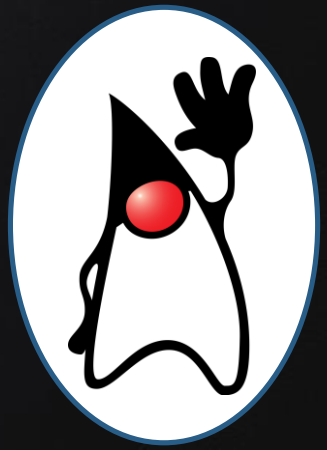
Ada

JAVA como lenguaje de programación



Objetivo general

Proporcionar familiaridad con los conceptos básicos de construcción de programas en JAVA.

Objetivos específicos

- Conocer sus conceptos generales y estructura de programa
- Describir identificadores, tipos de datos y sus operadores.
- Identificar y transcribir estructuras de DFD con las de JAVA.
- Describir procesos de entrada y salida

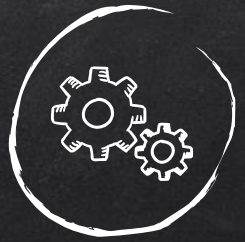
¿De qué hablaremos hoy?

- ✗ Caracteristicas generales
- ✗ Cómo funciona JAVA
- ✗ Qué tipos de datos tiene?
- ✗ Operadores
- ✗ DFD y su equivalencia en sentencias JAVA

Ada

1.

Caracteristicas de JAVA



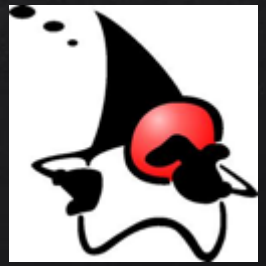
Las caracteristicas principales son

Fuente: https://www.manual-java.com/manualdejava/caracteristicas-de-java/

Ada

2.

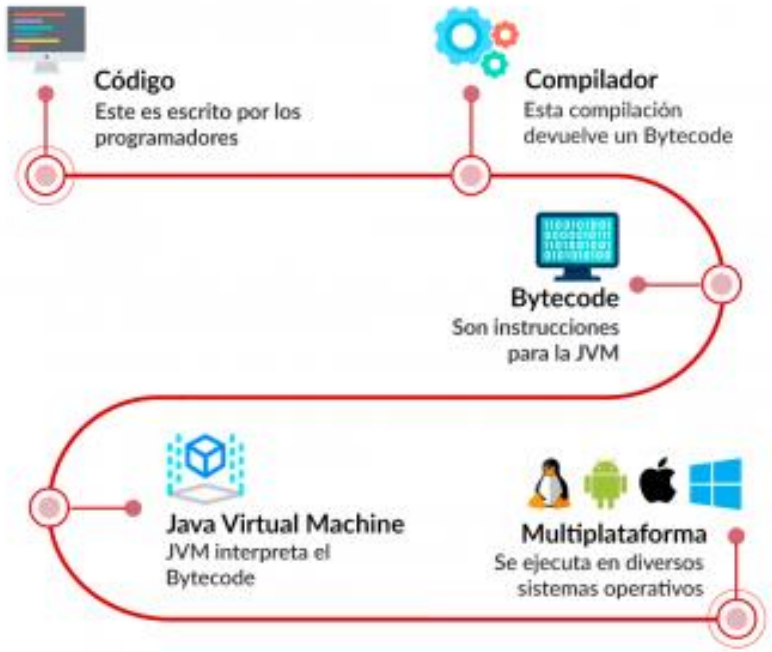
¿Cómo funciona JAVA ?



Una aplicación JAVA es compilada en una serie de BYTECODES que luego es procesada por la JAVA Virtual Machine(JVM) que interpreta y transforma las sentencias a lenguaje de máquina.

Fuente: https://es.wikipedia.org/wiki/Máquina_virtual_Java

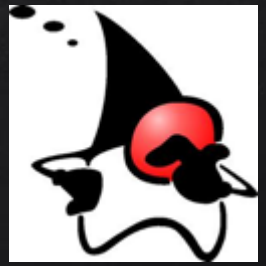
¿Cómo funciona Java?



Ada

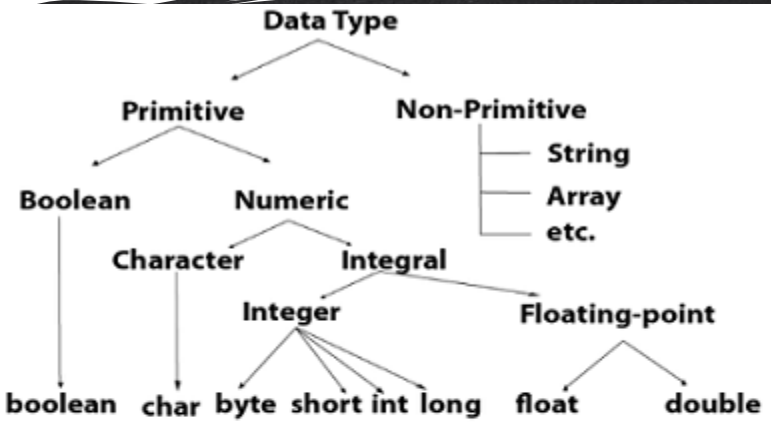
3.

¿Cuáles tipos de datos usa?



Tipos en JAVA

Java maneja 2 tipos de datos: primitivos y no primitivos. Los primitivos NO son objetos y son la estructura básica para manejar la informacion. Todos los demas, son objetos.



Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

Qué hago con los tipos de datos?

En su aplicación, va a ser muy importante definir cada una de las variables y sus tipos de datos. Para ello, cada variable debe tener un nombre autodocumentado y especifico al contexto. Es decir…

```
String n;
int e;
Date f;
double s;
```

```
String nombre;
int edad;
Date fechaNacimiento;
double sueldo;
```

Tomemos un Break!

Ada

4.

Operadores en Informática

Los operadores son símbolos que indican cómo se deben manipular los operandos. Los operadores junto con los operandos forman una expresión, que es una fórmula que define el cálculo de un valor. Los operandos pueden ser constantes, variables o llamadas a funciones, siempre que éstas devuelvan algún valor”

Precedence	Operator	Operand type	Description
1	++,	Arithmetic	Increment and decrement
1	+, -	Arithmetic	Unary plus and minus
1	~	Integral	Bitwise complement
1	!	Boolean	Logical complement
1	(*type*)	Any	Cast
2	*, /, %	Arithmetic	Multiplication, division, remainder
3	+, -	Arithmetic	Addition and subtraction
3	+	String	String concatenation
4	<<	Integral	Left shift
4	>>	Integral	Right shift with sign extension
4	>>>	Integral	Right shift with no extension
5	<, <=, >, >=	Arithmetic	Numeric comparison
5	instanceof	Object	Type comparison
6	==, !=	Primitive	Equality and inequality of value
6	==, !=	Object	Equality and inequality of reference
7	&	Integral	Bitwise AND
7	&	Boolean	Boolean AND
8	^	Integral	Bitwise XOR
8	^	Boolean	Boolean XOR
9	|	Integral	Bitwise OR
9	|	Boolean	Boolean OR
10	&&	Boolean	Conditional AND
11	||	Boolean	Conditional OR
12	?:	N/A	Conditional ternary operator
13	=	Any	Assignment

Ada

!=

Detalle completo en: https://javadesdecero.es/basico/operadores-en-java-ejemplos

5.

DFD y su equivalencia en sentencias JAVA

DFD y código JAVA

Durante la cursada hemos hecho diferentes dibujos en DFD, que pueden ser pasados directamente a su equivalente sentencia JAVA.

Una sentencia JAVA, es una “instrución” que tendra una “sintaxis y semantica” especifica segun el lenguaje de programación.

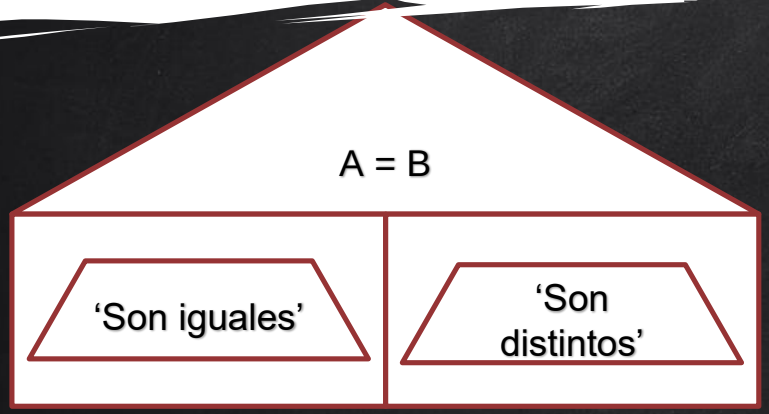
C	`public static void main(String[] args) {`

A = 0	`int a;`
	
	`a = 0;`

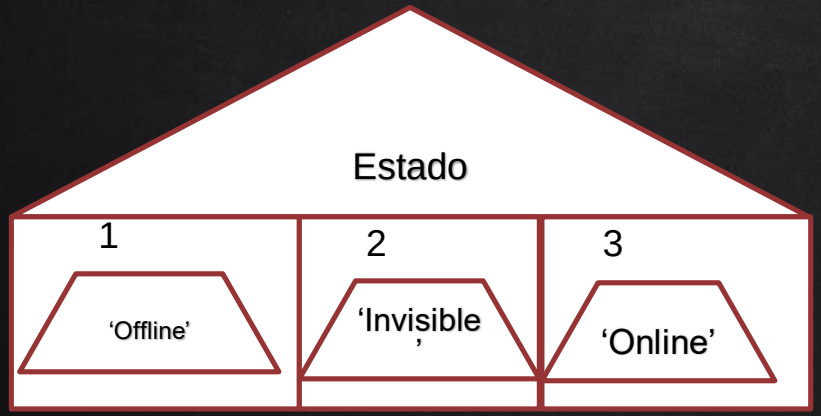
C = A + B	`int a, b, c;`
	`a = 3;`
	`b=1;`
	`c = a + b;`

EDAD	`int edad;`
	
	`Scanner Entrada = new Scanner(System.in);`
	
	`edad = Entrada.nextInt();`

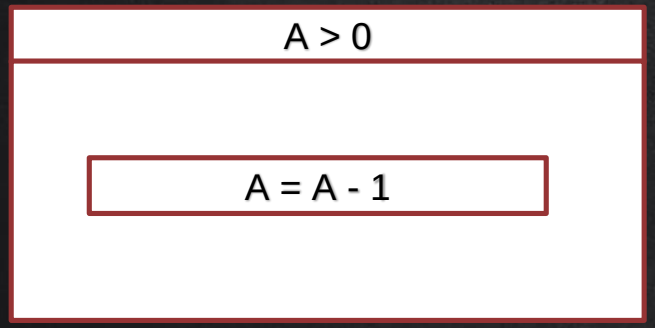
'La edad es', EDAD	`System.out.println("La edad es " + edad);`
F	`}`



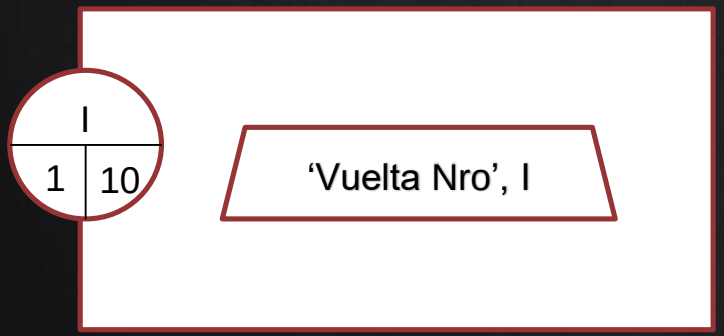

```java
if (a == b) {
    System.out.println("Son Iguales");
}
else{
    System.out.println("Son distintos");
}
```




```java
switch (estado) {
    case 1:
    System.out.println("Offline");
    break;
    case 2:
    System.out.println("Invisible");
    break;
    case 3:
    System.out.println("Online");
    break;
}
```




```
while (a > 0) {
    a = a - 1;
}
```




```
for (int i = 1; i <= 10; i++) {
    System.out.println("Vuelta Nro " + i);
}
```

Y qué hacemos con todo esto?!

Let's Rock
&
Code!!!!!!!!

Gracias!

Preguntas?

Ada