

Ejemplo de Banco con excepciones

```
public abstract class ACuenta {
    /*
     * Ejemplo de banco con excepciones
     * Juan Manuel fernández. Curso 2011
     */
    private double saldo;
    private String cuenta;

    public ACuenta(String cu, double salini){
        cuenta = cu;
        saldo = salini;
    }

    public double getSaldo() {
        return saldo;
    }

    protected void setSaldo(double kan){
        saldo = kan;
    }

    public String getCuenta() {
        return cuenta;
    }

    public void deposita(double kan) throws ExcepciónBanco{
        if (kan<=0) throw new ExcepciónBanco("Depósito ilegal", kan);
        saldo += kan;
    }

    public abstract double retira(double kan) throws ExcepciónBanco;
}
```

```

public class CuentaDébito extends ACuenta {
    /*
     * Ejemplo de banco con excepciones
     * Juan Manuel fernández. Curso 2011
     */

    public CuentaDébito(String ku, double sain) {
        super(ku, sain);
        // TODO Auto-generated constructor stub
    }

    @Override
    public double retira(double kan) throws ExcepciónBanco{
        double pago = 0;
        if (kan<=0) throw new ExcepciónBanco("pedido ilegal", kan);
        else
            if (kan> getSaldo()) throw new ExcepciónBanco("Pide más del saldo", kan);
            else{
                super.setSaldo(getSaldo()-kan);
                pago = kan;
            }
        return pago;
    }
}

```

```

public class ExcepciónBanco extends Exception {
    /*
     * Ejemplo de banco con excepciones
     * Juan Manuel fernández. Curso 2011
     */

    private double auxiliar;

    public ExcepciónBanco(String mens, double aux){
        super(mens);
        auxiliar = aux;
    }

    public double getAuxiliar(){
        return auxiliar;
    }
}

```

```
import junit.framework.TestCase;

public class CuentaDébitoTest extends TestCase {
    private CuentaDébito cuen;

    protected void setUp() throws Exception {
        super.setUp();
        cuen = new CuentaDébito("10010009", 2000.00);
    }

    public void testRetira() {
        double rr=0;
        System.out.println("Saldo inicial: "+cuen.getSaldo());
        try{
            rr = cuen.retira(500);
        }catch(ExcepciónBanco eb){
            System.out.println("Hubo un problema: "+eb);
            System.out.println("Con valor auxiliar: "+eb.getAuxiliar());
        }
        System.out.println("Saldo final: "+cuen.getSaldo());
        assertEquals(500.00, rr, 0.02);
        assertEquals(1500.00, cuen.getSaldo(), 0.02);
    }

    public void testDeposita() {
        System.out.println("Saldo inicial: "+cuen.getSaldo());
        try{
            cuen.deposita(600);
        }catch(ExcepciónBanco eb){
            System.out.println("Hubo un problema: "+eb);
            System.out.println("Con valor auxiliar: "+eb.getAuxiliar());
        }
        System.out.println("Saldo final: "+cuen.getSaldo());
        assertEquals(2600.00, cuen.getSaldo(), 0.02);
    }

    public void testRetira2() {
        double rr=0;
        System.out.println("Saldo inicial: "+cuen.getSaldo());
```

```
    try{
        rr = cuen.retira(-350);
    }catch(ExcepciónBanco eb){
        System.out.println("Hubo un problema: "+eb);
        System.out.println("Con valor auxiliar: "+eb.getAuxiliar());
    }
    System.out.println("Saldo final: "+cuen.getSaldo());
    assertEquals(350.00, rr, 0.02);
    assertEquals(2350.00, cuen.getSaldo(), 0.02);
}
}
```