

PRÁCTICAS JUNIT

Ejercicio 1: Crear una clase Calculadora con dos atributos. Realizar sus operaciones más básicas, es decir, suma, resta, multiplicación y división, y realizar las pruebas correspondientes para cada operación.

Ejercicio 2: Crear una clase Salario que tendrá dos métodos, además de sus métodos correspondientes para crear los objetos:

- float calculaSalarioBruto(String tipo, float ventasMes, float horasExtra)

El salario base será 1000 euros si el empleado es de tipo vendedor, y de 1500 euros si es de tipo encargado. A esta cantidad se le sumará una prima de 100 euros si ventasMes es mayor o igual que 1000 euros, y de 200 euros si fuese al menos de 1500 euros. Por último, cada hora extra se pagará a 20 euros.

- float calculaSalarioNeto(float salarioBruto)

Si el salario bruto es menor de 1000 euros, no se aplicará ninguna retención. Para salarios a partir de 1000 euros, y menores de 1500 euros se les aplicará un 16%, y a los salarios a partir de 1500 euros se les aplicará un 18%.

Deberéis de realizar pruebas de estos métodos en JUnit, las cuales serán (obviando aquellas que devuelven una BRException:

Método a probar	Entrada	Salida esperada
calculaSalarioNeto	2000	1640
calculaSalarioNeto	1500	1230
calculaSalarioNeto	1499.99	1259.9916
calculaSalarioNeto	1250	1050
calculaSalarioNeto	1000	840
calculaSalarioNeto	999.99	999.99
calculaSalarioNeto	500	500
calculaSalarioNeto	0	0
calculaSalarioNeto	-1	BRException
calculaSalarioBruto	vendedor, 2000 euros, 8h	1360
calculaSalarioBruto	vendedor, 1500 euros, 3h	1260
calculaSalarioBruto	vendedor, 1499.99 euros, 0h	1100
calculaSalarioBruto	encargado, 1250 euros, 8h	1760
calculaSalarioBruto	encargado, 1000 euros, 0h	1600
calculaSalarioBruto	encargado, 999.99 euros, 3h	1560
calculaSalarioBruto	encargado, 500 euros, 0h	1500
calculaSalarioBruto	encargado, 0 euros, 8h	1660
calculaSalarioBruto	vendedor, -1 euros, 8h	BRException
calculaSalarioBruto	vendedor, 1500 euros, -1h	BRException
calculaSalarioBruto	null, 1500 euros, 8h	BRException

Ejercicio 3:

Crea una clase llamada Cuenta que tendrá los siguientes atributos: titular y cantidad (puede tener decimales).

Crea dos constructores que cumpla lo anterior.

Crea sus métodos get, set y toString.

Tendrá dos métodos especiales:

- ingresar(double cantidad): se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
- retirar(double cantidad): se retira una cantidad a la cuenta, si restando la cantidad actual a la que nos pasan es negativa, la cantidad de la cuenta pasa a ser 0.

Cread las pruebas unitarias de esta clase.