

# 1 Introducción á tecnoloxía Java

## 1.1 Sumario

- 1 Orixes da linguaxe
- 2 Características da linguaxe
- 3 A máquina virtual de Java
- 4 O xestor automático de memoria
- 5 A seguridade da plataforma Java
- 6 Exercicios

## 1.2 Orixes da linguaxe



A linguaxe de programación Java nace no ano 1995 da man da empresa [Sun Microsystems](#). O termo Java refírese a unha taza de café en argot americano, de aí o seu logotipo.

Java é unha linguaxe interpretada polo que pode executarse en calquera computador que teña un intérprete Java. A este intérprete chámase a **Java Virtual Machine** (JVM) ou máquina virtual de Java. A JVM non precisa un hardware nin un S.O. concreto. De feito, o eslogan de Java é *Write once, run everywhere?*.

## 1.3 Características da linguaxe

Java é unha linguaxe **orientada a obxectos**, polo que incorpora toda a potencialidade deste paradigma de programación: reutilización de código, encapsulación, etc.

A súa sintaxe é similar á de C e C++. Permite desenvolver aplicacións convencionais, igual que estas linguaxes, pero tamén *applets*.

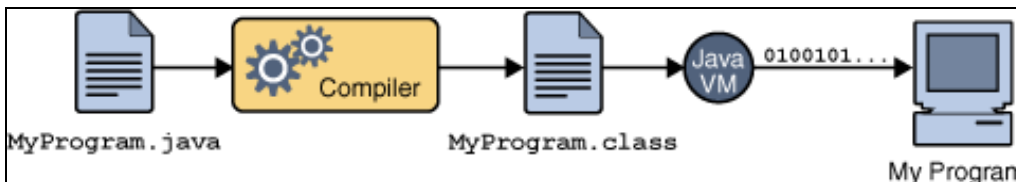
Fronte a C e C++ é unha linguaxe máis simple no que se refire ao tratamento de punteiros, xa que, dende o punto de vista do programador, non ten. Para xestionar a memoria utiliza unha ferramenta incluída na plataforma que se chama **garbage collector** que elimina obxectos cando non se utilizan.

Con Java pódense crear aplicacións concorrentes e incorpora mecanismos específicos para garantir a seguridade no contorno de desenvolvemento.

Por último, é unha linguaxe interpretada polo que incrementa a velocidade de desenvolvemento de programas, xa que se elimina o ciclo compilación-enlace-carga-verificación. Asemade, esta característica faíno portable, é dicir, pode executarse en calquera computador que teña unha JVM (*Java Virtual Machine*), é dicir, un intérprete.

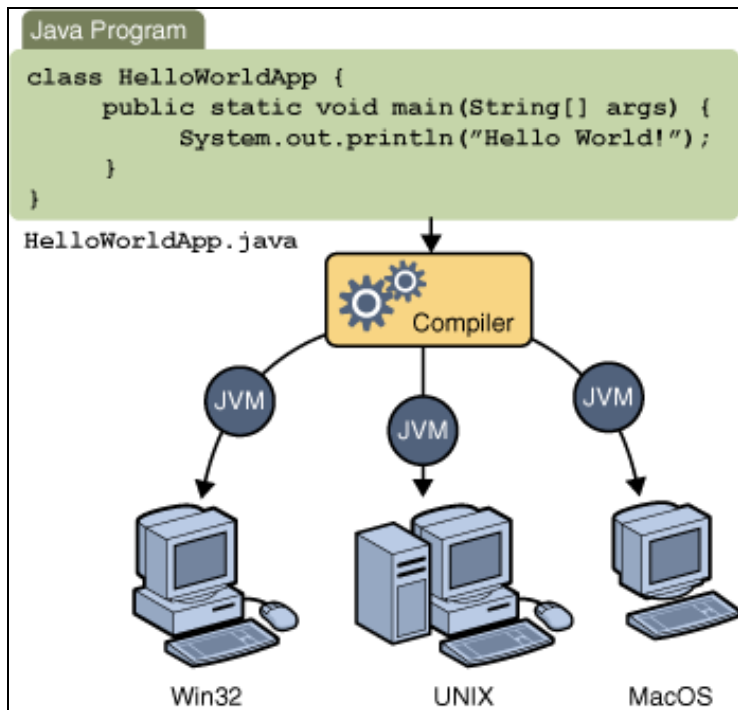
## 1.4 A máquina virtual de Java

O código fonte dos programas Java non se compilan directamente para crear código máquina dunha arquitectura de procesador concreta, senón que se traducen a un código intermedio chamado bytecodes. Estes bytecodes son instrucións que executan accións na máquina virtual (Java Virtual Machine), polo tanto serven para calquera arquitectura hardware.



A JVM non é como un intérprete dunha linguaxe de alto nivel no sentido de que a entrada da máquina virtual son bytecodes e a saída, código máquina, mentres que a entrada dun intérprete son instrucións da linguaxe e a saída, código máquina. Os bytecodes non son instrucións da linguaxe, senón código intermedio.

Deste xeito conseguimos que Java sexa multiplataforma xa que se pode executar en calquera arquitectura, sempre e cando teñamos unha máquina virtual para ela.



## 1.5 O xestor automático de memoria

Noutras linguaxes, como C, o programador é o encargado de reservar e liberar memoria para determinadas variables. Java usa unha ferramenta, o Recolector de Lixo (Garbage Collector), que comproba que memoria pode liberarse e libérraa automaticamente, sen intervención do programador. Con todo, existe a posibilidade de liberar recursos mediante mecanismos específicos que se verán noutras seccións.

## 1.6 A seguridade da plataforma Java

Java incorpora mecanismos específicos para garantir a seguridade do sistema. Por unha parte, todo o código execútase dentro da máquina virtual e por outra, cando se cargan os bytecodes existe un proceso de verificación dos mesmos antes de que estes se executen.

## 1.7 Exercicios

1. Cal é a diferenza entre un compilador e un intérprete?
2. Que son os bytecodes?
3. Cal é a diferenza entre a *Java Virtual Machine* e un intérprete?
4. Describe coas túas palabras tres características da linguaxe de programación Java.
5. Busca exemplos en Internet de linguaxes de programación interpretados, como Java
6. Utilizando un editor de textos copia o seguinte código fonte no teu computador:

```
/**  
 * A clase MeuPrograma implementa unha aplicación que  
 * simplemente imprime o texto "¿Qué hay de nuevo, viejo?" na saída estándar.  
 */  
class MeuPrograma {  
    public static void main(String[] args) {  
        System.out.println("¿Qué hay de nuevo, viejo?"); // Mostra a cadea.  
    }  
}
```

Compila o código fonte utilizando a ferramenta javac e tecleando a seguinte orde (ollo coas maiúsculas e minúsculas):

```
javac MeuPrograma.java
```

Isto dará como resultado un ficheiro .class que traduce o texto que acabas de teclear a instrucións para a JVM.

1. Que cres que conterá o ficheiro .class?

Comproba que o programa funciona en GNU/Linux e en MS-Windows e que o resultado é o mesmo. Para elo executa o ficheiro .class mediante a orde:

