TD1 - Comprendre Java

1. Qui est le créateur du langage Java ?

Java a été créé par James Gosling chez Sun Microsystems. Le développement a commencé en 1991 et la première version publique a été lancée en 1995.

2. Que signifie le slogan « Write Once, Run Anywhere » ?

« Write Once, Run Anywhere » (WORA) signifie qu'un programme Java écrit et compilé une seule fois peut s'exécuter sur n'importe quelle plateforme (Windows, Linux, macOS, etc.) sans modification ni recompilation, à condition qu'une JVM soit disponible sur cette plateforme.

3. Quel est le rôle de la JVM ?

La Java Virtual Machine (JVM) est une machine virtuelle qui :

- Exécute le bytecode Java
- Assure l'indépendance vis-à-vis de la plateforme
- Gère la mémoire (allocation et garbage collection)
- Vérifie la sécurité du code
- Optimise les performances à l'exécution (compilation JIT)

4. Expliquez la différence entre compilation et interprétation

Compilation : Le code source est traduit en une seule fois en code machine ou intermédiaire avant l'exécution. Le résultat est un fichier exécutable ou du bytecode.

Interprétation : Le code source est lu et exécuté ligne par ligne en temps réel, sans phase de traduction préalable complète.

Java utilise les deux : le code Java est d'abord compilé en bytecode, puis ce bytecode est interprété (et compilé à la volée) par la JVM.

5. Complétez le schéma suivant

Code Java (.java) → Compilateur (javac) → Bytecode (.class) → JVM → Exécution

6. Quel est le rôle du bytecode ?

Le **bytecode** est un code intermédiaire, indépendant de la plateforme :

- Format standardisé compris par toute JVM
- Résultat de la compilation du code Java
- Plus compact et optimisé que le code source
- Garantit la portabilité du programme
- Peut être vérifié pour la sécurité avant exécution

7. Pourquoi dit-on que Java est un langage portable ?

Java est portable car :

- Le bytecode est indépendant du système d'exploitation et de l'architecture matérielle
- La JVM fait abstraction des spécificités de chaque plateforme
- Les bibliothèques standard Java sont identiques sur tous les systèmes
- Un même fichier `.class` fonctionne partout où une JVM existe

8. Expliquez le rôle de chaque élément

Dans la déclaration : `public static void main(String[] args)`

public : Modificateur d'accès qui rend la méthode accessible depuis n'importe où. La JVM doit pouvoir appeler `main` depuis l'extérieur de la classe.

static : La méthode appartient à la classe elle-même et non à une instance. La JVM peut ainsi appeler `main` sans créer d'objet.

void : Type de retour indiquant que la méthode ne renvoie aucune valeur.

String[] args : Tableau de chaînes de caractères contenant les arguments passés en ligne de commande lors de l'exécution du programme.

9. Que se passe-t-il si le nom du fichier est différent du nom de la classe ?

Si le nom du fichier ne correspond pas au nom de la classe publique, le compilateur génère une erreur de compilation.

Règle : Pour une classe déclarée `public`, le fichier doit avoir exactement le même nom que la classe, avec l'extension `.java`.

Exemple : Une classe `public class Bonjour` doit être dans un fichier nommé `Bonjour.java`.

10. Que produit la commande suivante ?

```
javac Bonjour.java
```

Cette commande :

1. Lance le **compilateur Java** (`javac`)
2. Compile le fichier source `Bonjour.java`
3. Produit un fichier `Bonjour.class` contenant le bytecode
4. Génère des erreurs de compilation si le code contient des erreurs syntaxiques ou sémantiques

Si la compilation réussit, le fichier `.class` peut ensuite être exécuté avec : `java Bonjour`