Equipe : Oryphis

Membre :

- ARRIGO Jordan
- ARNONE Vincent
- MARIE Tristan

Choix de l'éditeur de texte :

Sublime Text

I-Guide pratique :

Description des commandes:

man cmd (*manual*): affiche le manuel de la commande *cmd*
cd destination (*change directory*): permet de se déplacer dans le disque
ls (*list segments*): permet d'afficher les fichiers du répertoire courant
cp fichier destination (*copy paste*): permet de copier/coller le fichier *fichier* vers la destination *destination*
(ex: cp fichier1.txt dossier/sous-dossier/fichier2.txt copiera le fichier "fichier1.txt" vers "dossier/sous-dossier/fichier2.txt")
mv fichier destination (*move*): permet de déplacer le fichier *fichier* vers la destination *destination* (similaire au couper/coller)
mkdir dossier (*make directory*): permet de créer un sous-dossier *dossier* dans le répertoire courant
rm fichier (*remove*): permet de supprimer le fichier *fichier*. Pour supprimer un dossier (et tout son contenu !), il faut ajouter l'argument **-r** (ex: rm -r dossier).
rmdir dossier (*remove directory*): permet de supprimer un dossier **vide**.

Historique des commandes entrées :

Flèche du haut : va à la commande précédente dans l'historique.

Crtl -r : fait une recherche par occurance de l'expression que l'on tape dans l'historique des commandes.

Auto-Complétion

Pour auto-compléter une commande/nom de fichier, on appuie sur **Tab** lors de l'écriture de celle-ci.

Guide d'utilisation de **svn** :

help cmd: permet d'afficher l'aide par rapport à la commande *cmd*.
checkout url: permet de télécharger le dépôt entièrement et créer une copie locale.
info: affiche les informations relatives au dépôt (numéro de révision, auteur dernière modification…)
update: permet de récupérer la dernière révision du dépôt
add: permet d'ajouter un fichier au dépôt, il sera considéré lors des prochain commits.
commit: opération inverse de update, il permet de mettre a jour le dépôt via la copie du fichier local.

Conflits

Un conflit apparaît lorsque 2 utilisateurs essayent de modifier le même fichier.

Comment résoudre un conflit ?

Pour résoudre un conflit, on doit exécuter une fusion des fichiers qui sont en conflit, ou annuler nos modifications, ou celles de notre collègue.

Commande javac

La commande **javac *fichiersource.java*** permet de compiler le fichier source "fichiersource.java" vers un fichier *.class* exécutable par la JVM.
L'option **-sourcepath *repertoire*** permet de spécifier le répertoire contenant les fichiers sources.

Commande java

La commande java permet d'exécuter un fichier *.class* à l'aide de la JVM.
ex: **java Oryphis** exécutera la fonction main() de la classe Oryphis.
L'option **-classpath *repertoire*** permet de spécifier le répertoire contenant les fichiers exécutables.

Comment compiler et exécuter un programme java

Pour compiler un programme java, il faut compiler le fichier source principal (celui contenant la méthode **main()**.

Ex: un programme dont la classe principale est Oryphis (contenant la méthode **main()**) se compilera à l'aide de la commande "javac Oryphis.java" /!\ il faut s'être placé dans le même répertoire où sont situées les sources ou le spécifier à l'aide de -sourcepath.
Pour exécuter un programme java, il faut utiliser la commande "java NomClassePrincipale -cp RepertoireDeLExecutable"
Ex: un programme se situant dans le répertoire courant, dont la classe principale est Oryphis, se lancera à l'aide de la commande "java -cp . Oryphis".

Politique de nommage des fichiers .java et structure des dossiers

Les fichiers .java doivent avoir le même nom que la classe contenue dedans (ex, un fichier contenant une classe Oryphis doit être nommé "Oryphis.java").
Les dossiers et sous-dossiers permettent de définir les packages, ils doivent etre écrits en minuscules, sans caractères spéciaux.

I.1 Bonjour le monde !

Description de la gestion des arguments de la ligne de commande :

Pour gérer les arguments de la ligne de commande, nous bouclons (à l'aide d'une boucle for) sur args, ce qui nous permet de récupérer chaque argument successivement.
Code permettant d'afficher chaque argument/paramètre un par un, séparés par un retour chariot:

```
public static void main(String[] args)
    {
        for(int i = 0; i < args.length; i++) {
               System.out.println(args[i] + " ");
        }
    }
```

Si nous utilisons par exemple la commande "java -cp . Oryphis -a arg -b arg2", alors le programme nous affichera:
-a
arg
-b
arg2