

## Exercicis de Herència

### Exercici 1

Suposem que necessitem implementar dues classes que anomenarem Suma i Resta.

Cada classe té com a atribut valor1, valor2 i resultat, variables privades.

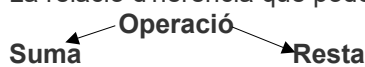
Els mètodes a definir són:

- cargar1 (que inicialitza l'atribut valor1),
- carga2 (que inicialitza l'atribut valor2),
- operar (que en el cas de la classe "Suma" suma els dos atributs i en el cas de la classe "Resta" fa la diferència entre valor1 i valor2)
- mostrarResultat.

Si analitzem totes dues classes trobem que molts atributs i mètodes són idèntics.

En aquests casos és bo definir una **classe pare** que agrupe aquests atributs i responsabilitats comunes.

La relació d'herència que podem disposar per a aquest problema és:



Solament el mètode operar és diferent per a les classes Suma i Resta (això fa que no ho puguem disposar en la classe Operacio).

Però els mètodes cargar1, cargar2 i mostrarResultat són idèntics a les dues classes, això fa que puguem disposar-los en la classe Operacio.

El mateix ocorre amb els atributs valor1, valor2 i resultat, es definiran en la classe pare Operacio.

Crea un projecte i després crea tres classes anomenades: Operacio, Suma, Resta i ajuda't de la següent classe Prova.

```
public class Prueba {
    public static void main(String[] ar) {
        Suma suma1=new Suma();
        suma1.cargar1();
        suma1.cargar2();
        suma1.operar();
        System.out.print("El resultado de la suma es:");
        suma1.mostrarResultado();
        Resta resta1=new Resta();
        resta1.cargar1();
        resta1.cargar2();
        resta1.operar();
        System.out.print("El resultado de la resta es:");
        resta1.mostrarResultado();
    }
}
```

## Exercici 2.

Construeix una aplicació sobre els diferents tipus de **vehicles** existents, **vehicles aquàtics** i **aeris**.

El factor comú és que sense importar si és un **avió** o un **vaixell**, estem parlant de vehicles, i tots dos poden compartir característiques comunes per aquesta raó la Classe principal del nostre sistema serà la classe Vehicule.

Podem dir que un **vaixell** a més de descendir d'un **vehicle**, també descendeix d'una mena de **vehicles Aquàtics** els quals tenen un conjunt de característiques comunes per a qualsevol vehicle aquàtic (sense oblidar que cada vehicle aquàtic pot tindre les seues característiques pròpies) i de la mateixa forma un **avió** descendeix d'una mena de **vehicles Aeris** (amb la mateixa lògica de l'anterior).

Estructura de classes:



Com veiem, la Classe Pare és Vehicle, la qual té com a filles a les classes Aquàtic i Aeri, aquestes al seu torn són classes Pare de Barco, Veler, Avió i Helicòpter que alhora són netes de Vehicle.

La classe Avió pot usar el mètode volar() de la classe Aeri, així com també usar el mètode transportar() de la classe Vehicle, perquè es troba en el mateix arbre d'herència, no obstant això no pot usar el mètode navegar() de la classe Aquàtic, ja que no és de tipus Aquàtic.

Crea cadascuna de les classes indicades en l'esquema, juntament amb els mètodes indicats també en l'esquema:

Haurà de poder executar-se la següent classe principal:

```
public class Principal {
    public static void main(String[] args) {
        Vehiculo miVehiculo = new Vehiculo();
        miVehiculo.setnombreVehiculo("El Gran Vehiculo");
        System.out.println("usando miVehiculo, nombreVehiculo : "+miVehiculo.getnombreVehiculo());
        System.out.println("usando miVehiculo llama a : "+miVehiculo.transportar());
        System.out.println();

        Acuatiko miAcuatiko= new Acuatiko();
        miAcuatiko.setnombreVehiculo("El Acuático");
        System.out.println("usando miAcuatiko, nombreVehiculo : "+miAcuatiko.getnombreVehiculo());
        System.out.println("usando miAcuatiko llama a : "+miAcuatiko.transportar());
        System.out.println("usando miAcuatiko llama a : "+miAcuatiko.navegar());
        System.out.println();

        Barco miBarco=new Barco();
        miBarco.setnombreVehiculo("Titanic");
        System.out.println("usando miBarco, nombreVehiculo : "+miBarco.getnombreVehiculo());
        System.out.println("usando miBarco llama a : "+miBarco.transportar());
        System.out.println("usando miBarco llama a : "+miBarco.navegar());
        System.out.println("usando miBarco llama a : "+miBarco.prenderMotor());
```

```

System.out.println();

Velero miVelero=new Velero();
miVelero.setnombreVehiculo("Velero");
System.out.println("usando miVelero, nombreVehiculo : "+miVelero.getnombreVehiculo());
System.out.println("usando miVelero llama a : "+miVelero.transportar());
System.out.println("usando miVelero llama a : "+miVelero.navegar());
System.out.println("usando miVelero llama a : "+miVelero.izarVelas());
System.out.println();

Aereo miAereo= new Aereo();
miAereo.setnombreVehiculo("El Aereo");
System.out.println("usando miAereo, nombreVehiculo : "+miAereo.getnombreVehiculo());
System.out.println("usando miAereo llama a : "+miAereo.transportar());
System.out.println("usando miAereo llama a : "+miAereo.volar());
System.out.println();

Avion miAvion=new Avion();
miAvion.setnombreVehiculo("El Condor");
System.out.println("usando miAvion, nombreVehiculo : "+miAvion.getnombreVehiculo());
System.out.println("usando miAvion llama a : "+miAvion.transportar());
System.out.println("usando miAvion llama a : "+miAvion.volar());
System.out.println("usando miAvion llama a : "+miAvion.bajarTrenDeAterrizaje());
System.out.println();

Helicoptero miHelicoptero=new Helicoptero();
miHelicoptero.setnombreVehiculo("El lobo del Aire");
System.out.println("usando miHelicoptero, nombreVehiculo : "+miHelicoptero.getnombreVehiculo());
System.out.println("usando miHelicoptero llama a : "+miHelicoptero.transportar());
System.out.println("usando miHelicoptero llama a : "+miHelicoptero.volar());
System.out.println("usando miHelicoptero llama a : "+miHelicoptero.encenderHelices());
System.out.println();

System.out.println("nombre Vehiculo : "+miVehiculo.getnombreVehiculo());
System.out.println("nombre Acuatico : "+miAcuatico.getnombreVehiculo());
System.out.println("nombre Aereo : "+miAereo.getnombreVehiculo());
System.out.println("nombre Barco : "+miBarco.getnombreVehiculo());
System.out.println("nombre Velero : "+miVelero.getnombreVehiculo());
System.out.println("nombre Avion : "+miAvion.getnombreVehiculo());
System.out.println("nombre Helicoptero : "+miHelicoptero.getnombreVehiculo());
System.out.println();

}
}

```

Realitza per a cada mètode de les diferents classes un codi semblant al següent:

```

public String transportar(){
    return "Metode transportar de classe Vehicle";
}

```