

RESUM D'HERÈNCIA BÀSICA EN JAVA.

DEFINICIONS PRÈVIES:

- **Herència:** es el mecanisme que proporcionen els llenguatges de programació orientats a objectes per a reutilitzar el disseny de classes ja existents per a definir noves classes.
- **Classe Base:** es la classe origen "pare" de les classes derivades o "filles". L'Herència va de Pares a Fills, però no de Fills a Pares. La classe base de totes les classes Java és **Object**
- **Classe Derivada:** es la classe que hereta de la classe base, se identifica en la seua capçalera indicant que s'estén a la classe base. Té accés a tot lo públic o protegit de la classe Base i pot afegir i modificar el que no és final.
- **extends:** és la paraula reservada per a indicar en la capçalera de una classe que hereta de altra.
- **protected:** es la paraula reservada per a indicar que un atribut es sols visible per a la classe base i les derivades.
- **super:** es la paraula reservada per a indicar en una classe derivada que se fa referència a la classe base. Per ejemplo en el constructor de la classe derivada es obligatori que siga la primera invocació, per a construir el objecte de la classe base i després afegir o modificar lo derivat. Si el constructor es invocat es el por defecte (super() sense paràmetres) el compilador l'afegeix automàticament. També serveix per a fer referència als mètodes de la classe base abans d'haver-los reescrit, per exemple super.toString().
- **Tipus Estàtic:** quan es defineix una variable de tipus referència es refereix a la classe que se escriu junt al nombre i que indica **QUÉ** es pot fer o vore en eixa variable. Ha de ser igual o més general que el tipus dinàmic. $TE \geq TD$
- **Tipus Dinàmic:** quan s'inicialitza una variable de tipus referència es refereix a la classe amb la que s'inicialitza dita variable i que indica **CÓM** es fan els mètodes que permet el tipus estàtic. Ha de ser igual o més específic que el tipus estàtic. $TD \leq TE$

EXEMPLE: (COMPLETAR AMB LES CLASSES RECTANGULO, CUADRADO i OVALO)

```
public class Figura{
    // instance variables
    protected String color;
    protected int coordX, coordY;

    public Figura(String c, int x, int y){
        color=c; coordX=x; coordY=y;
    }

    public String getColor()      { return color; }
    public int getCoordX()        { return coordX; }
    public int getCoordY()        { return coordY; }
    public void setColor(String c) { color=c; }
    public void setCoordX(int x)   { coordX=x; }
    public void setCoordY(int y)   { coordY=y; }

    // LOS SIGUIENTES METODOS SE REESCRIBE COMPLETA O
    // PARCIALMENTE EN LAS CLASES DERIVADAS.
    public String toString(){
        return "Color "+color+
            " en las coordenadas "+
            coordX+" "+coordY+"\n";
    }

    public double area()      { return -1; }
    public double perimetro() { return -1; }
}
```

```
public class Circulo extends Figura{
    // instance variables
    protected double radio;

    public Circulo(String c, int x, int i, double r){
        super(c, x, y);
        radio=r;
    }

    public double getRadio()      { return radio; }
    public void setRadio(double r) { radio=r; }

    public double area(){
        return Math.PI*radio*radio;
    }

    public double perimetro(){
        return 2*Math.PI*radio;
    }

    public String toString() {
        return "Circulo de radio " + radio
            + " y de " + super.toString();
    }
}
```

En el programa principal, si hiciéramos:

```
Figura f1 = new Figura("verde", 10, 10);
Circulo c1 = new Circulo("rojo", 20,10, 5.0)
Figura f2 = new Circulo("Azul", 4,11, 5.0)
Circulo c2 = new Figura("negro", 20,10, 5.0)
System.out.println(f1.getColor());
System.out.println(c1.getColor());
System.out.println(f2.getRadio());
System.out.println(c1.getRadio());
System.out.println(f1.getArea());
System.out.println(c1.getArea());
System.out.println(f2.getArea());
```

"T.D.: COM HO FA"

"T.E.: QUÉ POT FER"

- ✓
- ✓
- ✓
- ✗ ERROR de COMPILACION: T.D. > T.E.
- ✓ escriu "verde" en la salida estándar
- ✓ escriu "rojo" en la salida estándar
- ✗ ERROR de COMPILACION: getRadio() no definido
- ✓ escriu "azul" en la salida estándar
- ✓ escriu "-1" en la salida estándar
- ✓ escriu "78.5398..." en la salida estándar
- ✓ escriu "78.5398..." en la salida estándar