

# Programación orientada a objetos

---

SINTAXIS DE JAVA

A solid green horizontal bar at the bottom of the slide.

# Definición de clases

---

Declaración de clases	<code>&lt;modificador de acceso&gt; class Nombre { ... }</code>	<code>public class ClaseUno { ... }</code>

- Modificadores de acceso:
  - default
  - **public**
  - **private**
  - protected

# Sintaxis general

---

Comentarios	//comentario de una línea /* comentario multilínea*/ /**comentario de documentación automática*/	//comentario de una línea /* comentario multilínea*/ /**comentario de documentación automática*/
Declaración de variables	tipo nombre1, nombre2	int a; int b, c; int a=2;
Declaración de constantes	final tipo nombre1, nombre2	final int CONSTANTE_UNO = 100;

# Tipos de datos (I)

---

Entero	byte [-128 a 127] short [-32768 a 32767] int [-2147483648 a 2147483647]  long [-9223372036854775808 a 9223372036854775807]	byte variable; short variable; int variable; long variable; int i1,i2;
Coma flotante	float [32 bits, precisión simple; 3,4E-38 a 3,4E38] double [64 bits, precisión doble; 1,7E-308 a 1,7E308]	double dvar=3.1416; float fvar=3.1416f;
Booleano	true [valor cierto] false [valor falso]	boolean encontrado=false;

# Tipos de datos (II)

---

Carácter	char [carácter alfanumérico]	<pre>char car1 = 'A'; int i1 = 'q'; char c1 = 99;</pre>
Vectores (arrays) y matrices	array [puede ser de varias dimensiones]	<pre>char c[]; char[] c; char c[]=new char[10]; c[3] = 'b'; int i[][]=new int[10][15]</pre>

# Caracteres especiales

---

Caracteres Especiales	<code>\n</code> [nueva línea] <code>\t</code> [tabulador horizontal] <code>\b</code> [retroceso] <code>\r</code> [retorno de carro] <code>\f</code> [salto de página] <code>\\</code> [carácter \] <code>\'</code> [comilla simple] <code>\"</code> [comilla doble] <code>\udddd</code> [carácter unicode] (d={0..F}) <code>\ddd</code> [carácter ASCII] (d={0..9})	<code>System.out.print ("final\n");</code>

# Operadores (I)

---

Aritméticos	+ [adición] (también concatena cadenas de caracteres). - [sustracción] * [multiplicación] / [división] % [resto] ++ [incremento] -- [decremento]	<pre>int suma = a + b; int resta = c - d; suma++; resta--;</pre>
Relacionales	> [mayor que] >= [mayor o igual que] < [menor que] <= [menor o igual que] == [igual a] != [distinto de]	<pre>boolean verdad = 6 &gt; 2;</pre>

# Operadores (II)

---

Lógicos	&& [ambos ciertos]    [cierto al menos uno] ! [negación]	(a && b) es cierto si a y b son ciertos
Asignación con operación	operador +=, -=, *=, /=, %=	a += b; // a=a+b;
Conversión de tipos	(tipo) variable o expresión	a=(int)b/c;
Condicional	expresión booleana ? res1: res2	x=1; y=10; z=(x>y)?x+3:y+8;



# Estructuras de control (I)

---

Bucles for	<pre>for( [type] var=min; var &lt; max; inc ) {     // acciones }</pre>	<pre>for(int i=0; i &lt; 8; i++) {     System.out.println(i); }</pre>
Bucles while	<pre>while( condición ) {     // acciones }</pre>	<pre>while( i &lt; 8 ) {     System.out.println(i);    i++ }</pre>
Bucles do-while	<pre>do {     // acciones } while( condición );</pre>	<pre>do {    System.out.println(i);    i++; } while (i &lt; 8);</pre>

# Estructuras de control (II)

---

if / else	<pre>if( condición ) {     // acciones } [else {     // acciones }]</pre>	<pre>if( i == 0 ) {     System.out.println("caso 0"); }else if( i==1) {     System.out.println("caso 1"); } else {      System.out.println("caso otro"); }</pre>
switch	<pre>switch( variable ) {     case n1:         // acciones         break;     // otros case     [default:         // acciones alternativas] }</pre>	<pre>switch( i ) {     case 0:  System.out.println("cero");     break;     case 1:  System.out.println("uno");     break;     default: System.out.println("&gt; otro"); }</pre>

# Identificadores (I)

---

- Distinción entre mayúsculas y minúsculas →
  - identificadores como var1, Var1 y VAR1 son distintos.
- Caracteres del código Unicode →
  - se pueden declarar variables con el nombre *añoDeCreación* o *raïm*.
  - Aunque IntelliJ Idea te indica una advertencia para no usar caracteres no ascii.
- El primer carácter debe ser:
  - una letra,
  - el símbolo \_ (subrayado o guion bajo)
  - el símbolo \$ (dólar).

# Identificadores (II)

---

- En el identificador no pueden aparecer espacios en blanco ni símbolos coincidentes con operadores del lenguaje
- No hay longitud máxima.
- No pueden coincidir con una palabra reservada del lenguaje ni con los valores lógicos *true* o *false*.
- No pueden ser iguales a otro identificador declarado en el mismo ámbito.

# Palabras reservadas

---

abstract	continue	finally	int	public	throw
assert	default	float	interface	return	throws
boolean	do	for	long	short	transient
break	double	goto	native	static	true
byte	else	if	new	strictfp	try
case	enum	implements	null	super	void
catch	extends	import	package	switch	volatile
class	false	inner	private	synchronized	
const	final	instanceof	protected	this	while

# Recomendaciones de estilo

---

- Nombre de variables: `lowerCamelCase`
- Nombre de constantes: en mayúsculas y las palabras separadas con `_`.
- Nombre de clases: `UpperCamelCase`
- Nombre de métodos: `lowerCamelCase`
- Tamaño de líneas: 100 caracteres, una sentencia por línea.
- Tamaño de métodos: 20, 65, 80 líneas.
- Niveles de sentencias de control anidadas: 3, 4, 5.

# Enlaces de interés

---

Guía de estilo de google:

<https://google.github.io/styleguide/javaguide.html>