

# Introduction à Java

CIPA4 – Java

Florian Bescher

ISEN Ouest – Année 2024/2025

Introduction à Java .....	2
Historique .....	3
Le langage Java .....	6
Structure d'un programme .....	9
Installation des outils .....	12
Hello World .....	13

## Équipe chez Sun Microsystems

- James Gosling
- Mike Sheridan
- Patrick Naughton

## Insatisfaction du langage C++

- API complexe
- Gestion de la mémoire manuelle
- Peu d'interfaces portables pour le *multi-threading*, la gestion des fichiers et la sécurité

## Première réalisation

- Assistant personnel (PDA) **Star7**



Fig. 1. – James Gosling, 2008



Fig. 2. – Logo Sun Microsystems, années 1990, Wikipedia

## 1994

- L'équipe répond à des appels d'offre de **TimeWarner** et **3DO Company** pour des décodeurs TV intelligents.

## Fin 1994

- Changement d'objectif pour aller vers des applications Web interactives
- Prototype **WebRunner** (HotJava)

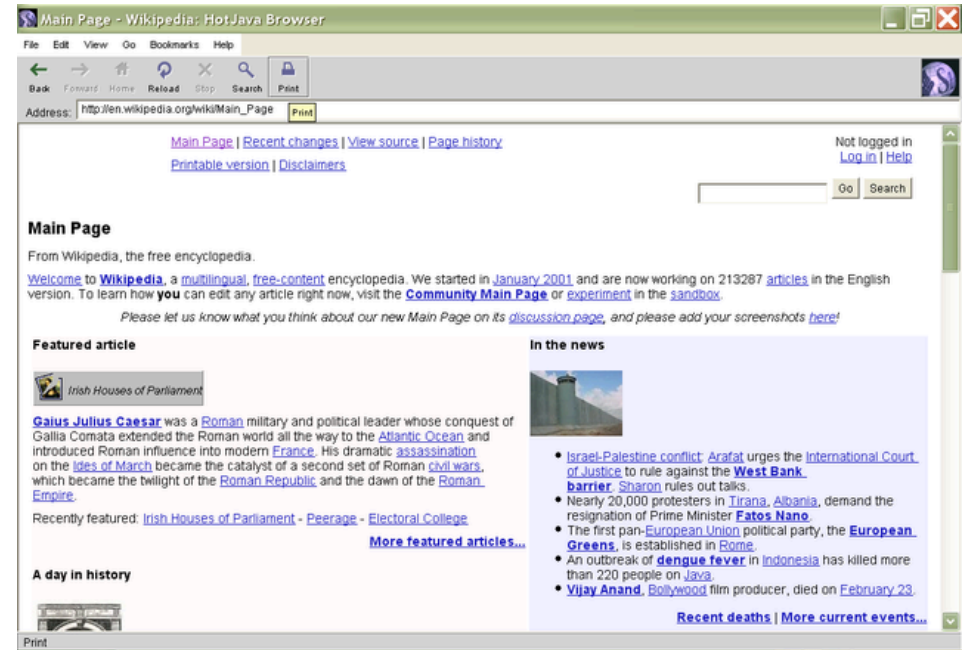


Fig. 3. – HotJava 3.0, Wikipedia

# Historique

## Java aujourd'hui

### Utilisations

- Systèmes embarqués grand public
- Applications de bureau
- Applications mobiles (Android)
- Applications Web et *backend*

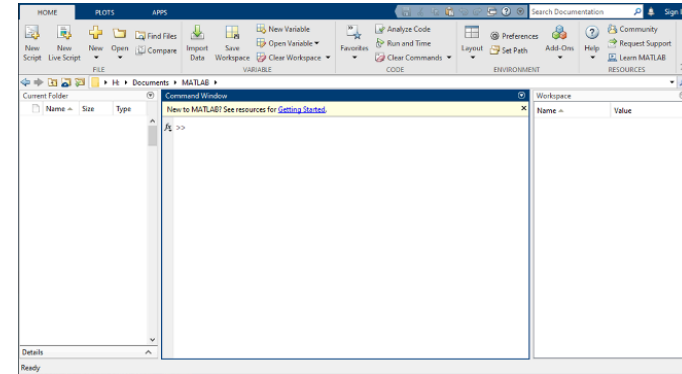


Fig. 4. – MatLab



Fig. 5. – Minecraft

### Le langage Java

- Orienté objet
- Fortement typé
- Gestion automatique de la mémoire  
(*Garbage collector*)



Fig. 6. – Logo du langage Java

# Le langage Java

# Java Virtual Machine

```
import java.io.IOException;

public class TomcatEmbedded {

    private static final String EMPTY = "";

    public static void main(String... args)
        throws Exception {
        File baseFolder = new File(System.getProperty("user.dir"));
        File appFolder = new File(baseFolder, "mid" + "apps");

        Tomcat tomcat = new Tomcat();
        tomcat.setBaseDir(baseFolder.getAbsolutePath());
        tomcat.setPort(8080);
        tomcat.getHost().setAppBase(appFolder.getAbsolutePath());

        // Call the connector to create the default connector.
        tomcat.getConnector();

        tomcat.addWebapp(EMPTY, docBase="");
        Wrapper wrapper = tomcat.addServlet(EMPTY, "hello", new HelloServlet());
        wrapper.setLoadOnStartup(1);
        wrapper.addMapping("/");

        tomcat.start();
        tomcat.getServer().await();
    }
}
```

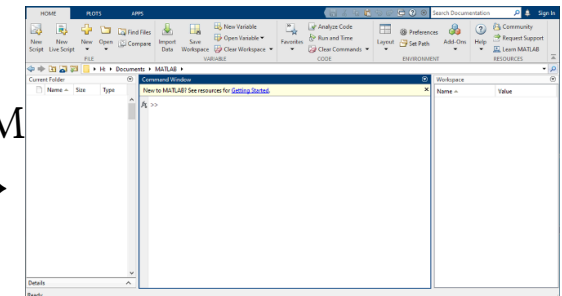
Source code

Java compiler  
→

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF
000000 CA FE BA BE 00 00 00 32 00 2C 07 00 02 01 00 09 .....2.....
000010 54 65 73 74 41 72 72 61 79 07 00 04 01 00 10 0A TestArray.....]
000020 01 76 61 2F 6C 61 0E 07 2F 4F 62 6A 65 63 74 01 ava/lang/Object.
000030 00 00 3C 69 6E 69 74 3E 01 00 03 28 29 56 01 00 .init...{V..
000040 04 43 6F 64 65 0A 00 03 00 09 0C 09 05 00 06 01 .Code.....
000050 00 0F 4C 69 6E 65 4E 75 6D 62 65 72 54 61 62 6C ..LineNumberTabl
000060 05 01 00 12 4C 6F 63 61 6C 56 61 72 69 61 62 6C e...LocalVariabl
000070 65 54 61 62 6C 65 61 00 94 74 68 69 73 61 00 08 eTable...this...
000080 4C 54 65 73 74 41 72 72 61 79 3B 01 00 04 6D 61 LTestArray...ma
000090 69 6E 01 00 16 28 5B 4C 6A 61 76 61 2F 6C 61 6E in...([Ljava/lang
0000A0 07 2F 53 74 72 69 6E 67 3B 29 56 07 00 11 01 00 g/String;)V....
0000B0 02 5B 49 09 00 13 00 15 87 00 14 01 00 18 6A 61 .[I.....jA
0000C0 76 61 2F 6C 61 6E 67 2F 53 79 73 74 65 6D 8C 00 va/lang/System.
0000D0 16 09 17 01 00 03 6F 75 74 01 00 15 4C 6A 61 76 .....out...Ijav
0000E0 61 2F 69 6F 2F 50 72 69 6E 74 53 74 72 65 61 6D a/io/PrintStream
0000F0 3B 00 00 19 01 00 07 65 8C 65 6D 65 6E 74 0A 00 ;....element..
000100 1B 00 10 07 00 1C 01 00 13 6A 61 76 61 2F 69 6F .....java/io
000110 2F 50 72 69 6E 74 53 74 72 65 61 6D 6C 00 1E 00 /PrintStream...
000120 1F 01 00 07 70 72 69 6E 74 6C 6E 01 00 15 28 4C ...println...(L
000130 6A 61 76 61 2F 6C 61 6E 67 2F 53 74 72 69 6E 67 java/lang/String
```

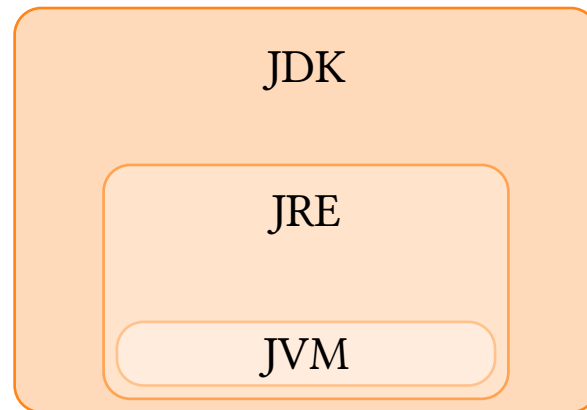
ByteCode

JVM  
→



Application

- **JRE (*Java Runtime Environment*)** : fourni la JVM et les bibliothèques Java
- **JDK (*Java Development Kit*)** : fourni le compilateur et les outils nécessaires aux développement d'applications Java





- Une application Java est constituée uniquement de **classes**.
- Une **classe** correspond à **un fichier .java** qui doit avoir le **même nom que la classe**.  
Par exemple la classe `Application` est écrite dans un fichier `Application.java`
- Le point d'entrée d'un programme Java est la fonction `main` avec la signature suivante :

```
public static void main(String args[])
```

java

La syntaxe est similaire au C et au C++

### Point-virgule

- Une ligne de code se termine par un point-virgule ;

### Commentaires

- Une ligne : //
- Multilignes : /\* ... \*/

### Structures de contrôle

- `if (/*...*/) {} else {}` java
- `while (/*...*/) {}` java
- `for (/*...*/) {}` java

- Les variables et fonctions sont en *camel case* :

```
public static int addNumbers(int firstNum, int secondNum);
```

java

- Les classes sont en *pascal case* :

```
public class Application
```

java

- Les constantes sont en **majuscules**. Les mots sont séparés par des *underscores* :

```
const int NOMBRE_ETUDIANTS = 50;
```

java

- Pour installer les outils, suivez les instructions dans la section Moodle du cours intitulée **Préparation de votre environnement.**

Hello World  
Hello World

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

java