

Ejercicio 2- Ejecución paso a paso

Source History

```
6 public static void main(String[] args) {
7     CuentaXXX miCuentaXXX;
8     double saldoActual;
9     int x;
10
11     miCuentaXXX = new CuentaXXX("Antonio Sánchez","ES55-3058-2365-8522-1234-5678",2500,0);
12     System.out.println("Saldo inicial de la cuenta: "+miCuentaXXX.estado()+"€");
13     try
14     {
15         //x=1500;
16         x=50;
17         System.out.println("Retirada en cuenta de "+x+"€");
18         miCuentaXXX.retirar(x);
19     } catch (Exception e)
20     {
21         System.out.print("Fallo al retirar");
22     }
23
24     try
25     {
26         // x=100;
```

Test Results Output Variables x

Name	Type	Value
x	int	50
<Enter new watch>		
Static		
args	String[]	#106(length=0)
miCuentaXXX	CuentaXXX	#107
nombre	String	"Antonio Sánchez"
cuenta	String	"ES55-3058-2365-8522-1234-5678"
saldo	double	2450.0
tipolInteres	double	0.0
x	int	50

Source History

```
17     System.out.println("Retirada en cuenta de "+x+"€");
18     miCuentaXXX.retirar(x);
19 } catch (Exception e)
20 {
21     System.out.print("Fallo al retirar");
22 }
23
24 try
25 {
26     // x=100;
27     x=25;
28     System.out.println("Ingreso en cuenta de "+x+"€");
29     miCuentaXXX.ingresar(x);
30 } catch (Exception e)
31 {
32     System.out.print("Fallo al ingresar");
33 }
34 saldoActual = miCuentaXXX.estado();
35 System.out.println("El saldo actual es "+ saldoActual );
36
37 }
```

Test Results Output Variables x

Name	Type	Value
x	int	25
<Enter new watch>		
Static		
args	String[]	#107(length=0)
miCuentaXXX	CuentaXXX	#108
nombre	String	"Antonio Sánchez"
cuenta	String	"ES55-3058-2365-8522-1234-5678"
saldo	double	2475.0
tipolInteres	double	0.0
x	int	25

Ejercicio 3- Casos de prueba retirar con un valor límite superior válido.

```
public void testRetirarLimiteSuperiorValido() throws Exception {
    System.out.println("Test de prueba para retirar el límite superior de dinero");
    double cantidad = 2500.0;

    try {
        CuentaXXX instance = new CuentaXXX("XXX", "ES55-3058-2365-8522-1234-5678", 2500, 0);
        instance.retirar(cantidad);
        assertTrue (instance.estado()==0);
    } catch (Exception e) { /* no debería saltar ninguna excepción en este caso,
        por lo que si lo hace es porque algo no está bien y el test debería fallar */

        fail("No debería haber saltado una excepción ya que se trata de un valor válido");
    }
}
```

Ejercicio 4- Casos de prueba retirar con valores no válidos

Las clases de equivalencia serían:

- Por debajo: $x \leq 0$ (esta clase de equivalencia correspondería a valores no válidos)
- Dentro: $0 < x \leq 2500$ (esta clase de equivalencia correspondería a valores válidos)
- Por encima: $x > 2500$ (esta clase de equivalencia correspondería a valores no válidos)

Los valores límite serían 0.00, 0.01, 2500.00, 2500.01 (los valores límite no válidos serían 0.00 como límite inferior no válido y 2500.01 como límite superior no válido).

Por lo tanto los tests de pruebas para valores no válidos serían:

- Un valor negativo
- Un valor superior a 2500
- Valor límite inferior no válido: 0.00
- Valor límite superior no válido: 2500,01

```
public void testRetirarNoValidoNegativo() throws Exception {
    System.out.println("Test de prueba para retirar un valor negativo");
    double cantidad = -10;

    try {
        CuentaXXX instance = new CuentaXXX("XXX", "ES55-3058-2365-8522-1234-5678", 2500, 0);
        instance.retirar(cantidad);

        fail("Debería haber saltado una excepción ya que se trata de un valor no válido");
    } catch (Exception e) {
        System.out.println(e);
        assertTrue(instance.estado()==2500);
    }
}

public void testRetirarLimiteInferiorNoValido() throws Exception {
    System.out.println("Test de prueba para retirar 0€");
    double cantidad = 0;

    try {
        CuentaXXX instance = new CuentaXXX("XXX", "ES55-3058-2365-8522-1234-5678", 2500, 0);
        instance.retirar(cantidad);

        fail("Debería haber saltado una excepción ya que se trata de un valor no válido");
    } catch (Exception e) {
        System.out.println(e);
        assertTrue(instance.estado()==2500);
    }
}
```

```

public void testRetirarLimiteSuperiorNoValido() throws Exception {
    System.out.println("Test de prueba para retirar 2500,01€");
    double cantidad = 2500.01;

    try {
        CuentaXXX instance = new CuentaXXX("XXX", "ES55-3058-2365-8522-1234-5678", 2500, 0);
        instance.retirar(cantidad);

        fail("Debería haber saltado una excepción ya que se trata de un valor no válido");

    } catch (Exception e) {
        System.out.println(e);
        assertTrue(instance.estado()==2500);
    }
}

public void testRetirarNoValidoExceso() throws Exception {
    System.out.println("Test de prueba para retirar más dinero del que hay en la cuenta");
    double cantidad = 3000;

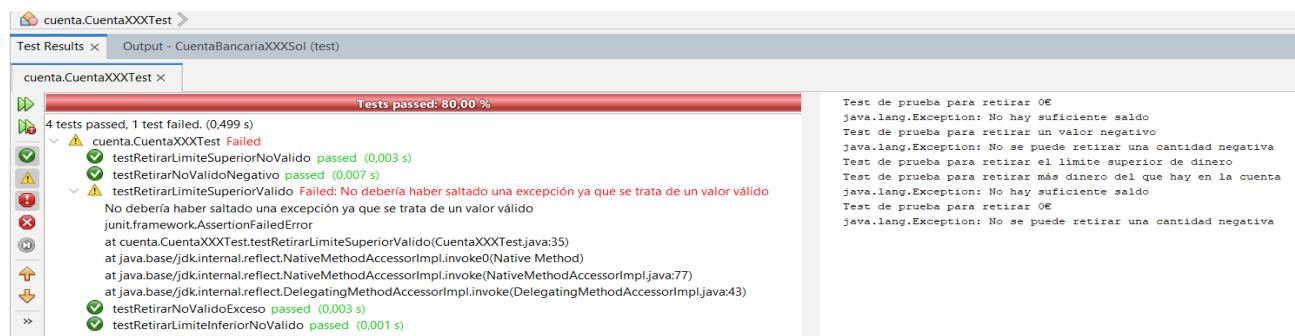
    try {
        CuentaXXX instance = new CuentaXXX("XXX", "ES55-3058-2365-8522-1234-5678", 2500, 0);
        instance.retirar(cantidad);

        fail("Debería haber saltado una excepción ya que se trata de un valor no válido");

    } catch (Exception e) {
        System.out.println(e);
        assertTrue(instance.estado()==2500);
    }
}

```

Ejercicio 5- Ejecución y argumentación del resultado.



La ejecución de los test de pruebas nos muestra que todos los test de pruebas para valores no válidos ha pasado, lo que significa que el método retirar funciona correctamente ante valores no válidos.

Sin embargo, el test para el valor límite superior válido ha fallado, lo que quiere decir que si el test está bien diseñado ha detectado que hay un error en la implementación del método retirar ya que no se comporta correctamente ante el valor límite superior válido.

Por lo tanto la ejecución de todos estos tests de pruebas y el fallo de uno de éstos nos ha detectado que el método retirar contiene errores en el código.

Si nos fijamos en el código podemos observar que el error está en la línea 64, la instrucción: `if (estado())<= cantidad)` debería ser `if (estado())< cantidad)` para permitir sacar el máximo dinero que hay.