

ENTORNOS DE DESARROLLO

IES Santiago Hernández
Curso 2017-2018
Ignacio Agudo Sancho

UD 2. Los Programas Software

- ⦿ Introducción
- ⦿ ¿Qué es un programa?
- ⦿ Lenguajes de programación
- ⦿ Tipos de código y compilación
- ⦿ La máquina virtual de Java

UD 2. Los Programas Software

- Introducción
- ¿Qué es un programa?
- Lenguajes de programación
- Tipos de código y compilación
- La máquina virtual de Java

Introducción

- ⦿ Ordenador = Hardware + Software
- ⦿ Hardware = Tangible
- ⦿ Software = Programas y aplicaciones que actúan sobre el hardware del ordenador y facilitan al usuario la realización de diferentes tareas.

UD 2. Los Programas Software

- Introducción
- ¿Qué es un programa?
- Lenguajes de programación
- Tipos de código y compilación
- La máquina virtual de Java

Concepto de Programa

- ⦿ Conjunto de instrucciones escritas en un lenguaje de programación que aplicadas sobre un conjunto de datos resuelven un problema o parte del mismo.
- ⦿ Es necesario traducirlo a lenguaje máquina. Esto lo hace el compilador.
- ⦿ Después, cargarlo en memoria principal para que el procesador ejecute una a una todas las instrucciones.

UD 2. Los Programas Software

- Introducción
- ¿Qué es un programa?
- Lenguajes de programación
- Tipos de código y compilación
- La máquina virtual de Java

Concepto de Lenguaje

- ⦿ Conjunto de caracteres, reglas para la combinación de esos caracteres y reglas que definen sus efectos cuando los ejecuta un ordenador.
- ⦿ Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Clasificación y características

- ⦿ Según su nivel de abstracción
 - Lenguajes de bajo nivel
 - Lenguajes de nivel medio
 - Lenguajes de alto nivel
- ⦿ Según la forma de ejecución
 - Lenguajes compilados
 - Lenguajes interpretados
- ⦿ Según el paradigma de programación
 - Lenguajes imperativos
 - Lenguajes funcionales
 - Lenguajes lógicos
 - Lenguajes estructurados
 - Lenguajes orientados a objetos

Según su nivel de abstracción

⦿ Lenguajes de bajo nivel

- Cerca del funcionamiento del ordenador
- El más bajo es el lenguaje máquina
 - Instrucciones formadas por 0s y 1s.
- Le sigue el lenguaje ensamblador.
 - Específico para cada procesador.
 - Necesita ser traducido a lenguaje máquina.
 - Las instrucciones trabajan directamente con registros de memoria física de la máquina.

Según su nivel de abstracción

⦿ Lenguajes de medio nivel

- Características que los acercan a lenguajes de bajo nivel pero también características de los lenguajes de alto nivel.
- Se suelen utilizar para creación de SO.
- Ejemplo: C

Según su nivel de abstracción

⦿ Lenguajes de alto nivel

- Fáciles de aprender, formados por palabras del lenguaje natural.
- Necesitan de un intérprete o un compilador para ser ejecutados, que traduzca las instrucciones escritas en lenguaje máquina.
- Son independientes de la máquina.
- Ejemplos: Java, PHP, Visual .Net, COBOL, Python, C++, etc.

Según la forma de ejecución

⦿ Lenguajes compilados

- El compilador traduce de un determinado lenguaje a un programa equivalente.
- El compilador devolverá errores si el programa en el lenguaje fuente no está bien escrito.
- El programa generado es más rápido que un intérprete.

Según la forma de ejecución

◉ Lenguajes interpretados

- Los intérpretes, en vez de producir un programa destino como resultado, nos da la apariencia de ejecutar directamente las operaciones especificadas en el programa fuente con las entradas proporcionadas por el usuario.
- Cada vez que se ejecuta una instrucción se debe interpretar y traducir a lenguaje máquina.
- El intérprete elimina la necesidad de realizar una compilación después de cada modificación.
- Ej: PHP, Java Script, Python, Ruby, etc.

Según el paradigma de programación

- ⦿ El paradigma es un enfoque particular para la construcción del software.
- ⦿ Define un conjunto de reglas, patrones y estilos de programación.
- ⦿ Dependiendo del problema a resolver, un paradigma resultará más apropiado que otro.

Según el paradigma de programación

⦿ Lenguajes imperativos

- Indican al ordenador CÓMO realizar una tarea.
- La sentencia principal es la asignación.
- Instrucciones simples.
- Los primeros fueron los lenguajes máquina y ensambladores.

Según el paradigma de programación

⦿ Lenguajes funcionales

- Basado en el concepto matemático de función.
- No existe la operación de asignación.
- Las variables almacenan definiciones o referencias a expresiones.
- La operación fundamental es la aplicación de una función a una serie de argumentos.
- Poco utilizados.

Según el paradigma de programación

⦿ Lenguajes lógicos

- En ellos un cálculo es el proceso de encontrar qué elementos de un dominio cumplen o no alguna relación definida sobre dicho dominio.
- Se pueden ver como bases de datos formadas por listas de reglas.
- La ejecución consistirá en realizar preguntas
- Ej: Prolog → Sistemas expertos, demostración de teoremas, etc.

Según el paradigma de programación

⦿ Lenguajes estructurados

- Utilizan estructura secuencial, condicional y repetitiva.
- Pueden ser leídos secuencialmente desde el comienzo hasta el final sin perderse.
- Problema: Si es demasiado grande, resulta difícil su lectura y manejo.
- Actualmente con programación estructurada nos referimos a la división de un programa en módulos más manejables, con sus entradas y salidas.

Según el paradigma de programación

● Lenguajes Orientados a Objetos

- Los programas están compuestos por un conjunto de objetos.
- Un objeto consta de una estructura de datos (atributos) y de una colección de métodos u operaciones que manipulan esos atributos. Las operaciones definen el comportamiento de los objetos y cambian valores de sus atributos.
- Las clases son plantillas para la creación de objetos. Al crear un objeto hay que indicar de qué clase es.

UD 2. Los Programas Software

- Introducción
- ¿Qué es un programa?
- Lenguajes de programación
- Tipos de código y compilación
- La máquina virtual de Java

Tipos de código

⦿ Código fuente:

- Es el escrito por los programadores.
- Se utiliza un lenguaje de programación de alto nivel.
- Tradicionalmente se parte de los diagramas de flujo o pseudocódigos diseñados en la etapa de diseño.
- No es directamente ejecutable por el ordenador.

Tipos de código

⦿ Código objeto

- Es el código resultante de compilar el código fuente.
- No es directamente ejecutable por el ordenador ni entendible por el ser humano.
- Es un código o interpretación intermedia de bajo nivel.

Tipos de código

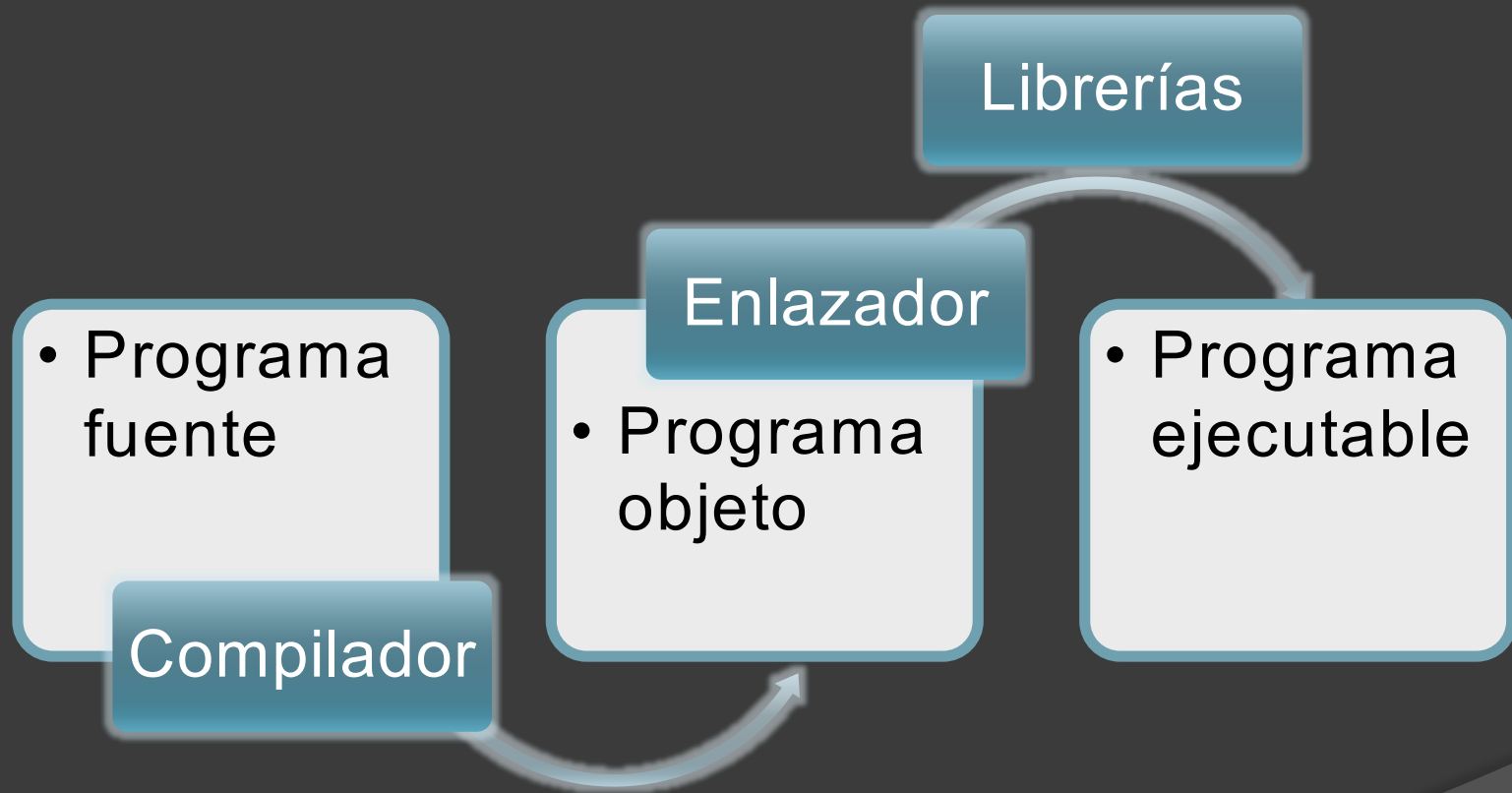
⦿ Código ejecutable

- Es el resultado de enlazar el código objeto con una serie de rutinas y librerías.
- Es directamente ejecutable por la máquina.

Compilación

- ⦿ Se lleva a cabo mediante dos programas: el compilador y el enlazador.
- ⦿ El compilador puede devolver errores, que habrá que subsanar y harán que no se genere el código objeto.
- ⦿ El enlazador inserta en el código objeto las funciones de librería necesarias para producir el programa ejecutable.

Compilación



Compilación

- ⦿ Se compone internamente de varias etapas o fases (1/2):
 - Análisis léxico: Lee todo y lo convierte a tokens.
 - Análisis sintáctico: Lee los tokens y los analiza.
 - Análisis semántico: Comprueba las declaraciones, los tipos de las expresiones, si las operaciones se pueden realizar sobre esos tipos, si los tamaños son adecuados...

Compilación

- ⦿ Se compone internamente de varias etapas o fases (2/2):
 - Generación de código intermedio: similar al código máquina para facilitar la traducción al código objeto.
 - Optimización de código: para que sea más fácil y rápido de interpretar por la máquina.
 - Generación de código: genera el código objeto de nuestro programa.

UD 2. Los Programas Software

- Introducción
- ¿Qué es un programa?
- Lenguajes de programación
- Tipos de código y compilación
- La máquina virtual de Java

Máquinas virtuales

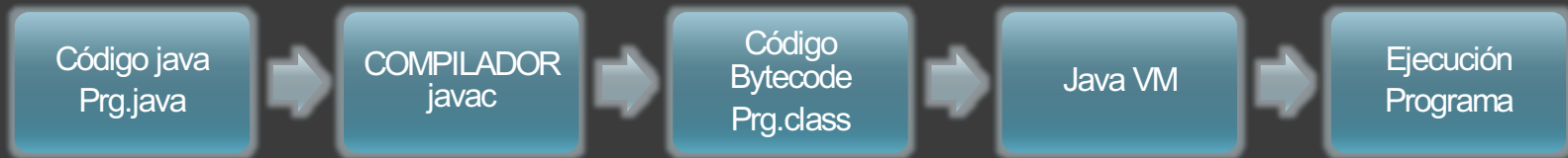
- ⦿ Una VM es una aplicación software que ejecuta programas como si fuera una máquina real.
- ⦿ Dos tipos:
 - De sistema: una máquina física, varias virtuales con sistemas operativos. VM Ware o Virtual Box.
 - De proceso o de aplicación: se ejecuta como un proceso normal dentro de un SO y soporta un solo proceso. Se inicia automáticamente cuando se lanza el proceso que se desea ejecutar y se detiene cuando finaliza. Objetivo: proporcionar un entorno de ejecución independiente de la plataforma. Java VM.

La Máquina Virtual de Java

- Una máquina virtual es una aplicación software de una máquina que ejecuta programas como si fuese la máquina real.
- Los programas compilados en lenguaje Java se pueden ejecutar en cualquier plataforma porque el código generado por el compilador no lo ejecuta el procesador del ordenador sino la JVM.

La Máquina Virtual de Java

● Funcionamiento:



La Máquina Virtual de Java

⦿ Tareas principales de la JVM:

- Reservar espacio en memoria para los objetos creados y liberar la memoria no utilizada.
- Comunicarse con el sistema huésped (ej. controlar el acceso a hardware)
- Vigilar el cumplimiento de las normas de seguridad de las aplicaciones Java

UD 2. Los Programas Software

- Introducción
- ¿Qué es un programa?
- Lenguajes de programación
- Tipos de código y compilación
- La máquina virtual de Java