

Mis cursos &gt; ... &gt; Crear nuestras propias Excepciones

## Ejercicio

Aprobado

### Instrucciones

Tenemos una clase cuenta con titular y saldo, los comportamientos posibles son depositar y extraer. Extender la clase Exception, para generar excepciones, cuando se intente crear una cuenta con saldo negativo o se intente extraer un importe mayor al saldo. Mostrar el nombre de la Excepción cuando se genera.

La clase Cuenta

```
public class Cuenta
{
    private String titular;
    private double saldo;
    public Cuenta(String titular, double saldo) throws CuentaException
    {
        this.titular=titular;
        if (saldo<0)
            throw new CuentaException("No se puede crear una cuenta con saldo negativo");
        this.saldo=saldo;
    }
    public void extraer(double importe) throws CuentaException
    {
        if (importe >saldo)
            throw new CuentaException("No hay saldo suficiente para esta extracción");
        saldo=saldo-importe;
    }
    public void depositar(double importe)
    {
        saldo+=importe;
    }
}
```

### Pistas

#### 💡 Pista 1



Las excepciones se usaran en el método extraer y en el constructor



Ejecutar respuesta

Ver modo vertical

### PlaygroundCode

CuentaExeption.java X

### Terminal

SRG

CuentaExeption.java

PATH: SRC &gt; CUENTAEXEPTION.JAVA

```
1  class CuentaException extends Exception{
2
3      public CuentaException() {
4          super();
5      }
6
7      public CuentaException(String mensajeExcepcion) {
8          super(mensajeExcepcion);
9      }
10
11     @Override
12     public String toString() {
13         return "Se produjo la excepción " + this.getClass().getName() +
14             " cuyo mensaje es: " + this.getMessage();
15     }
16
17 }
18
```

```
Executing tasks ':classes :testClasses'
Starting Gradle Daemon...
Gradle Daemon started
> Task :compileJava UP-TO-DATE
> Task :processResources UP-TO-DATE
> Task :classes UP-TO-DATE
> Task :compileTestJava
> Task :testClasses
> Task :processTestResources NO-SOURCE
```