



UCA

Programación Orientada a Objetos I

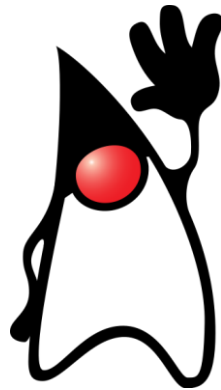
JAVA – ELEMENTOS BÁSICOS (I)

Generalidades

- Las instrucciones siempre terminan con punto y coma → ;
- Los bloques de instrucciones se encierran entre llaves → { }
- Los comentarios de una sola línea se anteceden a doble barra → //
- Los comentarios de más de una línea se encierran → /* ... */

Duke →

Mascota de Java



```
1 public class App {  
2     /* Este es el típico Hola Mundo que usamos siempre que comenzamos  
3     con un nuevo lenguaje para conocer su estructura mínima  
4     */  
5     public static void main(String[] args) throws Exception {  
6         // Salida por terminal o consola de un texto  
7         System.out.println("Hello, World!");  
8     }  
9 }  
10
```

Identificadores

- Nombran variables, funciones, clases y objetos.
- Comienzan con cualquier letra, un subrayado (_) o el símbolo de pesos (\$).
- Distinguen las mayúsculas de las minúsculas.
- No poseen longitud máxima.

Identificadores

Convenciones para los nombres:

- **En general:**
 - Que sean descriptivos
 - Evitar nombres muy largos
 - Para nombres multipalabra, juntarlas y coloca la primera letra de cada palabra en mayúscula (camelCase)
- **Clases:**
 - Primera letra en mayúscula y luego en minúsculas
 - Usar sustantivos en singular
- **Métodos miembro:**
 - Verbos en infinitivo o verbo más sustantivo
- **Constantes:**
 - Letras en mayúsculas
 - Separación de palabras con guion bajo (_)

¿Esto es mucho
muy importante!



Palabras Reservadas

abstract
boolean
break
byte
byvalue
case
cast
catch
char
class
const
continue

default
do
double
else
extends
false
final
finally
float
for
future
generic

goto
if
implements
import
inner
instanceof
int
interface
long
native
new
null

operator
outer
package
private
protected
public
rest
return
short
static
super
switch

synchronized
this
threadsafe
throw
transient
true
try
var
void
while

Variables

Java es un lenguaje orientado a objetos, y por eso es que no tiene variables globales.

- **Variables de instancias o de objetos:**

Son las que están definidas dentro de un objeto y no son estáticas. Su visibilidad depende del modificador.

- **Variables de clases (estáticas):**

Precedidas por la palabra “static”, son variables de la cual solo existe una sola instancia. Por defecto públicas, pero puede modificarse.

- **Variables locales:**

Son aquellas que solo son visibles dentro del método donde fueron definidas. No llevan visibilidad

- **Parámetros:**

Son aquellas que se reciben como parámetros de los métodos. Solo visibles dentro del método.

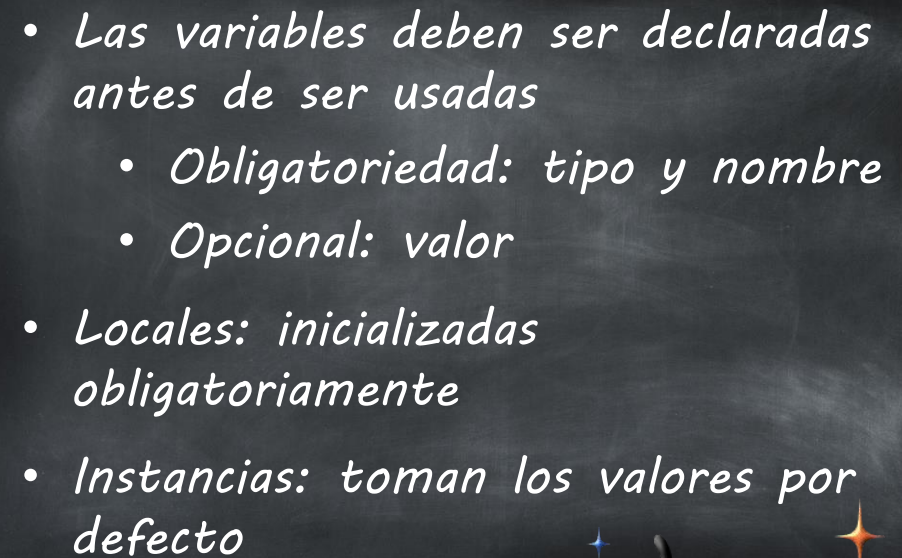
Variables

Visibilidad de variables (y métodos):

- public: visibles desde cualquier lugar
- private: visible solo desde la misma clase
- protected: visible desde dentro de la misma clase, subclases y clases del mismo paquete

Tipos de variables:

- Tipos primitivos de datos
- Variables de referencia
 - Vectores (arrays)
 - Clases (objetos – instancias)

- 
- *Las variables deben ser declaradas antes de ser usadas*
 - *Obligatoriedad: tipo y nombre*
 - *Opcional: valor*
 - *Locales: inicializadas obligatoriamente*
 - *Instancias: toman los valores por defecto*

Tipos primitivos

Definidos en el lenguaje con tamaño fijo.

- Enteros: todos tienen bit de signo
- Reales: puntos flotantes expresados en IEEE
- Caracteres: expresados como un entero
- Booleanos: no son enteros, sino true/false

Tipos Datos	Tamaño (bits)	Valor inicial	Rango
byte	8	0	-2^8 a 2^8-1 -128 a 127
short	16	0	-2^{16} a $2^{16}-1$ -32768 a 32767
int	32	0	-2^{32} a $2^{32}-1$ 2.147.483.648 a 2.147.483.64
long	64	0	-2^{64} a $2^{64}-1$ $-9*10^{18}$ a $9*10^{18}$
float	32	0.0F	Simple precisión $-3.4*10^{38}$ a $3.4*10^{38}$
double	64	0.0D	Doble precisión $-1.79*10^{308}$ a $1.79*10^{308}$
char	16	\0	ASCII Unicode
boolean	8	false	

Ejemplos

```
byte i;  
short depto, piso;  
int edad, cant = 1000;  
long superficie;  
float precio=10.6f, promedio = 3.25f;  
double peso=15450.834;  
char letra='a', Letra;           //comillas simples  
boolean esVerdad, esMentira=true;
```

Vectores

- Se pueden declarar vectores o array de cualquier tipo primitivo, incluso array de array (matrices). Ejemplos:
 - `char s[];`
 - `int iArray[], a[][];`
- Los vectores se consideran *objetos*, que tienen métodos asociados, como por ejemplo *length* (longitud del array).
- Se dimensionan en tiempo de ejecución. Ejemplo:
 - `int a[][] = new int[10][3];` // matriz de enteros (10x3)
 - `a.length;` // igual a 10
 - `a[0].length;` // igual a 3

Tipos de clases

- Se utilizan para crear objetos o instancias de una clase.

- Ejemplos de la clase *String*

```
String titulo;
```

```
String apellido, nombre="Christian";
```

```
String apellido="Lopez", nombre="Christian";
```

