**Notas del curso de Java**

+++++ Videos del 1 al 10 +++++

\*\*\*Variables\*\*\*

Las variables guardan datos (importantes para la lógica de negocio)

Java es fuertemente tipado, a partir de la versión 10 permite crear variables con tipado dinámico

**Tipos de datos**



Primitivos (representan un valor, como un número un carácter un booleano)

De referencia como cadenas arreglos e interfaces, más complejas representa una entidad completa como objetos

Las interfaces determinan el comportamiento de objetos (contrato de implementación)

int dato =10; (Primitivo)

Integer dato = 10; (de referencia, clase integer)

-variables comienzan con minúscula

-clases comienzan con mayúscula

Nombre en Unicode (sin caracteres especiales)

Hardcode código en duro (acoplado al programa)

Hay clases en java como Integer o String que se le pueden asignar valores directamente y no es necesario crear una instancia (uso de new()) -> java lo hace internamente (por Debajo del agua)

---INTELLIJ: atajo sout (System.out.println()) soutv lo mismo, pero coloca la primera variable que encuentra---

Tipo primitivo -> se escribe en minúsculas

int primitivo

Integer clase (tiene métodos y propiedades)

===========================================================

int numeroSuperior = 100;

Contexto (scope) -> donde vive la variable

if(){

int numero = 10; (número vive dentro de if)

}

===========================================================

también las variables que se crearon en un contexto superior se pueden ver en el if

se pueden crear variables dinámicas a partir de Java 10

var numero3 = "10"; (se infiere el tipo de dato y le asigna un String)

INTELLIJ: cambiar el lenguaje

ALT + CTRL + S (abrir settings) > editor > Proffeading > Spelling > Dictionaries y agregamos el español

---Operador relacional --> devuelve true o false---

<

>

==

!=

--- ---

int edad\_persona= 10; (se puede declarar variable asi pero no se recomienda)

int edadPersona= 10; (esta es la convención camel case)

Las clases empiezan con mayúscula, los nombres de métodos siguen las mismas reglas que las variables

---comentarios:---

//

/\*\*/

+++++ Video 11 +++++

--- Tipos Primitivos ---

son tipos de datos escalares que contienen un solo valor

Por ejemplo:

-enteros (byte, short, intermedio (comung), long y en otros lenguajes super integer)

-flotante (float [precisión simple], y double [precisión doble])

-caracteres (char, comillas simples, usa el código UNICODE, 16 bits)

-booleanos (el mas simple un solo bit, verdadero o falso)

long numero = 4283746827364L; (L para identificar el Long)

float numero = 3.1416f; (f para identificar el float)

se pueden transformar de un primitivo a otro, pero hay que tener cuidado porque puede haber errores o perdida de información (por ejemplo de long a byte)

Todos los primitivos tienen un valor por defecto al definirse por ejemplo los booleanos empiezan en false (aplica a atributos de variables y no a variables local de un método)

los primitivos tienen clases equivalentes como: Byte, Character, Short...

+++++ Video 12 +++++

--- Primitivos numéricos enteros ---

Byte.BYTES (propiedad que muestra la cantidad de bytes que corresponden a la clase) para este caso es 1

Byte.SIZE (propiedad que muestra la cantidad de bites que corresponden a la clase) para este caso es 8

Byte.MAX\_VALUE (valor máximo de la clase en este caso 127)

Byte.MIN\_VALUE (valor máximo de la clase en este caso -128)

long numero = 628762872L; (se recomienda usar L mayuscula)

float numero = 62876.2872F; (se recomienda usar F mayuscula, presicion simple)

var con iteral numerica siempre se define como un integer normal (int)

var numero = 6287345345345346.2345345872D; (D de double, doble precisión)

+++++ Video 13 +++++

--- Primitivos numéricos de punto flotante ---

float realFloat = 2.12e3F //es exactamente igual a 212

con float se puede hacer uso de la notación cientifica (con exponentes positivos y negativos)

al imprimir valores muy grandes los muestra en notacion cientifica como (lo compacta):

float numero = 0.00000000015 (valor de punto fijo cuando no viene en notacion cientifica)

imprime --> 1.5e-10 1.5E-10 (1.5 mantisa, E exponente, -10 valor del exponente)

+++++ Video 14 +++++

--- Primitivos numéricos de punto flotante y sus constantes ---

lo sigiente aplica para Float y Double:

Float.BYTES (bytes correspondientes)

Float.SIZE (bites correspondientes)

Float.MAX\_VALUE (maximo valor)

Float.MIN\_VALUE (minimo valor)

Float.MAX\_EXPONENT (maximo exponente)

Float.MAX\_EXPONENT (minimo exponente)

si utilizamos var para crear la variable de punto flotante por defecto la determina como double

\*\*\*Variables de clase (REVISAR) si tienen valor por defecto

- te permiten acceder a ellas desde cualquier parte sin crear un objetos de el (deben ser estaticas)

+++++ Video 15 +++++

--- Primitivos char o caracter ---

-para caracter usar comillas simples

-esta basado en el estandar de comunicacion UNICODE

-en windows podemos ver el mapa de caracteres y el codigo UNICODE

-por ejemplo si viene U+0040 en java se coloca \u0040

se puede poner un decimal el cual lo conviernte en caracter

char decimal = 64; // imprime en consola @

de char su clase es Character

+++++ Video 16 +++++

--- Char y caracteres especiales ---

\u0020 --> espacio

\b --> backspace o retroceso

\t--> tabulador

\n --> salto de linea

\r --> retorno carro (se comporta diferente dependiento el SO) en Windows se ocupa \r\n los dos juntos

Si no sabemos cual salto de línea ocupar se puede hacer lo siguiente:

System.getProperty("línea.separator")

Con esto java toma de sus variables de ambiente el separador adecuado de acuerdo al SO

En donde line.separator variables de ambiente predefinidas en la máquina virtual de java

+++++ Vídeo 17 +++++

--- Primitivo boolean ---

Dato de lógica binaria solo true o false

\_Buscar valores primitivos y de referencia\_

Sirve para flujos de control como if

Se pueden asignar expresiones de relación o boleanas por ejemplo a < b

Si a es 3 y b es 5

float expresion = a < b retorna verdadero

O también expresion = (3-2==1); retorna true

+++++ Video 18 +++++

---Sistemas numéricos (binario, decimal octal, hexadecimal)---

Para transformar de decimal a cualquier otro sistema numérico se utilizan las clases de los primitivos como INTEGER FLOAT

usando el método toBinaryString(número) transforma un número decimal a una cadena del número en binario

Para decirle a java que estamos trabajando con un número binario se hace lo siguiente

int num=0b0111010101; se coloca 0b al inicio lo interpreta como binario pero en consola lo muestra como entero

Para los octales solo se antepone el cero

Número 0764 es octal

Si colocas 0x al incio lo detecta como hexadecimal

0x1f4

+++++ Video 19 +++++

---Sistemas numéricos y entrada de datos en ventana ---

Utilizar la librería java swing (api de ventana de java, api swing)

La cual contiene clases y a su ves estás tiene métodos estáticos

Por ejemplo la clase JOptionPane

Y uno de sus métodos showOptionPane(clase padre, mensaje)

Try y catch para el manejo de excepciones en java

+++++ Video 20 +++++

---Sistemas numéricos y entrada de datos en terminal ---

Se utiliza la clase scanner de java para obtener datos de entrada desde la terminal

Parámetros desde la terminal

Crear instancia de la clase Scanner

Scanner scanner= new Scanner (System.in);

System.in -> entrada de datos

//System.in entrada de la consola  
Scanner scanner = new Scanner(System.*in*);  
System.*out*.println("Ingrese un número entero: ");  
//scanner.nextLine() obtener la línea escrita por el usuario en la consola de Java  
//Con la clase scanner podemos obtener una cadena y transformarla directamente a un número con scanner.nextInt()  
//String numeroString = scanner.nextLine();  
int numeroDecimal = 0;  
try {  
 //numeroDecimal = Integer.parseInt(numeroString);  
 numeroDecimal = scanner.nextInt();  
} catch (InputMismatchException e) {  
 //Para este caso cambia el tipo de excepción ahora es: InputMismatchException la cual se debe importar  
 //también se puede atrapar excepciones de forma genérica con: Exception (la más genérica) todas las demás son hijas de esa clase  
 System.*out*.println("Debe ingresar un numero entero: ");  
 *main*(args);  
 //System.exit(0); si no se coloca esta línea el main se ejecuta recursivamente tantas veces como error detectado por el catch  
 System.*exit*(0);  
}

+++++ Video 21 +++++

---Conversión de cadenas a primitivos---