Java

# Clase 1

Aplicaciones moviles que te ayudaran a aprender mas rapido.

* Learn java
* Sensei of code
* AIDE – IDE for android java C++

Este curso esta orientado a personas que desean aprender java para desarrollar aplicaciones moviles en android studio.

# Inicio

Para comenzar tienes que saber que java es un lenguaje de programación muy enganchado con la programación orientada a objetos por lo que escucharas la palabra **clase** muy seguido, por ahora solo tienes que saber la estructura inicial, deberás colocarla siempre, pero más adelante entenderás que significa.

public class NombreArchivo {

public static void main(String[] args){

//aquí ira tu código

}

}

Donde dice “NombreArchivo” siempre se empieza con mayúscula y obligatoriamente tendrá que tener el mismo nombre del archivo incluyendo la mayúscula, de lo contrario el programa no funcionara.

# Como ejecutarlo…

En windows se agrega la dirección del programa **javac** que se encuentra en archivos de programa, java y bin. Esa dirección se agrega a PATH en preferencias del equipo, variables de entorno.

Tanto en Linux como un S.O. de windows el código java debe ser compilado primero y luego ejecutado, primero debes colocarte en la carpeta donde se encuentre tu archivo .java (tanto en cmd como terminal de Linux) y primero debes escribir **javac NombreArchivo.java** esto compilara el código que escribiste y vera si contiene errores, de no tener ninguno generara un archivo **NombreArchivo.class,** si abres este archivo tendrá un monton de letras inentendibles, esto quiere decir que el computador “tradujo” el código que tu escribiste a algo más que el pudiera entender. Luego para poder ejecutarlo en la terminal debes escribir **java NombreArchivo** sin ninguna extensión y si todo sale bien debería funcionar.

Esto es necesario para cuando programamos en editores de texto como sublime text o notepad++ pero también hay una posibilidad de ocupar un IDE (entorno de desarrollo) como eclipse, entre otros. Ellos siempre tienen la funcionalidad de compilar y ejecutar al mismo tiempo por lo que no tendrás que estar todo el tiempo escribiendo en la terminal.

# Comentarios

Los comentarios en java son bastante sencillos, para hacer comentarios de una sola línea solo tienes que colocar “//”, por ejemplo

//este es un comentario de una línea

Para hacer comentarios de varias líneas seguidas debes empezar a comentar con “/\*” y terminar de comentar con “\*/”, ejemplo

/\* Este es un comentario

De varias líneas \*/

# Variables

Las variables en java se declaran algo diferente a otros lenguajes de programación, primero debes colocar el tipo de variable y luego asignarla como por ejemplo.

int x = 50;

Siempre debes colocar punto y coma al terminar una línea de código, en este caso asignamos el numero 50 a x el cual dijimos que era un numero entero (int).

Int = entero

Double = decimal

Char = carácter individual

String = texto (comillas dobles)

Boolean = verdadero o falso

Para concatenar string se hace con el “+” entre strings

# Imprimir en pantalla…

Para imprimir en pantalla se realiza con el siguiente código,

System.out.println(“Mensaje a imprimir aquí”);

Se tiene que seguir al pie de la letra mayúsculas.

# Operadores matemáticos

Suma (+), resta (-), multiplicación (\*), división (/), modulo(resto)(%)

# Leer datos desde consola

Hay que importar **package source;** y luego **import java.util.Scanner;**

En el código -> **Scanner nombreVar = new Scanner(System.in);**

Imprimimos el mensaje con println luego con métodos, cada uno llama algo distinto, en este caso para un entero es:  **int x; x = nombreVar.nextInt();** luego lo puedes imprimir.

**Package source;** fue para “empaquetar” el archivo java por lo tanto ahora el archivo para compilarlo es

**source/NombreArchivo.java**  y luego para poder ejecutarlo  **java source.NombreArchivo**

# Archivos

Al momento de ejecutar un java también se le pueden mandar los datos que requiera por Scanner tan solo agregando un carácter en la terminal, por ejemplo digamos que el código pide un numero entero (**int x; x=Scanner.nextInt();** ) este le puede mandar atraves de un archivo de texto, imaginemos que el archivo  **numero.txt** contiene el número 15, solo se debe ejecutar de esta manera en la terminal:

**java NombreArchivo < numero.txt**