



Calculadora

Implementar, exclusivamente mediante Java, un programa que haga las funciones de una calculadora con las cuatro operaciones básicas de suma, resta, multiplicación y división entre números enteros (positivos o negativos).

Para resolver este ejercicio debe enviarse un único fichero llamado `calc.zip` que contenga uno o mas ficheros `.java`, que deben situarse en el directorio raíz, (no utilizar paquetes ni subdirectorios) que se compilarán mediante la instrucción:

```
javac Calculadora.java
```

El programa resultante tendrá su entrada desde un fichero de pruebas y su salida se redirigirá a otro fichero.

Por ejemplo, la instrucción `java Calculadora exp.in exp.out` transforma el fichero `exp.in` en el fichero `exp.out`.

- Si no se especifica un fichero de salida, se utilizará la salida estándar `System.out`.
- Si tampoco se especifica uno de entrada, se usará la entrada estándar `System.in`.

La solución se evaluará de acuerdo a los casos de prueba descritos a continuación:

▼ **Prueba 0 - Compilación**

▼ **Prueba 1 - Operaciones básicas**

- Un operador binario y números positivos.

exp.in	exp.out
2+2	4
3*5	15
12*400	4800

▼ **Prueba 2 - Operaciones básicas**

- Un operador binario, ignorando espacios en blanco y tabuladores.
- Las líneas en blanco no generan ninguna salida.
- Las líneas que contienen un solo número dan como resultado ese número.

exp.in	exp.out
21 + 12	33
11 *22	242
5 - 1	4
123	123

▼ Prueba 3 - Operaciones básicas

- Cualquier número de operadores con operandos enteros, positivos o negativos, respetando la prioridad habitual entre los operadores.

exp.in	exp.out
10 + 20 * 30	610
10* 20+ 30	230
30 * 20 +10	610
1*2+3*4+5*6+7*8	100

▼ Prueba 4 - Incluir el operador «menos» unario

exp.in	exp.out
1+-2--3*-4	-13
-11+22-0+4*-1	7
2+32/2/2/2	6
- - - - 3	3
64/-2/2/2/-2*-2*-2	16

▼ Prueba 5 - Incluir paréntesis en las expresiones

exp.in	exp.out
$(1+-2)-(-3*-4)$	-13
$2 * ((-11+22-0)+(4*-1))$	14
$2 + (((32/2)/2)/2)$	6
$- (- - - 3 * (2 + 5))$	21
$(64/-2/2/2/-2)-(2*-2*-2)$	-4
$(((((1+2)*3)+4)*((5+6)*7)+8)*9)$	9081