

PROGRAMACIÓN

UD 1: Introducción a la programación(I).

Índice

1. Programa y lenguajes de programación.
2. P.O.O. El lenguaje JAVA.

1. Programa y lenguajes de programación.

- *¿Qué es un programa?* Es una serie de órdenes o instrucciones ordenadas con una finalidad concreta que realizan una función determinada.
- *¿Qué es un lenguaje de programación?*
 - Lenguaje: conjunto de símbolos, caracteres y palabras (léxico) y conjunto de reglas (sintaxis y semántica) que permiten agrupar los símbolos para formar las frases del lenguaje.
 - De programación: Sirve para *especificar algoritmos sobre un ordenador*. Un programa se escribe como una *secuencia de frases del lenguaje, con el objetivo de producir alguna acción o salida*.

1. Programa y lenguajes de programación.

- ¿Qué es un **algoritmo**?
 - Es un conjunto de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad.
 - Dados un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución.

1. Programa y lenguajes de programación.

- Tipos de **lenguajes de programación**:
 - *Lenguajes de bajo nivel.*
 - *Lenguajes de medio nivel.* (algunos autores)
 - *Lenguajes de alto nivel.*

1. Programa y lenguajes de programación.

Lenguajes de Bajo Nivel:

- Lenguaje muy cercano a la máquina.
- Las instrucciones del lenguaje son las instrucciones del microprocesador del ordenador.
- Difícil y costoso de programar.
- Cada máquina tiene su propio lenguaje.

● Tipos:

- *Lenguaje Máquina*: Único lenguaje que entiende el ordenador, el binario (secuencias de 0 y 1: 0110 1100 1111).
- *Lenguaje Ensamblador*: Códigos mnemotécnicos. Cada instrucción ensamblador equivale a una sola instrucción en lenguaje máquina.

● Ejemplo:

<u>Ensamblador</u>	<u>Código Máquina (Hex)</u>
mov ax, 003	B8 03 00

1. Programa y lenguajes de programación.

Lenguajes de Medio Nivel:

- Lenguaje muy cercano al usuario, pero con características del lenguaje de bajo nivel. También se acerca, en menor medida, al lenguaje máquina.
- Algunos autores no diferencian este nivel, y a estos lenguajes los clasifican dentro de los lenguajes de alto nivel.
- Mayor independencia de la máquina. Mismo programa en distintas máquinas.
- Programación más fácil y menos errores que los lenguajes de bajo nivel.
- No es difícil de programar, pero sí que es más difícil que programar que un lenguaje de alto nivel.
- ¿Cuál? El único que cumple todas estas características es el **lenguaje C**.
 - Tiene características de lenguaje de alto nivel (en su mayoría) pero algunas características de lenguaje de bajo nivel. Por ejemplo: tiene punteros a memoria (*a) y direcciones de memoria (&b).

1. Programa y lenguajes de programación.

Lenguajes de Alto Nivel:

- Lenguajes próximos al lenguaje humano.
 - Mayor independencia de la máquina. Mismo programa en distintas máquinas.
 - Programación más fácil y menos errores.
 - Instrucciones más flexibles y potentes.
- Ejemplos:
- Java, C++, C#...

1. Programa y lenguajes de programación.

- ¿Qué es el **código fuente**? Es el que escribe el programador en un lenguaje de programación que luego lo compila a código máquina.
- ¿Qué es el **compilar**? Consiste en transformar el lenguaje inteligible por el programador a lenguaje inteligible por la máquina.
- El código fuente está escrito en un lenguaje de programación y el compilador es un programa que se encarga de transformar el código fuente en código máquina.
- Cada lenguaje tiene su propio compilador.

1. Programa y lenguajes de programación.

- JAVA genera un lenguaje máquina intermedio interpretable por una máquina virtual instalada en el ordenador donde se va a ejecutar.
- Una *máquina virtual* es una máquina ficticia que traduce las instrucciones máquina ficticias en instrucciones para la máquina real.
- **Compiladores:** convierten lenguajes de alto nivel a código máquina. Realizan primero la **traducción completa**, y si no hay errores se genera el código máquina.
- **Intérpretes:** convierten lenguajes de alto nivel a código máquina. Se traduce y ejecuta instrucción a instrucción.

2. Programación Orientada a Objetos. El lenguaje JAVA.

Comienzos...

- Java fue diseñado en 1990 por *James Gosling*, de *Sun Microsystems*, como software para dispositivos electrónicos de consumo.
- Curiosamente, todo este lenguaje *fue diseñado antes de que diese comienzo la era World Wide Web*, puesto que fue diseñado *para dispositivos electrónicos como calculadoras, microondas y la televisión interactiva*.

2. Programación Orientada a Objetos. El lenguaje JAVA.

Tres de las principales razones que llevaron a crear Java son:

1. Creciente **necesidad de interfaces mucho mas cómodas** e intuitivas que los sistemas de ventanas que proliferaban hasta el momento.
2. **Fiabilidad del código y facilidad de desarrollo.** Gosling observó que muchas de las características que ofrecían C o C++ aumentaban de forma alarmante el gran coste de pruebas y depuración. Por ello en los sus ratos libres **creó un lenguaje de programación donde intentaba solucionar los fallos que encontraba en C++**.
3. **Enorme diversidad de controladores electrónicos.** Java permite escribir un código común para todos los dispositivos/máquinas.

2. Programación Orientada a Objetos. El lenguaje JAVA.

- Basándose en el conocimiento y estudio de gran cantidad de lenguajes, este grupo decidió *recoger las características esenciales que debía tener un lenguaje de programación moderno y potente, pero eliminando todas aquellas funciones que no eran absolutamente imprescindibles.*

2. Programación Orientada a Objetos. El lenguaje JAVA.

Características de Java.

- Potente.
 - *Orientación a objetos*
 - *Riqueza semántica*
 - *Robusto*
 - Descubrimiento de gran parte de los errores en tiempo de compilación.
 - Garbage Collector.
 - *Modelo de objeto rico*
 - Clases abstractas a partir de las cuales partir y extender de funcionalidades extra.
- Simple.
 - *Fácil aprendizaje*
 - *Completado con utilidades*
 - JDK incluye: compilador, debugger, visualizador de applets, etc.
- Interactivo y orientado a red (Ej. Los Applets dan animación a la Web.)
- Seguridad.
 - Diseñado para no realizar llamadas a funciones globales ni acceder a recursos arbitrarios del sistema.
- Basado en C++.
- Gestión de Entrada/Salida.
 - Permiten tratar los archivos, sockets, teclado y monitor como flujos de datos.

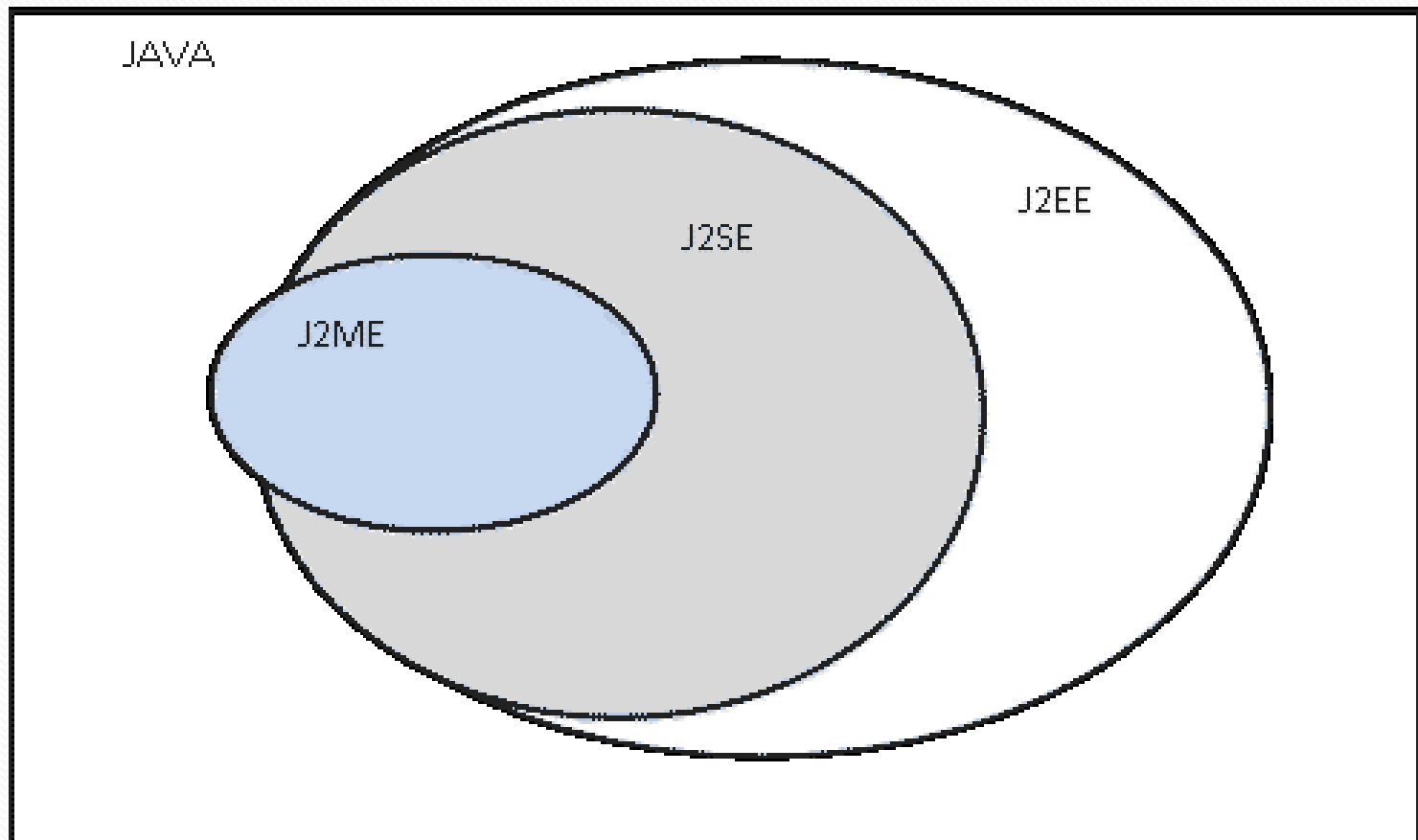
2. Programación Orientada a Objetos. El lenguaje JAVA.

Versiones y distribuciones de Java:

- **JSE: *Java Standard Edition*.**
 - Orientado al desarrollo de aplicaciones cliente/servidor. No incluye soporte a tecnologías para internet.
- **JEE: *Java Enterprise Edition*.**
 - Orientado a empresas y a la integración entre sistemas. Incluye soporte a tecnologías para internet.
- **JME: *Java Micro Edition*.**
 - Orientado a pequeños dispositivos móviles (teléfonos, tablets, etc.).

2. Programación Orientada a Objetos. El lenguaje JAVA.

Versiones y distribuciones de Java:



2. Programación Orientada a Objetos. El lenguaje JAVA.

Versiones y distribuciones de Java: EVOLUCIÓN JSE

- **JDK 1.0** (1996): primer lanzamiento del lenguaje Java.
- **JDK 1.1** (1997): mejora de la versión anterior.
- **J2SE 1.2** (1998): esta y las siguientes versiones fueron recogidas bajo la denominación *Java 2* y el nombre "J2SE" (*Java 2 Platform, Standard Edition*), reemplazó a JDK para distinguir la plataforma base de J2EE (*Java 2 Platform, Enterprise Edition*) y J2ME (*Java 2 Platform, Micro Edition*). Incluyó distintas mejoras.
- **J2SE 1.3** (2000): mejora de la versión anterior.
- **J2SE 1.4** (2002): mejora de la versión anterior.
- **J2SE 5.0** (2004): originalmente numerada 1.5, esta notación aun es usada en ocasiones. Mejora de la versión anterior.
- **Java SE 6** (2006): en esta versión, Sun cambió el nombre "J2SE" por Java SE y elimino el ".0" del numero de versión. Mejora de la versión anterior.
- **Java SE 7** (2011): nueva versión que mejora la anterior.
- **Java SE 8** (2014): nueva versión que mejora la anterior.
- ...

2. Programación Orientada a Objetos. El lenguaje JAVA.

Maquina Virtual Java (JVM).

- Entorno en el que se ejecutan los programas Java, su *misión principal es la de garantizar la portabilidad de las aplicaciones Java.*
- Define esencialmente un *computador abstracto y especifica las instrucciones (bytecodes) que este computador puede ejecutar.*
- El interprete Java especifico *ejecuta las instrucciones que se guardan en los archivos cuya extensión es '.class'.*

2. Programación Orientada a Objetos. El lenguaje JAVA.

- Los tipos de programas más comunes que se pueden hacer con Java son:
 - los **applets** (se ejecutan en el navegador de la maquina cliente) (** Obsoletos **)
 - y las **aplicaciones** (programas que se ejecutan directamente en la JVM).
 - Otro tipo especial de programa se denomina **servlet** que es similar a los applets pero se ejecutan en los servidores Java.
- La **API** de Java es muy rica, *está formada un conjunto de paquetes de clases que le proporcionan una gran funcionalidad.*

2. Programación Orientada a Objetos. El lenguaje JAVA.

- *Modelo de programación más utilizado actualmente.*
- La **clase** es el elemento central de un programa OO. Determina el comportamiento y características propias de sus componentes.
- Al caso particular (instancia) de una clase se le denomina **objeto**.
- Un programa se entiende como un conjunto de objetos que interactúan entre sí.
- Principal ventaja POO -> *facilita el reuso del código ya realizado.*

2. Programación Orientada a Objetos. El lenguaje JAVA. Conceptos Básicos.

Objetos:

- *Un objeto del mundo real es cualquier cosa que vemos a nuestro alrededor:* un árbol, una persona, un coche...
- Ejemplo: Una computadora está compuesta internamente por varios componentes: la placa base, el chip del procesador, un disco duro, una tarjeta de video, y otras partes más.
- El trabajo en conjunto de todos estos componentes hace operar a una computadora. Se comunican entre ellos a través de mensajes.
- No nos interesa qué es lo que hace internamente cada uno de estos componentes, sino qué hace (qué funciones nos brinda) que nos pueda ser útil.

2. Programación Orientada a Objetos. El lenguaje JAVA. Conceptos Básicos.

Clases:

- En el mundo real, normalmente tenemos muchos objetos del mismo tipo. Por ejemplo, *nuestro teléfono móvil es solo uno de los miles que hay en el mundo*.
- Si hablamos en términos de la programación orientada a objetos, podemos decir que *nuestro objeto móvil es una instancia de una clase conocida como “móvil”*.
- Los **móviles** tienen características (marca, modelo, sistema operativo, pantalla, teclado, etc.) y comportamientos (hacer y recibir llamadas, enviar mensajes multimedia, transmisión de datos, etc.).

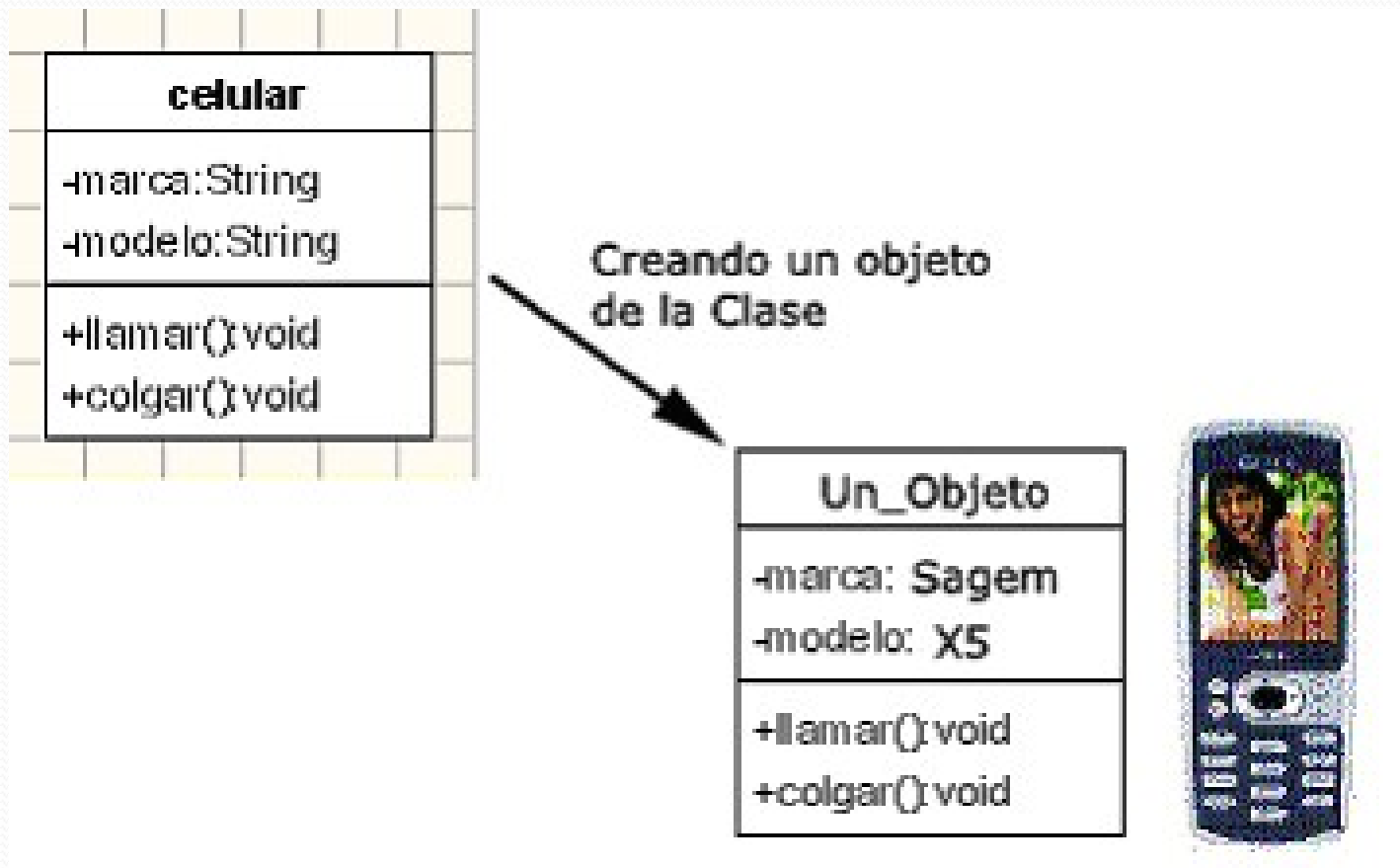
2. Programación Orientada a Objetos. El lenguaje JAVA. Conceptos Básicos.

Clases: (continuación)

- Cuando se fabrican los móviles, los fabricantes aprovechan el hecho de que los móviles comparten esas características comunes y construyen modelos o plantillas comunes, para que a partir de esas se puedan crear muchos equipos móviles del mismo modelo.
- A ese modelo o plantilla le llamamos **CLASE**, y a los equipos que sacamos a partir de ella la llamamos **OBJETOS**.
- Esto mismo se aplica a los objetos de software, se puede tener muchos objetos del mismo tipo y mismas características, pero cada uno de ellos será **UNICO**.

2. Programación Orientada a Objetos. El lenguaje JAVA. Conceptos Básicos.

Clases: (continuación)



2. Programación Orientada a Objetos. El lenguaje JAVA. Conceptos Básicos.

Clases: (continuación)

- Definición teórica: La **clase** es un *modelo o prototipo que define las variables y métodos comunes a todos los objetos de cierta clase*. También se puede decir que una clase es una *plantilla genérica para un conjunto de objetos de similares características*.
- Por otro lado, una instancia de una clase es otra forma de llamar a un objeto.
- En realidad *NO existe diferencia entre un objeto y una instancia*.