**Tema #1: Programación Orientada a Objetos**

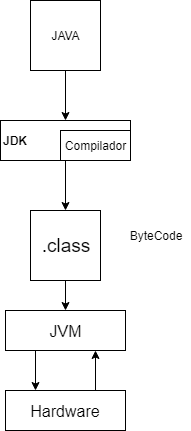
**Breve introducción a Java**

* Java es un lenguaje de programación orientado a objetos.
* Se rige con la filosofía “Write once, run anywhere”. (Es portable)
* Al compilar el código, no se genera lenguaje máquina, si no que se genera un código intermedio que se llama ByteCode.
* La única máquina capaz de ejecutar ByteCode es la “Java Virtual Machine”. (JVM)
* La JVM es especificada por Oracle y es implementada por varias empresas. O sea, existen distintas “marcas” de JVM como: Oracle, IBM entre muchas más.
* En una computadora pueden correr varias instancias de JVM.
* Cada proceso Java se ejecuta en una JVM aparte.
* Java + JVM + Utilidades para programar conforman una plataforma para el desarrollo de aplicaciones de usuario final.
* Existen varias plataformas en Java:

1. SE (Standar Edition): Aplicaciones de escritorio.
2. ME (Mobile Edition): Dispositivos móviles.
3. EE (Entreprise Edition): Aplicaciones servidor.

* Java se distribuye de varias formas:

1. JRE : Mínimo para ejecutar un programa.
2. JDK : JRE + SE.
3. Servidor de aplicaciones: Implementaciones de EE.



Tema 2: Programación Orientada a Objetos.

* **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 como el programa “ve” el mundo.
* Determina como debe ser usado el lenguaje por el programador.
* Algunos lenguajes de programación son multiparadigma.
* Hay 4 paradigmas principales:
  + - * + Funcional
        + Lógico
        + Imperativo
        + Orientado a objetos

1. **Paradigma lógico:** 
   * + El mundo se modela mediante predicados lógicos.
     + Se aplica directamente principios de matemática discreta.
     + Para utilizarla en aplicaciones comerciales.
     + **Prolog** es el principal lenguaje.
2. **Paradigma Funcional:**

* El mundo se modela como funciones 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.

1. **Paradigma imperativo:**

* El mundo se modela como instrucciones, pasos, procedimientos.
* Ampliamente utilizados en aplicaciones comerciales.
* Tienden a ser muy eficientes (generalmente compilan en lenguaje máquina).
* Gramaticalmente simples.
* Se consideran como de nivel de abstracción bajo.
* Ejemplos: **C, C++**

1. **Paradigma Orientado a Objetos:3**

* 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.
* **Java, C++, Python, JavaScript.**

**VARIABLES.**

Existen al menos dos tipos de **tipos de variables:**

* **Tipos primitivos:** Son los más básicos. Ejemplos: int, float, char, double, byte

* **Tipos referencia:** Son más complejos. String se hereda de Object.

**CONCEPTOS BÁSICOS DE LA PROGRAMACIÓN ORIENTADA A OBJETOS**

**Objetos:**

Es abstracto.

Se caracterizan por tener atributos y comportamiento.

Todos los objetos son referencias.

Todos los atributos deben ser por general privados.

**Abstracción:**

Modelar algo dependiendo de las características dependiendo del problema que tengo que solucionar, dejando de lado todo aquello innecesario o complejo.

**Encapsulamiento:**

Existen cuatro tipo de modificadores de acceso:

* Private
* Public
* Protected
* Friendly (Default)