

RICM1

TP de Programmation Orientée Objet n°1

Introduction aux outils du SDK

Dernière modification : le 18 janvier 2008
(Philippe.Morat@imag.fr)

pour en savoir plus sur : <http://java.sun.com/j2se/1.5.0/>

Un programme Java peut prendre deux formes distinctes, chacune étant adaptée à un contexte d'invocation et d'exécution différents.

- La première (applications java) permet de créer des applications au sens classique du terme, c'est à dire des programmes s'exécutant de manière autonome à l'aide d'un interpréteur Java.
- La deuxième (applets Java) est destinée à des programmes invoqués depuis des documents HTML (HyperText Markup Language) diffusés sur le World Wide Web (WWW) et exécutés à l'intérieur d'un navigateur (Netscape, Microsoft Internet Explorer) ou d'un visualiseur d'applets équipé d'un interpréteur Java (par exemple l'application `appletviewer` du SDK).

Les exercices qui vous sont proposés ici ont pour but de vous faire expérimenter ces deux types de programmes Java et de vous faire utiliser les outils de base du SDK :

- le compilateur (`javac`),
- l'interpréteur Java (`java`) pour les applications "stand-alone",
- `appletviewer` pour l'exécution d'applets
- `jar` pour créer des archives de fichiers de byte-code et de ressources
- `javadoc` pour générer de la documentation html à partir des commentaires documentants se trouvant dans le code source des classes.

exercice 0 : consulter la documentation en ligne pour le cours AOO

Avant de commencer le TP survoler cette documentation en quelques minutes, **savoir chercher dans celle-ci vous sera indispensable pour la suite.**

A titre d'exercice vous pouvez essayer de retrouver :

- la documentation de la classe `Math` de l'API java, classe définie dans le package `java.lang`

- la documentation de la classe `Vector`
 - la documentation du compilateur `javac`
 - le tutorial sur les fonctionnalités (features) du langage concernant les structures de données (Collections)
-

exercice 1 : compiler et exécuter une application java

pour en savoir plus sur : le compilateur [javac](#) et l'interpréteur [java](#)

- récupérez le programme source [HelloWorldApp.java](#) (pour sauvegarder le texte de ce programme dans un fichier cliquez sur le lien en maintenant le bouton droit de la souris enfoncé et sélectionnez l'option `Save As` du menu qui apparaît),
- compilez et exécutez l'application `HelloWorldApp.java`,
- modifiez ce programme de manière à ce qu'il provoque l'affichage suivant :

```
BONJOUR  
CECI EST MON  
PREMIER PROGRAMME JAVA
```

ou tout autre texte de votre choix.....

exercice 2 : Génération de documentation HTML de classes, écriture d'une application simple.

pour en savoir plus sur :

le générateur de documentation [javadoc](#)

le [class path](#) : pour définir les règles de recherche des byte-codes des classes java.

la construction de fichiers [jar](#) (fichiers Java ARchives)

1. Créer un répertoire `Visages` avec deux sous répertoires `src` et `classes`
2. Sauvegardez dans le répertoire `src` les fichiers suivants : [VisageRond.java](#), [Dessin.java](#) et [AppliVisage1.java](#)
3. Placez vous dans le répertoire `src` et compilez l'application `AppliVisage1` en utilisant la commande `javac -d ../classes AppliVisage1.java` pour placer les fichiers de byte-code dans le répertoire `classes`.
4. Placez-vous dans le répertoire `classes` et constatez que l'ensemble des classes nécessaires a été compilé.
5. Depuis ce répertoire `classes` exécutez l'application `AppliVisage1`. Faire CTRL C sur la ligne de commandes pour arrêter l'application.
6. Replacez vous dans le répertoire `Visages` et exécutez l'application `AppliVisage1` en utilisant l'option `-classpath` pour indiquer à l'interpréteur `java` où trouver les classes à charger.
7. Déplacer le fichier `AppliVisage1.class` du répertoire `classes` vers le répertoire `Visages` (commande `move` sous MS-DOS).
8. Essayez de réexécuter l'application `AppliVisage1` par la commande `java AppliVisage1`, constatez.
9. Replacez vous dans le répertoire `classes` et construisez un fichier `Visages.jar` contenant les fichiers byte-code des classes `Dessin` et `VisageRond`. Fichier `jar` que vous placerez dans le répertoire `visages`.
10. Replacez vous dans le répertoire `Visages` et lancez à nouveau l'exécution en utilisant ce fichier `jar` dans l'option `-classpath` de la commande `java`.
11. Pour comprendre le fonctionnement de cette application étudiez la documentation HTML des classes `VisageRond` et `Dessin` que vous générerez à partir des commentaires documentants situés dans les sources de ces classes (regardez ces sources).

- a. créez un répertoire docs dans le répertoire Visages
 - b. depuis le répertoire src lancez la commande `javadoc -d ../docs VisageRond.java Dessin.java`. Les fichiers html de documentation sont générés dans le repertoire docs, le point d'entrée étant le fichier `index.html`.
 - c. consultez cette documentation à l'aide d'un navigateur. La documentation de la classe VisageRond référence un image que vous trouverez ici : [figurevisage.gif](#). Pour que la documentation soit complète recopiez cette image dans le répertoire docs.
12. Modifiez la classe VisageRond afin que la taille par défaut d'un visage soit de 100x100 au lieu de 50x50 et recompilez cette classe.
 13. Réexécutez l'application AppliVisage1 en utilisant la même commande qu'au point 10, constatez.
 14. Réexécutez l'application en application la commande du point 4, constatez.
 15. Sauvegardez dans le répertoire src le fichier : [AppliVisage2.java](#)
 16. Placez vous dans le répertoire src et compilez AppliVisage2 avec la commande `javac -classpath ../classes -d .. AppliVisage2.java`, corrigez les erreurs de compilation.
 17. Exécutez l'application AppliVisage2.
 18. Sauvegardez dans le répertoire src le fichier : [AppliVisage3.java](#)
 19. Placez vous dans le répertoire src et compilez AppliVisage3 avec la commande `javac -classpath ../classes -d .. AppliVisage3.java`
 20. Exécutez l'application AppliVisage3 et corrigez.

Exercice 3 : compiler et exécuter une applet java

pour en savoir plus sur : le visualiseur d'applets [appletviewer](#)

L'affichage du texte ci-dessous dont les lettres se déplacent de façon aléatoire est en fait géré par une applet JAVA (écrite par Daniel Wyszynski, Branché JAVA, A.van Hoff,S. Shaio, O. Starbuck, Ed. International Thomsom Publishing France 1996)



Note that you can make it say anything! This text is flowing around the applet because it is left aligned. You can also right align it if you want it to appear on the other side.



- sauvegardez sur votre compte [NervousText.java](#) le source JAVA de cette applet,
- sauvegardez sur votre compte [essaiapplet.html](#) un exemple de fichier HTML qui incorpore cette applet,
- compilez la classe `NervousText.java` que vous avez sauvegardé sur votre compte,
- exécutez cette applet
 1. en chargeant le fichier `essaiapplet.html` dans votre navigateur Web
 2. en lançant depuis la ligne de commande la commande `appletviewer essaiapplet.html`
- modifiez `NervousText.java` et/ou `essaiapplet.html` de manière à afficher le texte suivant :
"VIVEMENT QUE NOUS PARTIONS AU SKI !" (ou tout autre texte de votre choix)

Exercice 4 : Si vous voulez voir d'autres démonstrations de programmes et d'applets Java :
<http://java.sun.com/j2se/1.5.0/docs/relnotes/demos.html>