

TP1 Java

Y. ALJ

16 mars 2020

1 Préliminaire

Séparation des sources et des classes compilées Nous allons organiser nos fichiers de sorte que le code source des classes (fichiers .java) soit séparés des fichiers contenant le bytecode (fichier .class).

Afin que les fichiers créés pendant la compilation soient dans un autre répertoire on utilise l'option `-d`.

Exemple :

```
javac -d monRepertoireDeTravail/JAVA/classes/ HelloWorld.java
```

Pour l'exécution on utilise alors l'option `-classpath`.

Exemple :

```
java -classpath monRepertoireDeTravail/JAVA/classes/ HelloWorld
```

2 Exercices

Exercice 1

Partie 1 :

1. Créer un répertoire `tp1` (avec la commande `md` sous Windows, `mkdir` sous Linux ou avec l'explorateur de votre choix).
Nous allons mettre les fichiers sources .java dans un sous répertoire de `tp1` nommé `src`. Le résultat de la compilation (.class) devrait se trouver dans un sous répertoire de `tp1` appelé `bin`.
2. Avec Notepad++ créer un nouveau fichier nommé `HelloWorld.java` dans le répertoire `tp1/src/`.
3. Écrire une classe appelée `HelloWorld` qui affiche "bonjour tout le monde" à la sortie standard.
4. On se place avec la commande `cd` dans le répertoire `tp1` et on compile le fichier `HelloWorld.java` avec l'option `-d` (voir préliminaire).
5. Exécuter avec la commande `java` du préliminaire.

Partie 2 :

1. Effacer le contenu de la fonction `main` (uniquement ce qui se trouve à l'intérieur des accolades).
2. Créer un nouvel attribut privé de nom `message` de type `String`.
3. Ajouter à votre classe `HelloWorld` deux méthodes :
 - Un constructeur qui attribue une valeur au message.
 - Une méthode `public String getMessage()` qui renvoie la valeur du message.
4. Compiler à nouveau avec l'option `-d`.
5. Exécuter. Quelle remarque peut on faire ? Pourquoi ?
6. Dans la fonction `main` Créer une instance de la classe `HelloWorld` et afficher son message. On utilisera pour cela la fonction `System.out.println(String t)`.