final

El atributo "final" puede ser aplicado tanto a clases como a sus campos y métodos. Indica que una vez definido el elemento no puede volverse a definir:

- •para clases, no pueden tener subclases
- •para variables, no puede alterarse (constantes, pero si predefinición)
- •para métodos, no pueden ser redefinidos en una subclase

Palabras reservadas en Java				
abstract	assert(1.4)	boolean	break	byte
case	catch	char	class	const*
continue	default	do	double	else
enum(5.0)	extends	final	finally	float
for	goto*	if	implements	import
instanceof	int	interface	long	native
new	package	private	protected	public
return	short	static	strictfp(1.2)	super
switch	synchronized	this	throw	throws
transient	try	void	volatile	while

static

"static" tiene dos utilidades

- •actuar como atributo aplicable a cualquier campo o método.
  - •para campos: residirán en la estructura de la clase
  - •para métodos: no están ligados a objetos (se pueden invocar a través de la clase)

```
public class ClaseA {
     private static int cont=0;
     public static inc() {cont++;}
     public static dec() {cont--;}
     // resto de la definición de clase
}

//desde otro punto cualquiera…
ClaseA.inc()                  //incrementa el contador
//o a través de cualquier objeto de clase ClaseA
ClaseA objA=new ClaseA();     //Un objeto de clase ClaseA
objA.inc();                   //incrementa el contador de clase
```

- •inicializar la clase.

```
public class ClaseA {
     private static int cont;
     public static inc() {cont++;}
     public static dec() {cont--;}
     static {
        cont=ClaseB.valorBase;
     }
     // resto de la definición de clase
}
```

```
1-  //
2-  // Aplicación ejemplo "HolaMundo"
3-  //
4-
5-  public class HolaMundo {
6-          public static void main(String[] args) {
7-                  System.out.println("Hola, mundo");
8-          }
9-  }
```