

Annexe 1 pour le cours POO en java: Installation et usage de **jdk**

Cours de POO Java
2LMD: GL-POO
2013-2014
Dr. L.Kahloul

Objectif de cet exposé

Présenter **comment installer**, et **utiliser le **Jdk****
pour la compilation et l'exécution des
programmes java

C'est quoi le JDK?

- **JDK= Java Development Kit**= ensemble d'outils nécessaires pour la compilation et l'exécution des programmes java;
- JDK peut être utilisé sous windows ou linux;
- JDK est un freeware, accessible depuis:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
- Il faut faire attention aux versions: **32 bits ou 64 bits**

D:\Program Files\Java\jdk1.7.0_13\bin

gestion des fichiers

- Créer un nouveau dossier
- Publier ce dossier sur le Web
- Partager ce dossier

autres emplacements

- jdk1.7.0_13
- Mes documents
- Documents partagés
- Poste de travail
- Favoris réseau

tails

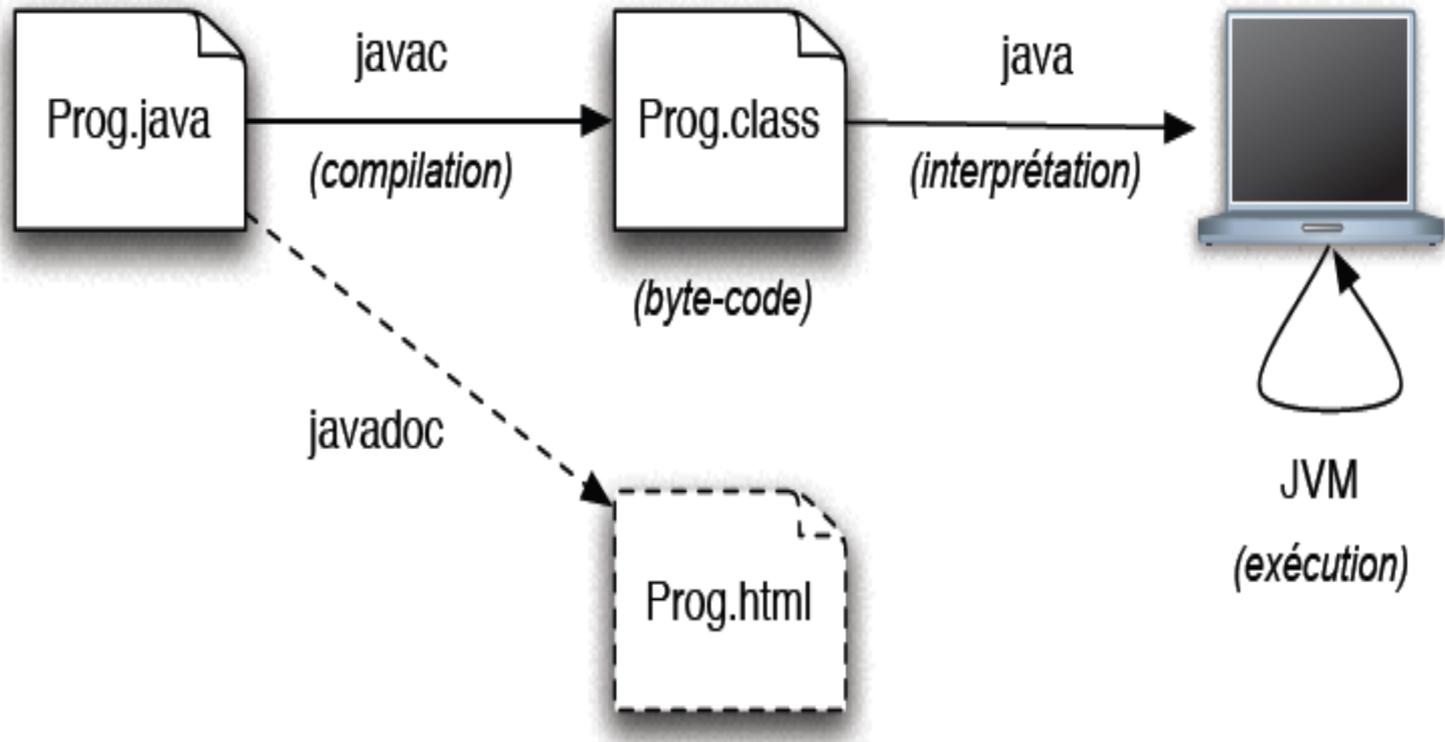
- | | |
|------------------|--------------------|
| msvcr100.dll | jdb.exe |
| jli.dll | jconsole.exe |
| beanreg.dll | jcmm.exe |
| xjc.exe | javaws.exe |
| wsimport.exe | javaw.exe |
| wsgen.exe | java-rmi.exe |
| unpack200.exe | javap.exe |
| tnameserv.exe | javah.exe |
| servertool.exe | javafxpackager.exe |
| serialver.exe | javadoc.exe |
| schemagen.exe | javac.exe |
| rmiregistry.exe | java.exe |
| rmid.exe | jarsigner.exe |
| rmic.exe | jar.exe |
| policytool.exe | jabswitch.exe |
| packager.exe | idlj.exe |
| pack200.exe | extcheck.exe |
| orbd.exe | apt.exe |
| native2ascii.exe | appletviewer.exe |
| ktab.exe | |
| klist.exe | |
| kinit.exe | |
| keytool.exe | |
| jvisualvm.exe | |
| jstatd.exe | |
| jstat.exe | |
| jstack.exe | |
| jsadebugd.exe | |
| jrunscript.exe | |
| jps.exe | |
| jmap.exe | |
| jinfo.exe | |
| jhat.exe | |



C'est quoi le JDK?

- Surtout on a besoin de deux applications:
 - 1) **javac.exe** : elle permet de compiler les fichiers java dont l'extension est « **.java** » et de générer des fichiers avec extension « **.class** »
la commande: **javac cercle.java**
génère le fichier: **cercle.class**
 - 1) **java.exe**: elle exécute les fichiers java générés par le compilateur **javac.exe**. La commande doit être:
java cercle

Fonctionnement (1)

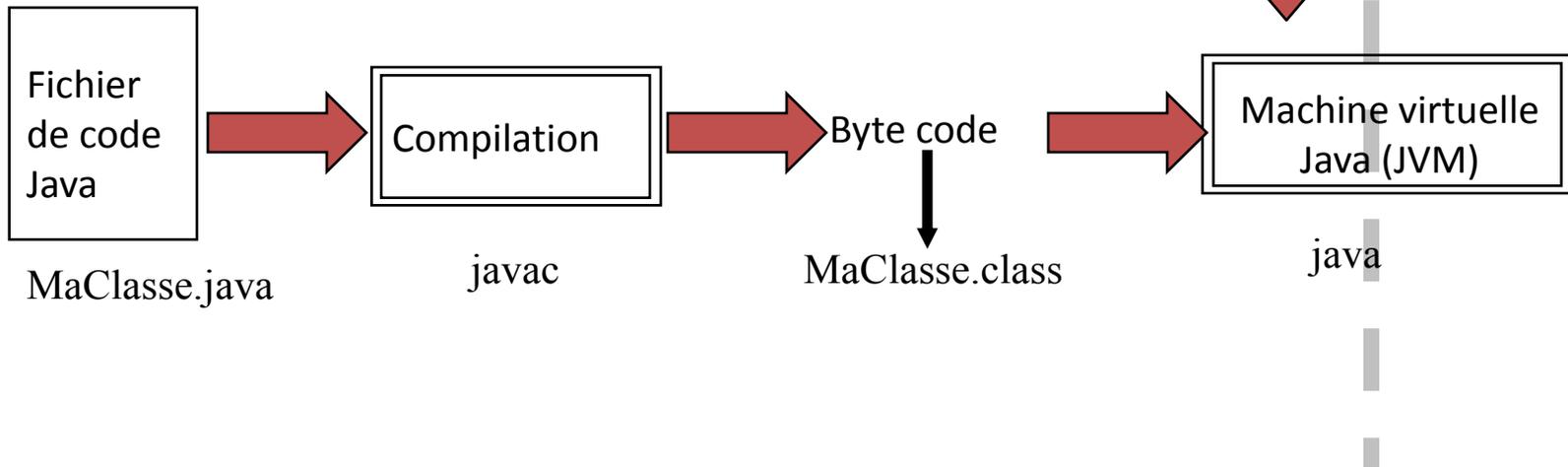


Fonctionnement (2)

Cas de Java

Avant exécution

Exécution



Fonctionnement (3): exemple

Attention: le nom du fichier (**avec extension java**) doit être comme le nom de la classe définie dans ce fichier;

Configuration de JDK: accès rapides des commandes javac et java

- Configurer votre système pour la reconnaissance des commandes **java** et **javac** :
- Normalement, si vous lancez l'invite de commande « **cmd** », vous allez avoir la possibilité d'exécuter des commandes **java**, et **javac**
- Sinon, vous devez configurer certains paramètres pour rendre ces deux commandes visibles dans l'invite de commandes, comme suit :
 1. Cliquez avec le bouton droit sur **Poste de travail**, puis cliquez sur **Propriétés**.
 2. Cliquez sur l'onglet **Avancé**.
 3. Cliquez sur **Variables d'environnement**.
 4. Dans l'onglet : **variable système**, choisir **path**, et ajouter le chemin de java et javac qui est dans mon cas : **D:\Program Files\Java\jdk1.7.0_13\bin**. On doit ajouter ce chemin, donc au **path** déjà existant.
 5. Vous devez redémarrer votre PC, pour que les nouveaux paramètres prennent effet. Une fois redémarré, votre système va reconnaître les commandes java et **javac** dans l'invite de commande **cmd**.

Construction d'un premier programme Java

Remarques sur la construction d'un programme java :

- Un programme java peut être composé de un ou de plusieurs fichiers;
- Un fichier doit avoir l'extension java : exemple **c1.java**
- Un fichier **c1.java** doit contenir une classe (une seule **déclarée publique**) nommé aussi **c1** (comme le nom du fichier);

```
public class c1{  
    public int x;  
    public static void main(String[] args){  
        int a;  
    }  
};
```

- La compilation sera faite comme suit : **javac c1.java**
- Après la compilation vous aurez un fichier **c1.class**
- L'exécution sera fait après avec la commande : **java c1.**

Exemple 1

```
public class Bonjour
{ //Accolade débutant la classe Bonjour
  public static void main(String args[])
  { //Accolade débutant la méthode main
    /* Pour l'instant juste une instruction */
    System.out.println("bonjour");
  } //Accolade fermant la méthode main
} //Accolade fermant la classe Bonjour
```

Fin de la première partie

Questions?