Curso de Java

# Introducción

Java es un lenguaje de programación de alto nivel, amigable para los desarrolladores. Es multiplataforma, lo que significa que, gracias a la Máquina Virtual de Java (JVM), los programas creados en Java pueden ejecutarse en cualquier sistema operativo. Fue creado en 1995 por James Gosling, originalmente llamado OAC. Hoy en día, muchas empresas utilizan Java para desarrollar sus aplicaciones. La empresa responsable del soporte y las actualizaciones de Java es Oracle.

# ¿Qué se necesita para programar en Java?

1. Java Development Kit (JDK): Un conjunto de herramientas que permite escribir y compilar código en Java.

2. IDE (Entorno de Desarrollo Integrado): Programa que facilita la escritura de código en Java (ejemplo: IntelliJ IDEA, Eclipse, NetBeans).

# Conceptos básicos

## Definición de una clase:

public class Main {  
  
Método main:  
  
public static void main(String[] args) {  
  
}

- public: Modificador de acceso, permite que el método sea accesible desde otras clases.  
- static: Define que el método pertenece a la clase, no a una instancia de ella.  
- void: Indica que el método no retorna ningún valor.

Nota: Todo lo que no termine con llaves {} debe terminar con un punto y coma (;).

# Variables en Java

Una variable es un contenedor que almacena datos durante la ejecución de un programa. Cada variable tiene un tipo de dato específico.  
Declaración de una variable sigue la estructura:  
Tipo de dato: Define qué tipo de información almacenará la variable.  
Nombre de la variable: Un identificador que debe ser legible y significativo.  
Valor (opcional): Puedes asignarle un valor inicial.

## Ejemplos de declaración de variables:

int numero = 2;  
  
String cadena = "Hola";  
  
long decimal = 2.1;

## Normas para Nombres de Variables

- El nombre de una variable debe ser legible y tener significado.  
- Debe comenzar con minúscula.  
- Las variables representan espacios de memoria que pueden cambiar su valor durante la ejecución del programa.

## Tipos de Datos Primitivos en Java

- boolean: Valores true o false.  
- char: Un carácter único.  
- byte: Entero con signo de 8 bits.  
- short: Entero con signo de 16 bits.  
- int: Entero con signo de 32 bits.  
- long: Entero con signo de 64 bits.  
- float: Número decimal de precisión simple (32 bits).  
- double: Número decimal de precisión doble (64 bits).

## Atajo Importante: soutv

El atajo "soutv" se utiliza para imprimir el valor de una variable en la consola rápidamente.

# Clases en Java

Una clase es una plantilla que contiene atributos (datos) y métodos (funciones).

# Palabras Reservadas en Java

En Java, existen palabras reservadas que no pueden ser utilizadas como identificadores de variables: abstract, finally, public, boolean, float, return, break, for, short, byte, static, etc.

# Operadores Aritméticos

- Suma: '+'  
- Resta: '-'  
- Multiplicación: '\*'  
- División: '/'  
- Residuo: '%'  
- Paréntesis para evaluar expresiones: '()'

# Operadores de Igualdad

- Igualdad: '=='  
- Distinto: '!='  
- Mayor: '>'  
- Menor: '<'

# Operadores de Incremento y Decremento

- Incremento: '++' (Ej: c++ equivale a c = c + 1)  
- Decremento: '--' (Ej: c-- equivale a c = c - 1)

# Operadores Lógicos

- And: '&&' (Ej: si sexo == masculino && edad >= 18)  
- Or: '||' (Ej: si promedio >= 5 || notaexamen >= 6 ApruebaCurso)  
- Negación: '!' (Ej: boolean flag = !(5 > 3); dará false.)

# Operador Ternario

Sentencia if-else en una sola línea:  
  
boolean var = (true) ? true : false;

# Estructuras de Control

## If-Else

if (condición) {  
  
// Código si es verdadero  
  
} else {  
  
// Código si es falso  
  
}

## While

while (condición) {  
  
// Código a ejecutar  
  
}

## For

for (inicialización; condición; incremento) {  
  
// Código a ejecutar  
  
}

## Foreach

String[] arreglo = {"hoLA", "QUE", "tal"};  
  
for (String elemento : arreglo) {  
  
// Código a ejecutar  
  
}

## Switch

switch (variable) {  
  
case valor1:  
  
// Código para caso valor1  
  
break;  
  
default:  
  
// Código si no coincide ningún caso  
  
}

# Clases y Métodos

Una clase es una plantilla con atributos y comportamientos (métodos).

## Método Constructor

Un método especial para crear objetos, debe tener el mismo nombre que la clase:  
  
public Persona(String nombre, int edad, String sexo) {  
  
this.nombre = nombre;  
  
}

# Objetos en Java

Un objeto es una instancia de una clase.

# Métodos en Java

public boolean mayorEdad(Persona p) {  
  
if (p.getEdad() >= 18) {  
  
return true;  
  
} else {  
  
return false;  
  
}  
}

# La clase String

Variable en la que se declaran variables de cadena o tipo texto, pero esta clase aborda muchas mas cosas que solo eso. Cuando un objeto es inmutable este no puede cambiar su valor.

String saludo = “Hola”;

Saludo = saludo + “Mundo”;

# La clase Math

Es una clase matematica que tiene muchos metodos a disposicion para poder programar, como poder sacar raiz cuadrada, maximos y minimos etc.

Public class final Math

# La clase System

Clase para mostrar cosas por pantalla, pero Tambien para obtener informacion del Sistema o del computador que se esta usando.

Public class final System

# La clase Date y Calendar

Como sus nombres indican sirven para colocar fechas del Sistema etc

# Que Son Los Arreglos

Los arreglos existen en todos los lenguajes de programacion, es una estructura donde puedes almacenar varios datos al mismo tiempo, pero deben ser del mismo tipo de dato, es la unica restriccion, estos se declaran asi:

Int nombreArr [} new int [12}

Los arreglos se pueden inicializar:

Int nombreArr{} = {1,2,3,4,5,…}

Cómo accedemos a esos elementos?

nombreArreglo[indice}

# Que Son Las Matrices

Son arreglos de 2 dimensiones organizadas en filas y columnas.

# Programación Orientada a Objetos

Se trabaja solo con Objetos

# Modificadores de Acceso

Existen varios:

Public: Una clase de tipo public, metodos tipo public o campo de tipo public, cualquier otra clase que este en el programa puede acceder a ese metodo o campo.

Private: Es un modificador mas restrictive, nadie puede acceder desde otra clase a este modificador.

Protected: Se puede acceder desde otras clases pero que esten en el mismo packadge.

# Encapsulamiento

Se logra usando los modificadores de acceso, se pueden definer atributos en una clase private, y acceder desde otra clase., ejemplo:

# Metodos get y set

Se usan para setear valor u obtener valores de esos atributos privados,

Constantes

Valores que no se pueden Modificar a lo largo de los programas.

This

Su uso principal es para referirse a objetos o mienbros de otra clase, en un metodo constructor. Basicamente es para usar atributos de la clase en el constructor.

Static

Puede ser accedido sin crear una instancia de la clase. Es compartida por otras instancias.

Herencia

Es darle atributos a otras clases, como su nombre indica, herada cosas de los padres a los hijos. Con la palabra extends

Interfaces