B-01 Java Basics

> Hamza El Kassir
>
> Khairi Adam

HISTOIRE DE LANGAGE JAVA :

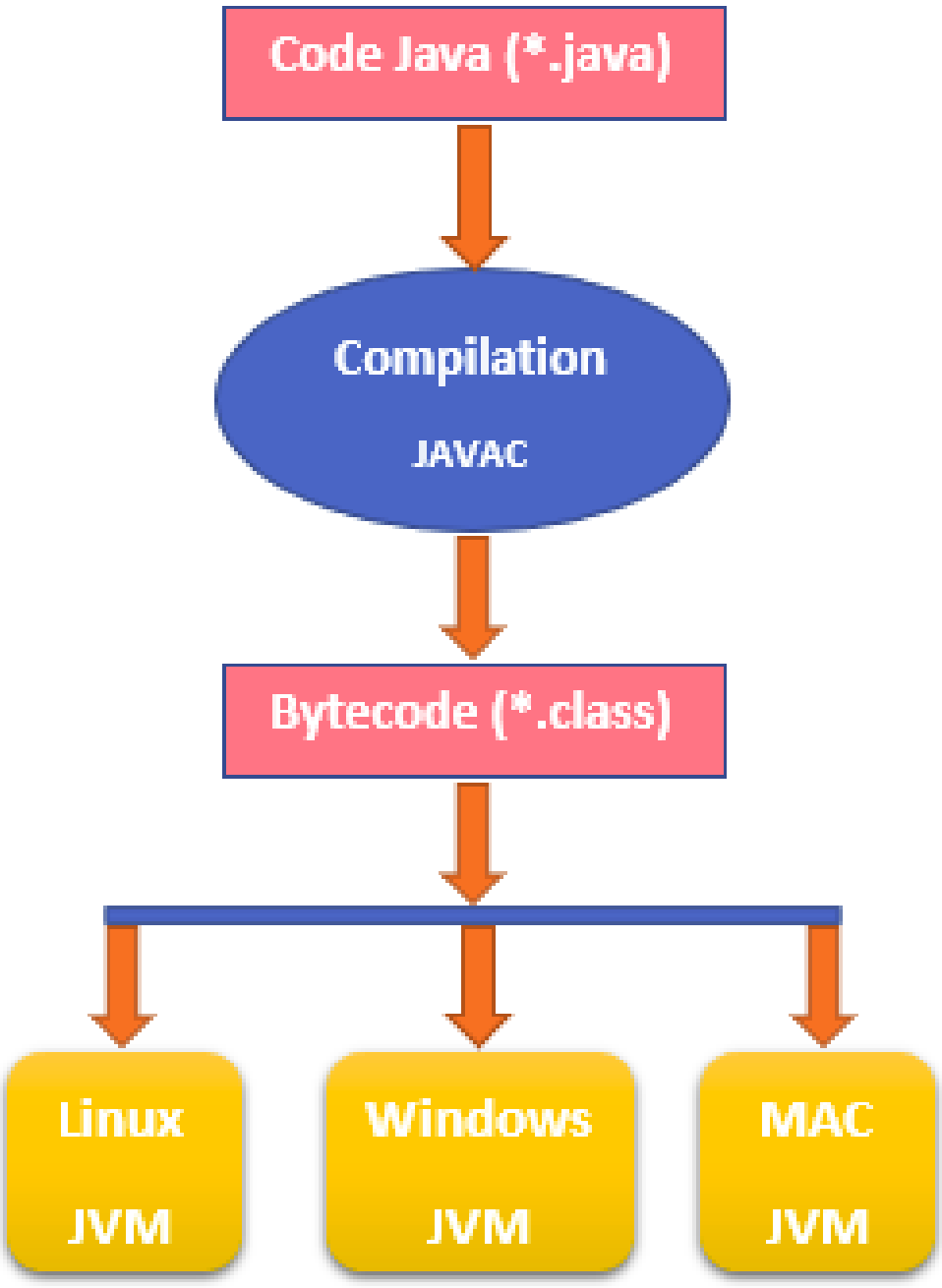
Le langage Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, Le projet Java est né en 1991, dans le secret d'une équipe de Sun Microsystem. C'est cette année-là que le Green Project a démarré, dans le but d'étudier l'impact de la convergence entre les appareils ménagers (blancs et bruns) contrôlés numériquement et les ordinateurs.

Fonctionnement de Java

Le langage Java a été conçu de la manière suivante :

- Le code source est écrit dans des fichiers dont l'extension est **.java**

- Il est ensuite compilé par le compilateur Java (javac) pour générer des fichiers dont l'extension est **.class** appelé **byte code.** C'est un format binaire intermédiaire et indépendant du code machine.
- Le byte code peut ensuite être interprété est exécuté par l'interpréteur Java qui le transforme en code machine natif.



Introduction to Java

* Variables :

1) Local Variable

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

A local variable cannot be defined with "static" keyword.

2) Instance Variable

A variable declared inside the class but outside the body of the method, is called instance variable. It is not declared as static.

It is called instance variable because its value is instance specific and is not shared among instances.

3) Static variable

A variable which is declared as static is called static variable. It cannot be local. You can create a single copy of static variable and share among all the instances of the class. Memory allocation for static variable happens only once when the class is loaded in the memory.

- Examples of the types of variables in java :

```
class A{
  int data=50;//instance variable
  static int m=100;//static variable
  void method(){
    int n=90;//local variable
  }
}//end of class
```

- Java Variable Example: Add Two Numbers:

```
class Simple{
    public static void main(String[] args){
    int a=10;
    int b=10;
    int c=a+b;
    System.out.println(c);
    }
  }
```

* Loops in Java

Loops in Java

Loops are used to iterate the code a specific number of times until the specified condition is true. There are three kinds of loop in Java

Loops :

Aa For Loop	Example
Iterates the code for a specific number of times until the condition is true.	`class TestForLoop { public static void main (String args[]) { for(int i=0;i<=5;i++) System.out.println("*"); } }`
While Loop	
If condition in the while is true the program enters the loop for iteration.	class TestWhileLoop `{ public static void main (String args[]) { int i = 1; while(i<=10) { System.out.println(i); i++; } } }`
Do While Loop	
The program enters the loop for iteration at least once irrespective of the while condition being true. For further iterations, it is depends on the while condition to be true.	`class TestDoWhileLoop { public static void main (String args[]) { int i = 1; do { System.out.println(i); i++; } while(i<=10); } }`

Methods in Java

The general form of method :

Where type return type of the methodname name of the methodparameter list - sequence of type and variables separated by a commareturn - statement to return value to calling routine

```
type name (parameter list)
{
//body of the method
//return value (only if type is not void)
}
```

- Example :

```
public void sexGenre(String genre, Integer age) {
        if (genre.equals("homme") || genre.equals("Homme") && age >= 18) {
            System.out.println("Vous êtes un homme et vous êtes majeur");
        } else if (genre.equals("homme") || genre.equals("Homme") && age < 18)
{
            System.out.println("Vous êtes un homme et vous êtes mineur");
        } else if (genre.equals("femme") || genre.equals("Femme") && age >= 1
8) {
            System.out.println("Vous êtes une femme et vous êtes majeur");
        } else if (genre.equals("femme") || genre.equals("Femme") && age < 18)
{
            System.out.println("Vous êtes une femme et vous êtes mineur");
        }
    }
```

```
public void concatStrNum(String str, Double num) {
        System.out.println(str.concat(String.valueOf(num)));
    }

public void bonjour(String fName, String lName, Integer age) {
        System.out.println("Hello ," + fName + lName + ",you have " + age + "
 years old");
    }
```

Collections

The collections framework contained in the `java.util` package defines a set of interfaces and their implementations to manipulate collections, which serve as a container for a group of objects.

- Example :

```
import java.util.*;
class Num
{
  public static void main(String args[])
  {
    ArrayList num = new ArrayList ();
    num.add(9);
    num.add(12);
    num.add(10);
    num.add(16);
    num.add(6);
    num.add(8);
    num.add(56);
    //printing array list
    System.out.println("Elements : ");
    num.forEach((s) -> System.out.println(s));
  }
}
```