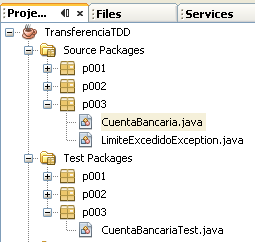
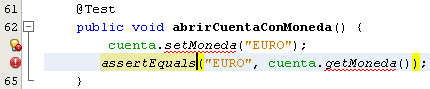
**Aplicación transferencia bancaria usando TDD**

**Parte 3**

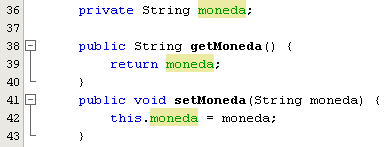
Creamos el paquete p003 y obramos de forma análoga a cuando pasamos de p001 a p002:



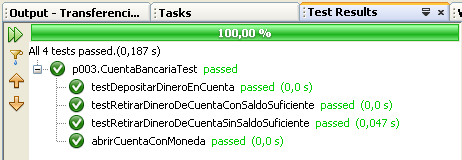
Ahora es el turno de añadir la funcionalidad de las divisas a la aplicación. Comenzamos por crear un test en el que tendremos un método setter para cambiar la propiedad ‘moneda’ del SUT:



Ahora creamos en el SUT tanto el atributo ‘moneda’ como su pareja getter/setter:



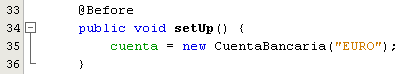
Comprobamos:



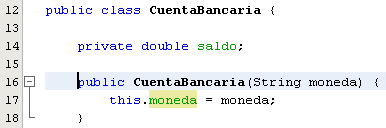
Ok.

Pero un momento, ¿no sería lo más adecuado indicar la moneda de una cuenta en el momento de su creación? Decidimos que sí y que el constructor sin parámetros ya no tiene sentido.

En la clase de test, cambiamos la manera en que se crean las cuentas:

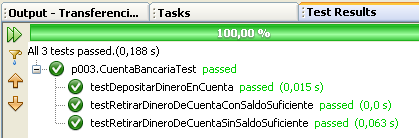


Eliminamos el constructor sin parámetros y creamos uno que acepte un String que permita indicar la divisa de la cuenta:

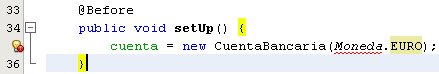


Por otro lado, parece que ya no tiene sentido mantener un método de test llamado abrirCuentaConMoneda(). Procedemos a eliminarlo.

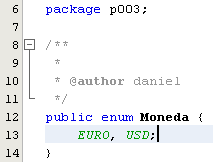
Comprobamos:



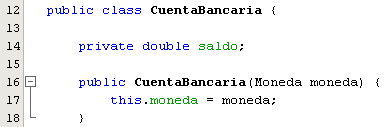
Ahora podemos refactorizar: sustituimos el String utilizado para las divisas por un enumerado:



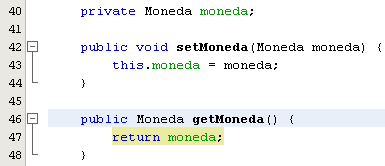
Para ello, creamos el enumerado junto a la clase CuentaBancaria:



Y en CuentaBancaria tenemos que cambiar el constructor:

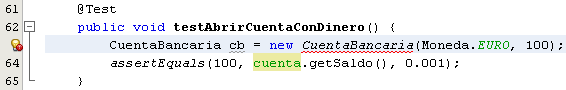


También el tipo del atributo moneda y su pareja de get/set:



Comprobamos que seguimos estando en verde.

Lo siguiente que podemos pensar es en crear una cuenta informando a la vez del importe con la que la inicializamos. Para ello creamos un test:



A continuación, el constructor:

