
TP de Segmentation - Analyse d'Image - M2 THCS

Dominique Fourer
fourer@labri.fr

5 octobre 2009

1 Présentation d'ImageJ

ImageJ est une application développée en JAVA permettant d'effectuer de l'analyse et du traitement d'image. Cette application servira de base pour tester les algorithmes qui seront implémentés en cours de travaux dirigés.

ImageJ peut être directement téléchargé à l'adresse : <http://rsbweb.nih.gov/ij/download/zips/ij142.zip>

1.1 Installation et utilisation

Décompressez l'archive *ij142.zip* (avec la commande `unzip`). L'application peut être exécutée par la commande

```
java -cp <chemin_ImageJ>/ij.jar ij.ImageJ
```

Cette commande permet d'indiquer à java d'exécuter la classe *ImageJ* située dans le package *ij.jar*. Si tout se passe bien, une fenêtre graphique devrait apparaître : l'application est lancée.

1.2 Utilisation des plugins

Lors de son exécution, *ImageJ* recherche tous les plugins disponibles situés dans le dossier "plugins", il est possible d'indiquer à *ImageJ* un dossier différent lors de son exécution si nécessaire en ajoutant

```
-Dplugins.dir=<nouveau_dossier_des_plugins>
```

lors du lancement de la commande *java* (cf. feuille de TD 1).

2 Quelques notions de JAVA

2.1 Bref aperçu du langage

Le langage JAVA est un langage orienté objet. Un programme JAVA s'exécute sur une machine virtuelle (JVM¹) qui lui permet de fonctionner sur n'importe quelle plateforme (Windows, Linux, Mac).

Les programmes JAVA sont structurés en Package (Dossier) qui peuvent eux-même inclure plusieurs classes.

Les classes peuvent modéliser des objets et comporter un certain nombre de propriétés (variables) et des méthodes (fonctions).

La structure d'un projet java est de la forme :

```
\<Dossier parent>\
    \src\Main.java
        \Package1\classe11.java
            \classe12.java
        \Package2\classe21.java
    \lib\lib.jar
        \lib2.jar
    \ressource\image1.pgm
        \image2.pgm
```

Le dossier *src* correspond au package *default*. Une classe JAVA est exécutable lorsqu'elle comporte une méthode *main* (similaire au langage C) :

```
public static void main(String[] args)
```

2.2 Développement d'application JAVA

Les fichiers *.java* peuvent être édités à partir de n'importe quel éditeur de texte (emacs, vi, kwrite,...) cependant il est recommandé d'utiliser *eclipse* qui est une application libre conçue initialement pour JAVA et incluant un compilateur et un débogueur. Il existe aussi de nombreux plugins pour *eclipse* qui lui permettent de travailler sur d'autres langages (C/C++, Python, Latex, PHP, ...).

La compilation d'un fichier *.java* donne un fichier *.class* et se fait par la commande *javac*

```
javac fichier.java -classpath \lib_path;\lib\bibliotheque.jar
```

La partie *-classpath ...* permet d'indiquer au compilateur le chemin où celui-ci doit aller chercher les bibliothèques contenant les fonctions externes utilisées.

Pour utiliser une fonction implémentée dans une archive *.jar* ou un fichier *.class*, la fonction doit comporter la ligne

```
import package1.package2.classe_a_inclure
```

¹Java Virtual Machine

2.2.1 Quelques règles importantes

Une seule classe par fichier *.java*. Le nom du fichier est le même que celui de la classe implémentée. L'arborescence du projet définit directement le chemin des *import* en début de fichier.

Même chose pour les packages : le package doit respecter le nom du dossier correspondant. En général *eclipse* affiche une erreur si la convention n'est pas respectée.