

TAREA 03. ENTORNOS DE DESARROLLO.

En esta tarea se considera una clase **Java** CCuenta que dispone de los métodos main, ingresar y retirar. Este es el código de los métodos main, ingresar y retir que deberás tener en cuenta para resolver la tarea:

Método main

```
public static void main(String[] args) {  
    // Depuracion. Se detiene siempre  
    CCuenta miCuenta = new CCuenta();  
    System.out.println("Saldo Inicial: " + miCuenta.dSaldo + " euros");  
    // Depuracion. Provoca parada por ingreso con cantidad menor de 0  
    miCuenta.ingresar(-100);  
    System.out.println("Saldo Inicial: " + miCuenta.dSaldo + " euros");  
    miCuenta.ingresar(100);  
    System.out.println("Saldo tras ingreso: " + miCuenta.dSaldo + " euros");  
    miCuenta.ingresar(200);  
    System.out.println("Saldo tras ingreso: " + miCuenta.dSaldo + " euros");  
    // Depuracion. Provoca parada con codicion de tercer ingreso  
    miCuenta.ingresar(300);  
    System.out.println("Saldo tras ingreso: " + miCuenta.dSaldo + " euros");  
    miCuenta.retirar(50);  
    System.out.println("Saldo tras retirada: " + miCuenta.dSaldo + " euros");  
}
```

Método ingresar

```
public int ingresar(double cantidad)  
{  
    int iCodErr;  
    if (cantidad < 0)  
    {  
        System.out.println("No se puede ingresar una cantidad negativa");  
        iCodErr = 1;  
    }  
    else if (cantidad == -3)  
    {  
        System.out.println("Error detectable en pruebas de caja blanca");  
        iCodErr = 2;  
    }  
    else  
    {  
        // Depuracion. Punto de parada. Solo en el 3 ingreso  
        dSaldo = dSaldo + cantidad;  
        iCodErr = 0;  
    }  
    // Depuracion. Punto de parada cuando la cantidad es menor de 0  
    return iCodErr;  
}
```

Método retirar

```
public void retirar (double cantidad)
{
    if (cantidad <= 0)
    {
        System.out.println("No se puede retirar una cantidad negativa");
    }
    else if (dSaldo < cantidad)
    {
        System.out.println("No se hay suficiente saldo");
    }
    else
    {
        }
    }
}
```

1. Realiza un **análisis de caja blanca** completo del método ingresar.
- a. Creación del grafo del método ingresar

Nodo 1: if(cantidad < 0)

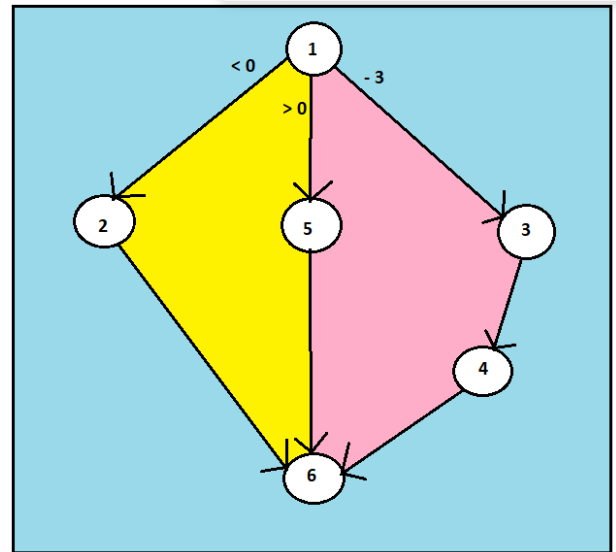
Nodo 2: iCodErr = 1;

Nodo 3: else if (cantidad == -3)

Nodo 4: iCodErr = 2;

Nodo 5: iCodErr = 0;

Nodo 6: return iCodErr;



- b. Complejidad de McCabe o ciclomática

7 aristas – 6 nodos + 2 = 3

2 nº de condiciones + 1 = 3 (nodo 1)

3 regiones (las 3 zonas sombreadas, amarillo, rosa y azul)

- c. Caminos de prueba.

Camino 1: 1 – 2 – 6

Camino 2: 1 – 5 – 6

Camino 3: 1 – 3 – 4 – 6

Podemos observar que el camino 3, nunca se va a realizar, queda consumido dentro del camino1, ya que -3 es < 0.

- d. Casos de uso, resultados esperados y análisis.

Camino	Cantidad	iCodError
1	-5	1
2	5	0
3	-3	2

Tenemos 3 casos, que la cantidad sea un número negativo, un número positivo y el valor concreto – 3, que va a quedar consumido por el caso 1.

2. Realiza un **análisis de caja negra**, incluyendo valores límite y conjetura de errores del método retirar. Debes considerar que este método recibe como parámetro la cantidad a retirar, que no podrá ser menor a 0. Además en ningún caso esta cantidad podrá ser mayor al saldo actual. Al tratarse de pruebas funcionales no es necesario conocer los detalles del código pero te lo pasamos para que lo tengas.

- a. Determinar las clases de equivalencia y análisis de valores límite

Retirar				
Dominio	Clase	Tipo	Límite inferior	Límite superior
Número enteros	$(-\infty, 0)$	No Válida	$-\infty$	-1
	$[0, dSaldo]$	Válida	0	dSaldo
	$(dSaldo, \infty)$	No Válida	dSaldo + 1	∞

- b. Casos de uso, resultados esperados y análisis

Imagino que tenemos un dSaldo = 200, para ejemplificar con los valores límites.

Retirar		
Caso	Cantidad	Resultado esperado
1	-500	No se puede retirar una cantidad negativa
2	-1	
3	0	Válida
4	200	
5	201	No se hay suficiente saldo
6	500	

3. Crea la clase CCuentaTest del tipo **Caso de prueba JUnit** en **Eclipse** que nos permita pasar las pruebas unitarias de caja blanca del método ingresar. Los casos de prueba ya los habrás obtenido en el primer apartado del ejercicio. Copia el código fuente de esta clase en el documento.

Realizo 3 Tests:

```
class CCuentaTest {  
  
    CCuenta miCuenta = new CCuenta();  
  
    int ingreso1 = -5;  
    int ingreso2 = 5;  
    int ingreso3 = -3;  
  
    @Test  
    void testIngresar1() {  
        assertEquals(1, miCuenta.ingresar(ingreso1));  
    }  
  
    @Test  
    void testIngresar2() {  
        assertEquals(0, miCuenta.ingresar(ingreso2));  
    }  
  
    @Test  
    void testIngresar3() {  
        assertEquals(2, miCuenta.ingresar(ingreso3));  
    }  
}
```

4. Genera los siguientes **puntos de ruptura** para validar el comportamiento del método ingresar en modo depuración.
 - Punto de parada sin condición al crear el objeto miCuenta en la función main. Línea 3 del código del método main que se presenta en la siguiente página de este libro.
 - Punto de parada en la instrucción return del método ingresar sólo si la cantidad a ingresar es menor de 0. Línea 20 del código del método ingresar que se presenta más adelante.
 - Punto de parada en la instrucción donde se actualiza el saldo, sólo deberá parar la tercera vez que sea actualizado. Línea 16 del código del método ingresar que se presenta más adelante.

Pulsando el botón derecho sobre la ventana de puntos de ruptura selecciona la opción "Exportar puntos de ruptura". Seleccionas los tres puntos de ruptura generados y guardas el fichero. El fichero tendrá la extensión bkpt, la cambias por txt. Ahora abres el fichero y copias el contenido íntegramente al documento.

```
<?xml version="1.0" encoding="UTF-8"?>
<breakpoints>
<breakpoint enabled="true" persistent="true" registered="true">
<resource path="/ED03_Tarea/src/CCuenta/CCuenta.java" type="1"/>
<marker charStart="127" lineNumber="6"
type="org.eclipse.jdt.debug.javaLineBreakpointMarker">
<attrib name="charStart" value="127"/>
<attrib name="org.eclipse.jdt.debug.core.suspendPolicy" value="2"/>
<attrib name="org.eclipse.jdt.debug.ui.JAVA_ELEMENT_HANDLE_ID"
value="ED03_Tarea/src/CCuenta{CCuenta.java[CCuenta]"/>
<attrib name="charEnd" value="165"/>
<attrib name="org.eclipse.debug.core.enabled" value="true"/>
<attrib name="message" value="Line breakpoint:CCuenta [line: 6] - main(String[])/>
<attrib name="org.eclipse.debug.core.id" value="org.eclipse.jdt.debug"/>
<attrib name="org.eclipse.jdt.debug.core.typeName" value="CCuenta.CCuenta"/>
<attrib name="workingset_name" value=""/>
<attrib name="workingset_id" value="org.eclipse.debug.ui.breakpointWorkingSet"/>
</marker>
```

```
</breakpoint>

<breakpoint enabled="true" persistent="true" registered="true">

<resource path="/ED03_Tarea/src/CCuenta/CCuenta.java" type="1"/>

<marker charStart="1614" lineNumber="46"
type="org.eclipse.jdt.debug.javaLineBreakpointMarker">

<attrib name="charStart" value="1614"/>

<attrib name="org.eclipse.jdt.debug.core.suspendPolicy" value="2"/>

<attrib name="org.eclipse.jdt.debug.ui.JAVA_ELEMENT_HANDLE_ID"
value="ED03_Tarea/src/CCuenta/CCuenta.java[CCuenta]"/>

<attrib name="org.eclipse.jdt.debug.core.hitCount" value="3"/>

<attrib name="charEnd" value="1650"/>

<attrib name="org.eclipse.debug.core.enabled" value="true"/>

<attrib name="org.eclipse.jdt.debug.core.expired" value="false"/>

<attrib name="message" value="Line breakpoint:CCuenta [line: 46] [hit count: 3] -
ingresar(double)"/>

<attrib name="org.eclipse.debug.core.id" value="org.eclipse.jdt.debug"/>

<attrib name="org.eclipse.jdt.debug.core.typeName" value="CCuenta.CCuenta"/>

<attrib name="workingset_name" value=""/>

<attrib name="workingset_id" value="org.eclipse.debug.ui.breakpointWorkingSet"/>

</marker>

</breakpoint>

<breakpoint enabled="true" persistent="true" registered="true">

<resource path="/ED03_Tarea/src/CCuenta/CCuenta.java" type="1"/>

<marker charStart="1775" lineNumber="51"
type="org.eclipse.jdt.debug.javaLineBreakpointMarker">

<attrib name="org.eclipse.jdt.debug.core.conditionEnabled" value="true"/>

<attrib name="charStart" value="1775"/>

<attrib name="org.eclipse.jdt.debug.core.suspendPolicy" value="2"/>

<attrib name="org.eclipse.jdt.debug.core.condition" value="cantidad < 0"/>

<attrib name="org.eclipse.jdt.debug.ui.JAVA_ELEMENT_HANDLE_ID"
value="ED03_Tarea/src/CCuenta/CCuenta.java[CCuenta]"/>

<attrib name="charEnd" value="1798"/>

<attrib name="org.eclipse.debug.core.enabled" value="true"/>
```

```
<attrib name="message" value="Line breakpoint:CCuenta [line: 51] [conditional] -
ingresar(double)"/>
<attrib name="org.eclipse.debug.core.id" value="org.eclipse.jdt.debug"/>
<attrib name="org.eclipse.jdt.debug.core.typeName" value="CCuenta.CCuenta"/>
<attrib name="workingset_name" value=""/>
<attrib name="workingset_id" value="org.eclipse.debug.ui.breakpointWorkingSet"/>
</marker>
</breakpoint>
</breakpoints>
```