



## Taller Creación de Cuentas

## Objetivos

Aplicar los siguientes conceptos:

- Clases y Objetos
- Principio de Encapsulamiento
- Constructores

## Consideraciones

Todo el código debe cumplir con los estándares de programación Java revistados hasta el momento:

Nombrado de paquetes

Nombrado de clases

Nombrado de métodos

Nombrado de atributos

## Desarrollo del Taller

### 1. Estructura del Proyecto

- Nombre del Proyecto: Taller1
- Paquetes:
  - clearminds.cuentas
  - clearminds.cuentas.test

### 2. Instrucciones

a. En el paquete clearminds.cuentas, crear una clase Cuenta

b. Atributos de Cuenta:

Id (cadena de caracteres)

Tipo (cadena de caracteres)

Saldo (decimal)

Para cada atributo se debe aplicar el principio de encapsulamiento.

Los atributos tipo y saldo deben tener métodos get y set.

El atributo id solo tendrá el método get, de manera que no podrá ser modificado desde otra clase.

El valor por defecto del atributo tipo será A.

c. Constructores de Cuenta:

Crear un constructor que reciba el id de cuenta y setee el atributo id del objeto con este valor, además debe setear el atributo tipo en el valor por defecto (A), no es necesario setear el valor de saldo porque el atributo tiene el valor 0.0 por defecto.

En este ejemplo no aplica el constructor vacío, ya que al no poseer un método setId, una vez creado el objeto con el constructor vacío, no tendría forma de darle valor al Id de cuenta y una cuenta sin Id no tiene sentido. (NO crear el constructor vacío)

Crear un constructor que reciba los tres atributos de cuenta y setee sus valores.

## d. Otros métodos de Cuenta:

Crear un método imprimir que despliegue los datos de la cuenta en consola con el siguiente formato:

```
*****
```

```
CUENTA
```

```
*****
```

```
Número de Cuenta: <valor de id>
```

```
Tipo: <tipo de cuenta>
```

```
Saldo: USD <saldo de la cuenta>
```

```
*****
```

Ejemplo:

```
*****
```

```
CUENTA
```

```
*****
```

```
Número de Cuenta: 00342
```

```
Tipo: A
```

```
Saldo: USD 456
```

```
*****
```

Un método imprimirConMiEstilo que no reciba parámetros y que imprima en consola los datos de la cuenta en un formato a su elección.

e. Clase TestCuenta:

En el paquete clearminds.cuentas.test, crear una clase TestCuenta que posea un método main con el siguiente código:

```
//Creo el objeto Cuenta y lo referencio con cuenta1
```

```
Cuenta cuenta1=new Cuenta( "03476");
```

```
//Coloco un valor de saldo en la cuenta1
```

```
cuenta1.setSaldo(675);
```

```
//Creo el objeto Cuenta y lo referencio con cuenta2
```

```
Cuenta cuenta2=new Cuenta( "03476","C",98);
```

```
//Creo el objeto Cuenta y lo referencio con cuenta3
```

```
Cuenta cuenta3=new Cuenta( "03476");
```

```
//Modifico el tipo de cuenta
```

```
cuenta3.setTipo('C');
```

```
//Imprimo cuenta1, cuenta2 y cuenta3
```

```
System.out.println("-----Valores Iniciales-----");
```

```
cuenta1.imprimir();
```

```
cuenta2.imprimir();
```

```
cuenta3.imprimir();
```

```
System.out.println("-----Valores modificados-----");
```

- Adicionalmente, en el método main:
  - Modificar el saldo de la cuenta1 a 444
  - Modificar el saldo de la cuenta3 a 567
  - Modificar el tipo de la cuenta2 a D
  - Imprimir los nuevos valores de cuenta1, cuenta2 y cuenta3
- 
- Crear una cuenta4 utilizando el constructor que recibe el Id de cuenta y modificar los atributos de ser necesario.

Al final la cuenta4 debe ter los siguientes valores:

Id: 0987

Tipo: 'A'

Saldo:10

- Crear una cuenta5 utilizando el constructor que recibe los tres parámetros y modificar los atributos de ser necesario.

Al final la cuenta5 debe ter los siguientes valores:

Id: 0557

Tipo: 'C'

Saldo:10

- Crear una cuenta6 utilizando cualquier constructor y modificar los atributos de ser necesario.

Al final la cuenta6 debe tener los siguientes valores:

Id: 0666

Tipo: 'A'

Saldo:0

Para la creación y seteo de atributos de la cuenta 6 utilizar la menor cantidad de código posible.

- Imprimir los valores de cuenta4, cuenta5 y cuenta6 utilizando el método imprimirConMiEstilo.