Java Cours 1

**Introduction**

**Diapo 4**

Le nom d’une classe commence toujours par une Maj, les mots suivant composant le nom commence aussi par une Maj ex : public class HelloWorld

String[] args : un tableau de chaine de caractères, args nom de la variable.

La classe qui sert à démarrer l’exécution contient une méthode main.

Javac = compilateur pour java.

Le fichier source en java à le même nom et casse que sa classe : HelloWorld.java

Javac le compile en HelloWorld.class. Comme ici HelloWorld contient la méthode main ce fichier est exécutable. Le .class est du byteCode Java

On peut le lancer en ligne de commande avec java HelloWorld

**Diapo 5**

La commande : « Javap –c fichier.class » permet de créer le fichier assembleur généré par Java. Ce langage est créé pour un processeur virtuel qui n’existe pas.

**Diapo 6**

Les programmes Java sont hautement portables grâce à l’utilisation de la machine virtuelle qu’il utilise.

.java : source, .class : bytecode

**Diapo 7**

La machine virtuelle charge les classes en fonctions des besoins du programme. Si code de 1000 classes et seulement 500 utilisées, seulement 500 chargées par la machine virtuelle.

La machine virtuelle vérifie aussi le code (security manager) pour détecter du code « étranger ».

**Diapo 10**

Applet pas de méthode main, c’est le navigateur qui gère l’exécution. Applet plutôt culturel aujourd’hui.

**Diapo 11**

En java la modularité se fait par l’utilisation de plusieurs classes. La classe exécutable doit avoir la méthode « public static void main (string[] args) ». Si c’est absent le compilateur javac donnera une erreur indiquant qu’il n’y a pas de point d’entrée.

**Diapo 15**

API Java : toutes les bibliothèques disponibles pour Java. Ce sont des classes et interfaces définies avec des rôles spécifiques. On les retrouve dans des packages.

**Diapo 16**

Pour bien programmer en Java il faut savoir aller chercher les bons API permettant de réaliser ce que l’on veut faire.

**Diapo 17**

Java EE : pour faire des applications articulées autour du Web. Pour les applications serveurs.

**Diapo 20/21 :**

JRE pour exécuter les programmes en java

JDK tout pour exécuter et produire du code java.

**Diapo 22**les API que l’on utilisera de la version JEE.

**Diapo 23**

La couche métier est dans un conteneur qui gère les Tiers métier qu’il contient.

Conteneur web prend en charge les classes web.

Pour utiliser JEE, il faut un environnement complet pour JEE.

**Diapo 24**

On utilisera un sous ensemble de GlassFish (pour serveur TomCat). C’est une plateforme JEE open-source.

**Diapo 27**

.jar c’est une archive java. Utilisé lors de l’échange de fichier. Exemple quand on récupère un package.