



CASTING



El casting es un procedimiento para transformar una variable primitiva de un tipo a otro, o transformar un objeto de una clase a otra clase siempre y cuando haya una relación de herencia entre ambas (este último casting es el más importante y se verá más adelante).

Dentro del casting de variables primitivas se distinguen dos clases:

Implícito: no se necesita escribir código para que se lleve a cabo. Ocurre cuando se realiza una conversión ancha (widening casting), es decir, cuando se coloca un valor pequeño en un contenedor grande.

Explícito: sí es necesario escribir código. Ocurre cuando se realiza una conversión estrecha (narrowing casting), es decir, cuando se coloca un valor grande en un contenedor pequeño. Son susceptibles de pérdida de datos.

Detallaremos estas clases:



Casting implícito

En este caso no se necesita escribir código para que la conversión se lleve a cabo. Ocurre cuando se realiza lo que se llama una conversión ancha (widening casting), es decir, cuando se coloca un valor pequeño en un contenedor grande.

Ejemplo de casting explícitos serían:

```
int num1 = 100;

long num2 = num1;    // Un int cabe en un long

long num2 = 100;    // 100 en un int
```

Casting explícito

En el casting explícito sí es necesario escribir código. Ocurre cuando se realiza una conversión estrecha (narrowing casting), es decir, cuando se coloca un valor grande en un contenedor pequeño. Son susceptibles de pérdida de datos.

```
int num1 = 100;

short num2 = (short) num1;    // Aquí hace falta un casting
                               // explícito: short tiene menor rango que int
```

Como se ve, el formato general para indicar que queremos realizar la conversión es:

(tipo) valor_a_convertir

En este ejemplo, si se sustituye la primera línea `int num1=100` por `int num1=1000000`, el código compilaría bien, pero habría pérdida de datos, pues el 1000000 se sale del rango de `short`, que comprende desde -32768 a 32767. Al mostrar por consola el valor se obtendría un resultado incongruente.

```
double num1 = 25.5;

float num2 = (float) num1;

float num3 = 17.25;
```



En este ejemplo se hace un casting explícito en la segunda línea porque float es de menor rango que double. En la tercera línea el compilador avisaría de que hay un error: cuando escribimos directamente un número en decimal (como aquí el 17.25), Java lo interpreta siempre como un double; por eso sería necesario un casting explícito para que compilase:

```
float num3 = (float) num3;
```

Para finalizar con el casting entre primitivas, conviene tener en cuenta lo siguiente:

No es posible realizar casting entre una variable primitiva booleana y cualquier otra variable primitiva.

Sí es posible realizar casting entre una variable primitiva char y una variable primitiva que almacene enteros:

```
int num1 = 164;  
char letra = (char) num1;  
System.out.println(letra);  
System.out.println((char) 164);
```

Siendo la salida del programa:

ñ

ñ

La siguiente tabla resume las posibilidades de casting existentes:

Convertir desde		Convertir a...						
	boolean	byte	short	char	int	long	float	double



boolean		no	no	no	no	no	no	no
byte	no		si	cast	si	si	si	si
short	no	cast		cast	si	si	si	si
char	no	cast	cast		si	si	si	si
int	no	cast	cast	cast		si	si*	si*
long	no	cast	cast	cast	cast		si*	si*
float	no	cast	cast	cast	cast	cast		si
double	no	cast	cast	cast	cast	cast	cast	

Donde:

- no: indica que no hay posibilidad de conversión.
- sí: indica que el casting es implícito.
- sí*: indica que el casting es implícito, pero se puede producir pérdida de precisión.
- cast: indica que hay que hacer casting explícito.