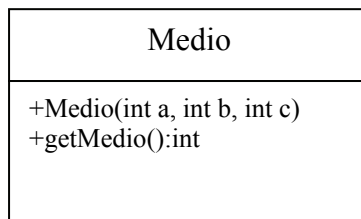


Ejercicios del Tema 14

Ejercicio 14.1:

Escriba una clase para calcular el valor medio de 3 números que se pasan como parámetros. A continuación someta el programa al proceso de pruebas descrito en la transparencia 14 hasta que:

- a) Se hayan recorrido todas las sentencias.
- b) Se hayan recorrido todas las condiciones a true y a false.



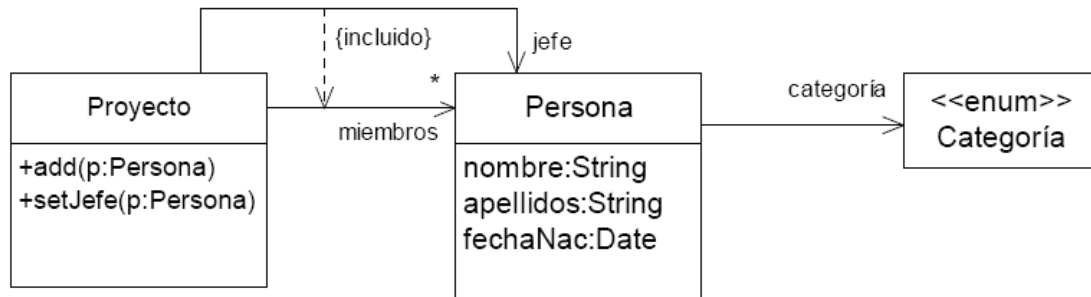
```
public class Medio{
    int a, b, c;

    public Medio (int a, int b, int c) {
        this.a=a;
        this.b=b;
        this.c=c;
    }
    public int getMedio () {
        int aux=0;
        for (int i=0;i<3;i++) {
            if (a>b) {
                aux=a;
                a=b;
                b=aux;
            }
            if (b>c) {
                aux=b;
                b=c;
                c=aux;
            }
            System.out.println(a+" "+b+" "+c);
        }
        return b;
    }
}
```

Casos de prueba: (1,2,3), (3,2,1), (2,1,3), (2,3,1), (1,3,2), (3,2,1)

Ejercicio 14.2:

Suponga que disponemos de un sistema como el mostrado en la figura. Proponga valores interesantes de tipo Persona para las operaciones add y setJefe del tipo Proyecto.



Posibles valores que pueden tomar los atributos de la clase Persona:

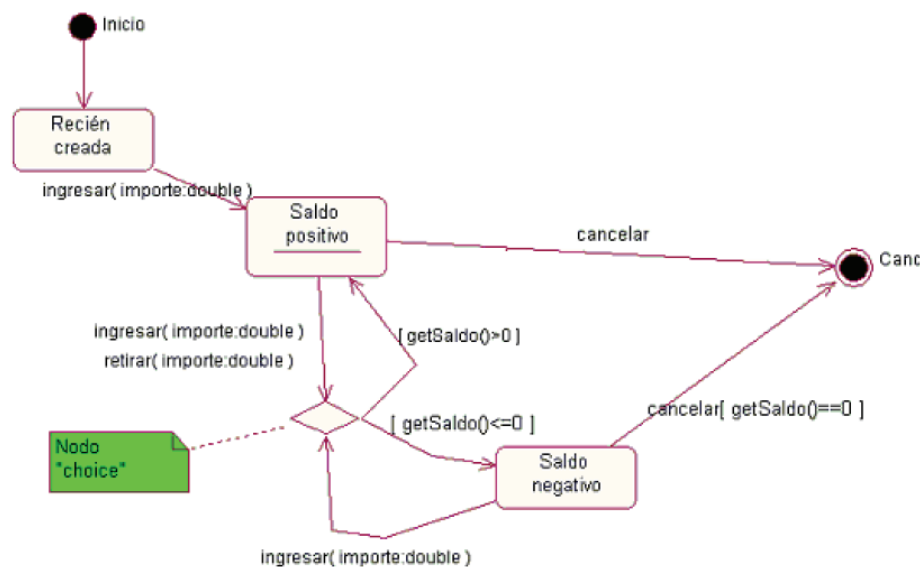
fechaNac = {válida (entre 16 y 65), no válida (futuro, menor 16, mayor 65)}
 nombre = {con datos (menor o igual 256 letras o dígitos, mayor a 256), sin datos}
 apellidos = {con datos (menor o igual 256 letras o dígitos, mayor a 256), sin datos}

De esta manera, esos atributos estarían tomando todos los valores posibles que pudieran causar errores.

Para probar la clase Proyecto, se pueden usar diversas técnicas, como añadir a una persona varias veces en el mismo proyecto (el programa debería quejarse, por tanto), poner a una persona como jefe de sí misma, etc.; y así crear un conjunto de casos de prueba.

Ejercicio 14.3:

Escriba casos de prueba que cumplan los criterios de cobertura de transición de estados para la siguiente máquina de estados, que representa el comportamiento de una cuenta corriente.



Lo que se persigue es realizar todas las transiciones posibles hacia todos los estados de la clase cuenta.

- Del estado cuenta_creada → ingresar(saldo positivo)
- Del estado saldo positivo → retirar(importe_mayor_saldo)
- Del estado saldo positivo → cancelar()
- Del estado saldo negativo → ingresar(importe_positivo)
- Del estado saldo negativo → cancelar()