

TP1 : Installation des outils pour le GL

Objectif du TP

L'objectif de ce TP est d'installer les outils qui vous seront utiles pour la suite des TP ou pour d'autres modules de la formation (Développement Android, Gestion de projets, Programmation Orientée Objets...) et de s'initier à l'utilisation d'Eclipse.

Pour l'ensemble des TP de ce module, il est demandé que chaque étudiant envoie à <mailto:denis.pallez@unice.fr> le travail effectué avec le sujet [LPSIL_DAM] et contenant un fichier attaché au format zip nommé <Nom étudiant>_TP<N° du TP>_<date du TP en anglais>.zip. Exemple de nom de fichier `Pallez_TP1_20130909.zip`. *Tout manquement à cette règle entrainera la note de 0 pour le TP.*

UN LANGAGE DE DÉVELOPPEMENT

1. Installez un kit de développement standard de Java à partir du site [d'Oracle](#) ou à partir du fichier disponible dans le répertoire `SupportCours`.

GESTIONNAIRE DE VERSIONS (SVN ET GIT)

2. Installez TortoiseSVN sous windows pour gérer le versionning de fichiers (prochain TP) à partir de leur [site](#) ou à partir du fichier disponible dans le répertoire `SupportCours`.

ENVIRONNEMENT DE DÉVELOPPEMENT

3. Installer Eclipse à partir du site de la fondation [Eclipse](#) ou à partir du fichier disponible dans le répertoire `SupportCours`. Un très bon tutoriel d'installation et de configuration est disponible [ici](#) (§III et IV).
4. Sachez qu'il est possible d'utiliser Eclipse pour développer en C/C++, Python, PHP, Ruby, Perl, Flex(Flash). Installer le package CDT qui permet de développer en C/C++. Pour cela :
 - a. Aller dans le menu `Help > Install New Softwares> Add` ;
 - b. Saisissez CDT ou C/C++ et l'adresse <http://download.eclipse.org/tools/cdt/releases/kepler> et cliquez sur `Select All` et `Next` ;
5. Faites de même que précédemment avec le plugin `Subclipse` qui se trouve à l'adresse http://subclipse.tigris.org/update_1.8.x.
6. Installer le package ADT qui permet de développer des applications Android. Pour cela, suivez ce [tutoriel](#).
7. Configuration d'Eclipse
 - a. Lisez et appliquez le précédent tutoriel disponible [ici](#).
 - b. Vérifier la version du compilateur Java d'Eclipse.
 - c. Créer sa propre version de formatage du code en remplaçant les tabulations par seulement 2 espaces.
 - d. Un des problèmes usuels de l'informatique est le codage des caractères (c'est le cas quand vous voyez fleurir les `Ã©` et `Ãª` et autres dans les pages web). En particulier, windows, Unix et MacOS n'utilisent pas toujours le même (en fait jamais), et recopier des programmes d'une machine à l'autre peut poser des problèmes de conversion. Tant qu'il s'agit de programme, le problème n'apparaît pas

puisque les codages sont souvent identiques pour les caractères non accentués. Le problème apparaît en général quand vous ajoutez des accents dans les chaînes de caractères. Une des solutions est de forcer Eclipse à utiliser le même codage pour toutes les plateformes. Cela se fait en décochant `default` et en cochant `other`, puis en choisissant le codage (même si c'est le même par défaut, car le codage du système peut changer sans prévenir). Le codage désormais standard international est l'UTF-8 (qui permet d'avoir tous les caractères, comme κ , μ , \acute{o} , \acute{u} ou ϵ), mais vous pouvez aussi choisir ISO-8859-1 (appelé aussi latin1) ou ISO-8859-15 (appelé aussi latin9, qui permet, par rapport au latin1, d'avoir œ Œ et €).

8. Création d'un projet :

- Créez un projet `test` dans le `workspace` par défaut. Les sources de ce projet seront dans un répertoire `src` et les classes compilées dans un répertoire `classes`.
- Si elles ne sont pas disponibles, ajoutez les vues `Navigateur` et `Explorateur de fichiers` et liez les à l'éditeur.
- Quelles options sont disponibles pour ces vues grâce au menu contextuel?
- Testez différents placements pour ces vues.

9. Quelques classes :

- Créer une classe `Point` contenant 2 entiers ; écrivez uniquement la déclaration de la classe et des 2 entiers. À l'aide d'Eclipse (menu `Source`), générez automatiquement un constructeur, des getters et des setters et la méthode `toString()` sans taper une ligne de code !
- Créez un package `test` dans votre projet.
- Ajoutez une classe `Test` contenant une méthode `main`.
- Générez (sans l'écrire) un constructeur sans paramètre, public, avec des commentaires utilisables par la javadoc.
- Cherchez dans les préférences un moyen de modifier ces commentaires; ajoutez-y votre nom.
- Ajoutez à la classe `Test` un champ `nom` de type `String` et un champ `valeur` de type `int` avec une visibilité protégée.
- Générez les méthodes `get/set` (getter/setter) pour ces champs.
- Générez un constructeur à 2 paramètres qui initialise ces champs.
- Créez une méthode `echo()` provoquant un affichage ressemblant à "Test: mon nom est ... et ma valeur est ...".
- Créez une classe `SousTest`, qui hérite de `Test`, et contenant les constructeurs de son parent.
- En utilisant la vue hiérarchique, trouvez toutes les méthodes héritées par `SousTest`.
- Remplacez la méthode `echo()` du parent par une version identique modifiant juste "Test" en "SousTest".
- Créez dans `SousTest` une méthode `test()` qui appelle la méthode `echo()` du parent. Vérifiez que c'est bien le cas avec la hiérarchie d'appels.

10. Exécution locale :

- Dans la méthode `main()` de la classe `Test`, créez une instance de `Test` et appelez `echo()`.

- b. Créez une nouvelle configuration d'exécution pour lancer `Test` et testez la.
- c. Ajoutez à la classe `Test` une méthode `echo(String s)` qui affiche le message passe en paramètre.
- d. Modifiez la méthode `main` pour appeler `echo(String s)` si un argument est passe en paramètre.
- e. Modifiez votre configuration pour tester.
- f. Générez une méthode `main()` dans `SousTest` et créez une configuration d'exécution comme précédemment.



Pour aller plus loin

11. Vous pouvez éventuellement suivre le tutoriel qui vous permet de développer votre première application Android : <http://www.tutomobile.fr/premiere-application-hello-moi-tutoriel-android-n%C2%B03/15/06/2010/>.

Sources pour ce TP

Tutoriel Eclipse : <http://henri-garreta.developpez.com/tutoriels/eclipse/installation-utilisation-eclipse-developpement-java/#LX>

Installer SDK Android sur Eclipse : <http://www.tutomobile.fr/installer-le-sdk-android-sur-eclipse-tutoriel-android-n%C2%B01/09/06/2010/>

Cours de Fabrice Huet : <http://www-sop.inria.fr/oasis/personnel/Fabrice.Huet/Enseignement/IUT/GL/>