



Taller 4 - Java -2

<input checked="" type="checkbox"/> Completado	<input type="checkbox"/>
Cursos	<u>POO</u>

1. El código nos da la siguiente salida

```
char : Entra a int: 103
short : Entra a int: 2
byte : Entra a int: 1
long : Entra a double: 9.99999999E8
integer : Entra a int: 51232
double : Entra a double: 12.4
float : Entra a double: 5.6500000095367432
```

¿A qué se debe? Se debe una sobrecarga en el método funcion, así que la salida depende de el dato que ingresemos, para esto debemos tener en cuenta la longitud bytes de los primitivos en orden ascendente:

- bool
- byte
- short
- char → va más arriba por ser Unicode
- int
- long
- float
- double

Ahora hechemos un vistazo a nuestras entradas.

```
char c = 'g';
short s = 2;
```

```
byte b = 1;
long l = 999999999;
int i = 51232;
double d= 12.4;
float f = 5.65f;
```

Y como lo imprimos

```
System.out.println("char : " + funcion(c));
System.out.println("short : " + funcion(s));
System.out.println("byte : " + funcion(b));
System.out.println("long : " + funcion(l));
System.out.println("integer : " + funcion(i));
System.out.println("double : " + funcion(d));
System.out.println("float : " + funcion(f));
```

Y nuestro método Super cargado:

```
static String funcion (int a) {
    return "Entra a int: " + a;
}

static String funcion (double a) {
    return "Entra a double: " + a;
}
```

Ya con esto podemos dar claramente el porque:

- a. Para el primer dato (c), es un char, su inmediato mayor es el int, entonces allí entra, el valor 103, es el código ascii del caracter c.
- b. Para el segundo dato (s), es un short, su inmediato mayor es el int, entonces allí entra el valor.
- c. Para el tercer dato (b), es un byte, su inmediato mayor es el int por lo que allí entra el valor.
- d. Para el cuarto dato (l), es un long, su inmediato mayor es el double, por lo que entra allí. El valor se muestra en notación científica debido

a su magnitud.

- e. Para el quinto dato (i), es un int, por lo que entra directamente en la función int sin necesidad de promoción.
- f. Para el sexto dato (d), es double entonces entra directamente.
- g. Para el séptimo (f), es float, su inmediato mayor es double por lo cual entra en el segundo metodo.

2. Hagamos la comprobaciones:

- a. Activando la funcion de short nuestra salida es la siguiente

```
char : Entra a int: 103
short : Entra a short: 2
byte : Entra a short: 1
long : Entra a double: 9.99999999E8
integer : Entra a int: 51232
double : Entra a double: 12.4
float : Entra a double: 5.650000095367432
```

Esto genera ya que como ahora tenemos una funcion para short, el inmediata mayor de los valores byte(b) y equivalente de short(s) pasan a entrar a esta.

- b. Activando la de floats se genera la salida:

```
char : Entra a int: 103
short : Entra a short: 2
byte : Entra a short: 1
long : Entra a float: 1.0E9
integer : Entra a int: 51232
double : Entra a double: 12.4
float : Entra a float: 5.65
```

Ahora los datos que entraban de double siendo promovidos pasan a ser de float ya que este es el inmediatamente menor, ahora double como es un valor más grande (hablando de bytes) y como ya tiene un método inmediato se que da en double.

- c. Ahora lo que recibimos es un error de compilación, no tenemos función para valores double, y un valor más grande no puede ser

depreciado ya que perdería valor (por eso ninguno valor disminuye su primitivo, comprometería una pérdida de información).

d. Si comentamos todo y dejamos double nuestra salida será:

```
char : Entra a double: 103.0
short : Entra a double: 2.0
byte : Entra a double: 1.0
long : Entra a double: 9.99999999E8
integer : Entra a double: 51232.0
double : Entra a double: 12.4
float : Entra a double: 5.6500000095367432
```

Ahora todo entra a double debido a que es el primitivo más grande de todos, entonces todos los valores pueden entrar a este y como no tenemos mas métodos que cambien esto...