

Tarea para ED03.

Detalles de la tarea de esta unidad.

Enunciado.

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

```
1 public static void main(String[] args) {
2     // Depuracion. Se detiene siempre
3     CCuenta miCuenta = new CCuenta();
4     System.out.println("Saldo Inicial: " + miCuenta.dSaldo + " euros");
5     // Depuracion. Provoca parada por ingreso con cantidad menor de 0
6     miCuenta.ingresar(-100);
7     System.out.println("Saldo Inicial: " + miCuenta.dSaldo + " euros");
8     miCuenta.ingresar(100);
9     System.out.println("Saldo tras ingreso: " + miCuenta.dSaldo + " euros");
10    miCuenta.ingresar(200);
11    System.out.println("Saldo tras ingreso: " + miCuenta.dSaldo + " euros");
12    // Depuracion. Provoca parada con codicion de tercer ingreso
13    miCuenta.ingresar(300);
14    System.out.println("Saldo tras ingreso: " + miCuenta.dSaldo + " euros");
15    miCuenta.retirar(50);
16    System.out.println("Saldo tras retirada: " + miCuenta.dSaldo + " euros");
17 }
```

Método ingresar

```
1 public int ingresar(double cantidad)
2     int iCodErr;
3     if (cantidad < 0)
4     {
5         System.out.println("No se puede ingresar una cantidad negativa");
6         iCodErr = 1;
7     }
8     else if (cantidad == -3)
9     {
10        System.out.println("Error detectable en pruebas de caja blanca");
11        iCodErr = 2;
12    }
13    else
14    {
15        // Depuracion. Punto de parada. Solo en el 3 ingreso
```

```

16         dSaldo = dSaldo + cantidad;
17         iCodErr = 0;
18     }
19     // Depuracion. Punto de parada cuando la cantidad es menor de 0
20     return iCodErr;
21 }

```

Método retirar

```

1  public void retirar (double cantidad)
2  {
3      if (cantidad <= 0)
4      {
5          System.out.println("No se puede retirar una cantidad negativa");
6      }
7      else if (dSaldo < cantidad)
8      {
9          System.out.println("No se hay suficiente saldo");
10     }
11     else
12     {
13     }
14 }
15 }

```

Deberás realizar un documento donde dar respuesta a los siguientes apartados:

1. Realiza un **análisis de caja blanca** completo del método ingresar.
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.
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.
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.

Criterios de puntuación. Total 10 puntos.

Los criterios de puntuación serán los siguientes.

- Pruebas de caja blanca. 2,5 puntos.
- Pruebas de caja negra. 2,5 puntos.
- 2,5 puntos.
- Depuración. 2,5 puntos.

Recursos necesarios para realizar la Tarea.

Ordenador con un **IDE que permita programar en Java. Si es Eclipse mejor, porque el proyecto inicial y la solución está en Eclipse.**

Pulsa [aquí](#) para descarga el proyecto **Java "java-sw.rar"**.

Consejos y recomendaciones.

Se pretenden que partiendo de una clase con dos métodos ya diseñados, se utilicen herramientas de depuración para corregir errores en el código. A continuación, se deberá diseñar todos los casos de prueba, para verificar el comportamiento de los métodos ingresar y retirar.

Indicaciones de entrega.

Una vez realizada la tarea elaborarás un único documento donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna **Begoña Sánchez Mañas para la tercera unidad del MP de ED**, debería nombrar esta tarea como...

sanchez_manas_begona_ED03_Tarea