24/07/2019

**Programación orientada a objetos**

Breve introducción a Java

-Java es un lenguaje de programación orientado

-Se rige con la filosofía “Write Once, Run Anywhere”, Lo que significa que es portable

-Al compilar el código, **no se** genera lenguaje maquina si no que se genera un código intermedio conocido como ByteCode

-La única maquina capaz de ejecutar ByteCode es la “Java Virtual Machine” (JVM)

-La JVM es especificada por Oracle y es implementada por varias empresas. Ósea existen distintas “marcas” de

JVM Oracle

IBM

etc.

-En una computadora puede correr varias instancias de JVM

-Cada proceso Java se ejecuta en una JVM aparte

-Java+JVM+Utilidades para programar, conforman una plataforma del software par ale desarrollo de aplicaciones de usuario final

-Existen varias plataformas en Java:

* SE (Standard Edition) Aplicaciones de escritorio
* ME (Mobile Edition) Dispositivos Moviles
* EE (Enterprice Edition) Aplicaciones Servidor

-Java se distribuye de varias formas:

* JRE- Mínimo para ejecutar programas
* JDK – JRE+SE
* Servidores de aplicaciones – Implementaciones de EE

Java

JVM

.class (ByteCode)

JDK(Compilador)

ompilador)

Hardware

26/07/2019

**Paradigmas de Programación**

* En general, es una forma de ver/entender/modelar el mundo
* En el ámbito del software, es un estilo fundamental de programación. Determina cómo el programa “ve” el mundo
* Determina como debe ser usado el lenguaje por el programador
* Algunos lenguajes son multiparadigma (No se hace una selección explícita en el código)
* Hay 4 paradigmas:
* Funcional
* Lógico
* Imperativo
* Orientado a objetos
* Paradigma Lógico
  + El mundo se modela mediante predicados lógico
  + Se aplican directamente principios de matemática discreta
  + Poco utilizado en aplicaciones comerciales
  + Prolog es el principal lenguaje (Que está al 100% en este paradigma)
* Paradigma Funcional
  + El mundo se modela como funciona matemáticas
  + Lenguajes 100% funcionales:
    - Lisp
    - Scheme
* Permite declarar funciones, pasar funciones por parámetro, retornar funciones entre otros
* Muchos lenguajes modernos, han incorporado este paradigma por conveniencia
* Expresiones Lambda
* Paradigma imperativo
  + El mundo se modela como instrucciones, pasos, procedimientos
  + Tienden a ser muy eficientes, generalmente compilan en lenguaje máquina
  + Gramaticalmente simples
  + Se consideran como de nivel de abstracción bajo
  + Ejemplos: c / c++
* Paradigma Orientado a objetos
  + El mundo se modela como objetos del mundo real junto con su interacción
  + Muy naturales para el ser humano
  + Facilitan la reutilización de código
  + Mayoritariamente, no generan lenguaje máquina
  + Poseen muchas estructuras sintácticas
  + Java / C++ / Python / Javascript…

Existen 2 tipos de “tipos” (palabras reservadas):

* Primitivos
* Por referencia

|  |  |
| --- | --- |
| Primitivos | Por referencia |
| int | Object |
| float | String |
| double |  |
| char |  |
| short |  |
| byte |  |

Nota: **Por referencia** significa que la variable guarda la posición en la memoria donde se encuentra el valor asignado

Primitivo

int x=5;

int y=x;

|  |  |
| --- | --- |
| x | 5 |
| y | 5 |

Copia el valor que tiene la variable que le asignamos

Por referencia

String a =”ey!”;

String b=a;

|  |  |
| --- | --- |
| 1) | ey! |
| a | 1) |
| b | 1) |

En este caso lo que copia es la dirección donde esta guardado el valor

Para un String totalmente independiente tiene que utilizarse “new”

Ejemplo:

String c= new String(“casa”);

String d= new String(c);

|  |  |
| --- | --- |
| 1) | “casa” |
| c | 1 |
| 3) | “casa” |
| d | 3 |