

# CONVERSIÓN DE TIPOS

## VARIABLES Y OPERADORES

# CONVERSIÓN DE TIPO DE DATO

- En ocasiones nos puede interesar convertir el tipo de dato de una variable en otro.
- También se le conoce como **casting**.
- Dos tipos:
  - Conversión **automática** o implícita.
  - Conversión manual o **explícita**.

# CONVERSIÓN AUTOMÁTICA

- Java es capaz de realizar una conversión automática entre tipos compatibles.
  - Se conocen como **conversiones primitivas de ampliación** (Widening primitive conversions).
- Por ejemplo:
  - Podemos almacenar un valor *int* en una variable *long* sin pérdida de información.

```
int i = 1234567;
```

```
long l = i; // No hay pérdida de información
```

# CONVERSIÓN AUTOMÁTICA

- *byte* a *short*, *int*, *long*, *float*, o *double*
- *short* a *int*, *long*, *float*, o *double*
- *char* a *int*, *long*, *float*, o *double*
- *int* a *long*, *float*, o *double*
- *long* a *float* o *double*
- *float* a *double*

**¡OJO! Casos especiales con *int* o *long* hacia *float* o *double*.**

# CONVERSIÓN AUTOMÁTICA

- Si convertimos *int* → *float*, *long* → *float* o *long* → *double* podemos **perder precisión**, y obtendremos un **número redondeado**.

```
long l2 = 123_456_789_123_456l;
```

```
System.out.println(l2);
```

```
float f2 = l2;
```

```
System.out.printf("%.2f",f2);
```

```
System.out.println();
```

Así podemos manejar la cantidad de dígitos a mostrar

# CONVERSIÓN EXPLÍCITA

- **Obligatoria** cuando los tipos **no** son **compatibles**.
- Es una conversión **forzada**.
- Se puede **perder información** en muchos casos.
- Se puede **provocar un error** si el número a convertir excede el máximo del tipo de dato de destino.

# CONVERSIÓN EXPLÍCITA

- Se realiza indicando el tipo al que queremos convertir entre **paréntesis**.
- Se indica a la **izquierda del valor** que queremos transformar.
- Ejemplo: de *int* a *short*.

*int i3 = 1234;*

*short s3 = (short) i3;*