



## Ejercicio 11

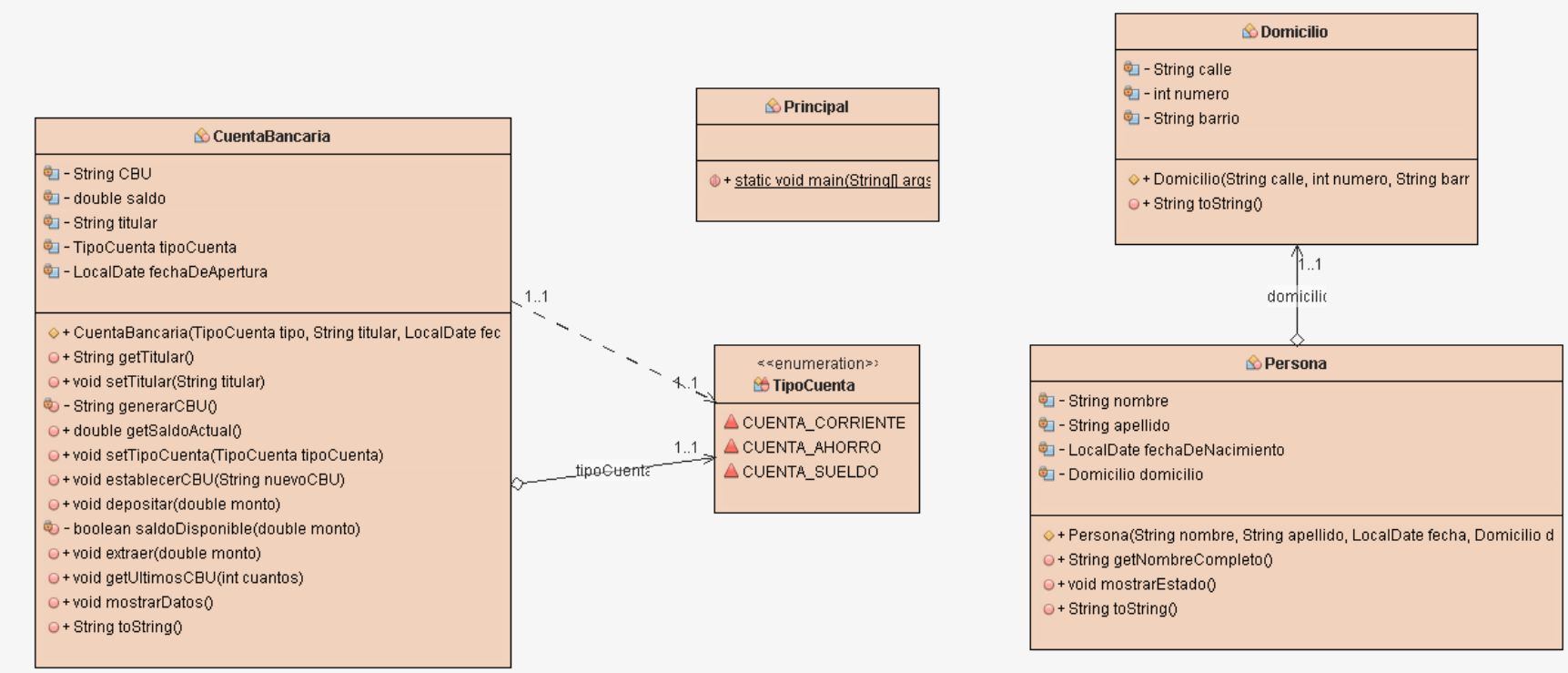
Utilizando las clases generadas hasta ahora, codificá en Java la siguiente situación:

Una cuenta bancaria de tipo caja de ahorros le pertenece a Fulano Gomez, nacido el 16/04/1990 y otra de tipo cuenta corriente le pertenece a Mengana Torres, nacida el 23/11/1991. Ambos están casados y viven juntos en Av. Triunvirato 3174, Villa Ortúzar.

Supongamos que Fulano y Mengana se mudan a nuevo hogar: ¿Hay que cambiar el domicilio de cada uno o basta con cambiar uno de los dos?

**Rta:** Como en el programa que armé creé 2 variables instanciadas a la clase Domicilio, tendré que modificar ambos objetos. Solo en el caso de que hubiera igualado ambos valores, en ese caso si podría haber solo modificado un valor.

### Diagrama UML:



```
package ejercicio_11;

import java.time.LocalDate;
public class Principal {

    public static void main(String[] args) {
        // Crear Personas
        Domicilio domP1 = new Domicilio("Av. Triunvirato", 3174, "Villa Ortúzar");
        Domicilio domP2 = new Domicilio("Av. Triunvirato", 3174, "Villa Ortúzar");
        Persona P1 = new Persona("Fulano", "Gomez", LocalDate.of(1990, 04, 16), domP1);
        Persona P2 = new Persona("Mengana", "Torres", LocalDate.of(1991, 11, 23), domP2);

        // Crear Cuentas Bancarias
        CuentaBancaria cbP1 = new CuentaBancaria(TipoCuenta.CUENTA_AHORRO, P1.getNombreCompleto(), LocalDate.now());
        CuentaBancaria cbP2 = new CuentaBancaria(TipoCuenta.CUENTA_CORRIENTE, P2.getNombreCompleto(), LocalDate.now());

        //mostrar Datos
        P1.mostrarEstado();
        cbP1.mostrarDatos();
        P2.mostrarEstado();
        cbP2.mostrarDatos();
    }
}
```

Modifiqué algunas clases:

- A la `clase Persona`
  - Le modifiqué el año de nacimiento por la fecha de nacimiento (cosa que no había hecho antes en la refactorización) —> (ya que en ese momento no entendía como hacerlo)
  - Eliminé algunos métodos que no necesité para este proyecto.
- A la clase `CuentaBancaria`
  - Creé la clase enum TipoCuenta para mandar el tipo de cuenta con los valores constantes correspondientes en vez de mandarlos como String (cosa que no había hecho antes en la refactorización) —> (ya que en ese momento no entendía como hacerlo)

```
package ejercicio_11;

import java.util.Random;
```

```

import java.time.LocalDate;

public class CuentaBancaria {
    private String CBU;
    private double saldo;
    private String titular;
    private TipoCuenta tipoCuenta;
    private final LocalDate fechaDeApertura;

    //CONSTRUCTOR
    public CuentaBancaria(TipoCuenta tipo, String titular, LocalDate fecha) {
        this.CBU=generarCBU();
        this.saldo=0;
        this.tipoCuenta=tipo;
        this.titular = titular;
        this.fechaDeApertura = fecha;
    }

    public String getTitular() {
        return titular;
    }

    public void setTitular(String titular) {
        this.titular = titular;
    }

    private String generarCBU() {
        Random r = new Random();
        String elCbu = "";
        for (int i = 0; i < 22; i++) {
            elCbu += r.nextInt(10);
        }
        return elCbu;
    }

    public double getSaldoActual() {
        return saldo;
    }

    public void setTipoCuenta(TipoCuenta tipoCuenta) {
        this.tipoCuenta = tipoCuenta;
    }

    public void establecerCBU(String nuevoCBU){
        if(nuevoCBU!=null){
            CBU=nuevoCBU;
        }
    }

    public void depositar(double monto){
        if(monto > 0){
            saldo = saldo + monto; // saldo += monto
        }
    }

    private boolean saldoDisponible(double monto){
        return saldo >= monto;
    }

    public void extraer(double monto){
        if(saldoDisponible(monto)||tipoCuenta.equals("Cuenta corriente")){
            saldo = saldo - monto; //saldo -= monto
        }
    }

    public void getUltimosCBU(int cuantos){
        String ultimos = CBU.substring(CBU.length() - cuantos);
        System.out.println("Últimos " + cuantos + " dígitos del CBU: " + ultimos);
    }

    public void mostrarDatos(){
        System.out.println(toString());
    }

    @Override
    public String toString() {
        return "CuentaBancaria{" + "CBU=" + CBU + ", saldo=" + saldo + ", tipoCuenta=" + tipoCuenta + ", titular=" + titular + ", fechaDeApertura=" + fechaDeApertura + '}';
    }

}

```

```

package ejercicio_11;

import java.time.LocalDate;

public class Persona {
    private String nombre;
    private String apellido;
    private LocalDate fechaDeNacimiento;
    private Domicilio domicilio;

    // CONSTRUCTOR
    public Persona(String nombre, String apellido, LocalDate fecha, Domicilio dom) {
        this.nombre=nombre;
        this.apellido=apellido;
        this.fechaDeNacimiento=fecha;
        this.domicilio=dom;
    }

    public String getNombreCompleto (){
        return nombre + " " + apellido;
    }

    public void mostrarEstado () {
        System.out.println(toString());
    }

    @Override
    public String toString() {
        return "Persona{" + "nombre=" + nombre + ", apellido=" + apellido + ", fechaDeNacimiento=" + fechaDeNacimiento + ", domicilio=" + domicilio + '}';
    }

}

```

```
package ejercicio_11;

public enum TipoCuenta {
    CUENTA_CORRIENTE,
    CUENTA_AHORRO,
    CUENTA_SUELDO
}
```

```
package ejercicio_11;

public class Domicilio {
    private String calle;
    private int numero;
    private String barrio;

    public Domicilio(String calle, int numero, String barrio) {
        this.calle = calle;
        this.numero = numero;
        this.barrio = barrio;
    }

    @Override
    public String toString() {
        return "Domicilio{" + "calle=" + calle + ", numero=" + numero + ", barrio=" + barrio + '}';
    }
}
```