

TP1

Objectifs du TP :

1. Comprendre ce qu'est le **bytecode** et son rôle dans l'exécution des programmes Java.
2. Explorer le fonctionnement de la **JVM (Java Virtual Machine)**.
3. Apprendre à utiliser l'**API Javadoc** pour documenter et comprendre le code Java.

Étape 1 : Comprendre le Bytecode et la JVM

1.1. Qu'est-ce que le Bytecode ?

- Le **bytecode** est un code intermédiaire généré par le compilateur Java (javac) à partir du code source Java (.java).
- Il est indépendant de la plateforme et peut être exécuté par la **JVM** sur n'importe quel système d'exploitation.
- Le bytecode est stocké dans des fichiers .class.

1.2. Qu'est-ce que la JVM ?

- La **JVM** est une machine virtuelle qui exécute le bytecode Java.
- Elle est responsable de la gestion de la mémoire, de l'optimisation du code, et de l'exécution des programmes Java.
- La JVM est spécifique à chaque plateforme (Windows, macOS, Linux, etc.), mais elle permet au bytecode d'être portable.

Exercice 1 : Générer et Analyser le Bytecode

1. Créer un fichier Java :

- Créez un fichier HelloWorld.java avec le code suivant :

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello, Bytecode!");  
  
    }  
  
}
```

2. Compiler le fichier Java :

- Ouvrez un terminal et compilez le fichier avec la commande :

```
javac HelloWorld.java
```

- Cela générera un fichier HelloWorld.class contenant le bytecode.

Étape 2 : Explorer la JVM

2.1. Fonctionnement de la JVM

- La JVM charge les fichiers .class et exécute le bytecode.
- Elle gère la mémoire (allocation, garbage collection) et assure la sécurité.

Exercice 2 : Exécuter un Programme Java

1. Exécuter le programme :

- Utilisez la commande suivante pour exécuter le programme :

```
java HelloWorld
```

Vous devriez voir la sortie :

Hello, Bytecode!

1. Comprendre le processus :

- La JVM charge le fichier HelloWorld.class.
- Elle interprète le bytecode et exécute les instructions.

Étape 3 : Utiliser l'API Javadoc

3.1. Qu'est-ce que Javadoc ?

- **Javadoc** est un outil qui génère une documentation HTML à partir des commentaires dans le code source Java.
- Les commentaires Javadoc commencent par `/**` et se terminent par `/`.

3.2. Comment utiliser Javadoc ?

1. Ajouter des commentaires Javadoc :

- Modifiez le fichier HelloWorld.java pour inclure des commentaires Javadoc :

```
/**
```

```
 * La classe HelloWorld affiche un message de bienvenue.
```

```
 */
```

```
public class HelloWorld {
```

```
/**
```

*** La méthode principale du programme.**

*** @param args Les arguments de la ligne de commande (non utilisés ici).**

***/**

```
public static void main(String[] args) {  
  
    System.out.println("Hello, Javadoc!");  
  
}  
  
}
```

2. Générer la documentation Javadoc :

- Utilisez la commande suivante pour générer la documentation :

javadoc HelloWorld.java

- Cela créera un dossier doc contenant des fichiers HTML.

3. Explorer la documentation :

- Ouvrez le fichier index.html dans le dossier doc pour voir la documentation générée.
- Vous verrez une description de la classe HelloWorld et de la méthode main.

Exercice 3 : Explorer l'API Javadoc Officielle

1. Accéder à l'API Javadoc :

- Rendez-vous sur la documentation officielle de l'API Java : [Java SE Documentation](#).

2. Rechercher une classe :

- Recherchez la classe String et explorez ses méthodes (par exemple, length(), substring()).

3. Comprendre la structure :

- Notez comment les méthodes sont documentées (description, paramètres, valeur de retour, exceptions).