Palabras reservadas

- Import IMPORT
- public PUBLIC
- privated PRIVATED
- protected PROTECTED
- class CLASS
- int INT
- boolean BOOLEAN
- String STRING
- double DOUBLE
- char CHAR
- Object (Nombre de clases) OBJECT
- if IF
- else EISE
- for FOR
- while WHILE
- do while DO_WHILE
- switch SWITCH
- final FINAL
- Break BREAK
- Return RETURN

Signos de puntuación

;	P_COMA
.	PUNTO
,	COMA

Signos de agrupación

{	LLAVE_A
}	LLAVE_C
(	PARENTESIS_A
)	PARENTESIS_C

Operadores aritmeticos

+	MAS
−	MENOS
*	POR
/	DIVISION

Operadores relacionales

= ASIGNAR
> MAYOR_QUE
< MENOR_QUE
>= MAYOR_IGUAL
<= MENOR_IGUAL
!= DIFERENTE
== IGUAL

operadores logicos

&& AND
|| OR
! NOT

Operadores incremento / decremento

++ INCREMENTO
-- DECREMENTO

Comentarios

//

/* */

Estructuras

* Imports: import java.util.*;

  IMPORT ID PUNTO ID PUNTO POR P_COMA
  IMPORT ID PUNTO ID PUUTO ID P_COMA

* clases: Public class Gato { sentencias }

  visibilidad CLASS ID LLAVE_A sentencias LLAVE_C

* Metodos y funciones: privated String hola(parametros){sentencias}

  visibilidad tipo ID PARENTESIS_A parametros PARENTESIS_C
  LLAVE_A sentencias LLAVE_C

* Declaraciones de variables: public int miEntero;

  visibilidad tipo ID P_COMA

* Asignaciones: public int miEntero = 5; miEntero = 3;

  visibilidad tipo ID ASIGNACION ENTERO P_COMA
  ID ASIGNACION ENTERO P_COMA

* If: if(condicion){sentencias}

  IF PARENTESIS_A condicion PARENTESIS_C LLAVE_A sentencias
  LLAVE_C

* If else: if(condición){sentencias} else{sentencias}

  IF PARENTESIS_A condición PARENTESIS_C LLAVE_A sentencias
  LLAVE_C ELSE LLAVE_A sentencias LLAVE_C

*for: for(int i=0; i<70; i++){ sentencias }

FOR PARENTESIS_A tipo ID ASIGNACION P_COMA
ID relacional ENTERO P_COMA ID INCREMENTO
PARENTESIS_C LLAVE_A sentencias LLAVE_C

*while: while(condición){ sentencias }

WHILE PARENTESIS_A condicion PARENTESIS_C LLAVE_A
sentencias LLAVE_C

*do while: do{ sentencias } while(condición);

DO LLAVE_A sentencias LLAVE_C WHILE PARENTESIS_A
condicion PARENTESIS_C P_COMA

*switch: switch(condición){ case valor: sent break case valor: sent
default: sent }

SWITCH PARENTESIS_A condición PARENTESIS_C LLAVE_A
casos y default LLAVE_C

*Constructor: public Miclase(parametros){ }

visibilidad ID PARENTESIS_A parametros PARENTESIS_C
LLAVE_A sentencias LLAVE_C

Llamadas a funciones: mifun(parametros);

ID PARENTESIS_A parametros PARENTESIS_C P_COMA

*Parametro: string hola

tipo ID