isGRAND_GROUPE()` qui retournent un booléen.

Ajouter une méthode `miseEnForme()` qui passe le nom de toute les personnes d'une entreprise en majuscule.

Modifier votre classe `Test`, pour l'on puisse lui passer en paramètre le nom et le type d'une entreprise, puis la liste des personnes qui la compose.

```
java Test maBoite PME Pierre Durand 1950 2000 Paul Dupond 1960 3000 Marie Martin 1975 2500
```