

PROGRAMMING  
LANGUAGE

# JAVA

FILE HANDLING  
FUNCTIONS  
MULTITHREADING  
OPERATORS  
OOP  
**BASICS**  
INHERITANCE  
**OPERATORS**  
ARRAYS  
WRAPPER CLASSES  
JAVA PACKAGES

## T9 – Herencia en JAVA

# T9 – Herencia en JAVA



Crea la super - class Electrodomésticos con las clases hijas Lavadora y Televisión.

Sigue las relaciones requeridas.

```
public class TA9Ej1_Electrodomestico {

    // Atributos (o campos/propiedades)
    protected double precioBase;
    protected colorE color;
    protected consumoE consumo;
    protected int peso;

    // CONSTANTES
    private final static double precioBase_DEFAULT = 100;
    private final static colorE color_DEFAULT = colorE.BLANCO;
    enum colorE {
        BLANCO, NEGRO, ROJO, AZUL, GRIS
    }
    private final static consumoE consumo_DEFAULT = consumoE.F;
    enum consumoE {
        A, B, C, D, E, F
    }
    private final static int peso_DEFAULT = 5;

    // Constructor por defecto
    public TA9Ej1_Electrodomestico() {
        this.precioBase = precioBase_DEFAULT;
        this.color = color_DEFAULT;
        this.consumo = consumo_DEFAULT;
        this.peso = peso_DEFAULT;
    }
}
```

```
public class TA9Ej1_Lavadora extends TA9Ej1_Electrodomestico{

    // Atributos (o campos/propiedades): heredados de superclase
    private int carga;

    // CONSTANTES
    private final static int carga_DEFAULT = 5;

    // Constructor por defecto
    public TA9Ej1_Lavadora() {

    }

    // Constructor precio+peso, resto defecto
    public TA9Ej1_Lavadora(double precio, int peso) {

    }

    // Constructor TODO
    public TA9Ej1_Lavadora(double precioBase, colorE color, consumoE consumo,
                           int peso, int carga) {
        super(precioBase, color, consumo, peso);
        this.carga = carga;
    }
}
```

```
public class TA9Ej1_Television extends TA9Ej1_Electrodomestico{

    // Atributos (o campos/propiedades): heredados de superclase
    private int resolucion;
    private boolean sinTDT;

    // CONSTANTES
    private final static int resolucion_DEFAULT = 20;
    private final static boolean sinTDT_DEFAULT = false;

    // Constructor por defecto
    public TA9Ej1_Television() {
        super();
        this.resolucion = resolucion_DEFAULT;
    }

    // Constructor precio+peso, resto defecto
    public TA9Ej1_Television(double precio, int peso) {

    }

    // Constructor TODO
    public TA9Ej1_Television(double precioBase, colorE color,
                             consumoE consumo, int peso, int resolucion) {
        super(precioBase, color, consumo, peso);
        this.resolucion = resolucion;
    }
}
```

## 1

# MainApp - Consola

```
public class TA9Ej1_MainApp {  
    public static void main(String[] args) {  
        TA9Ej1_Electrodomestico arrayE[] = new TA9Ej1_Electrodomestico [10];  
  
        // --- ELECTRODOMESTICOS ---  
        TA9Ej1_Electrodomestico elec1 = new TA9Ej1_Electrodomestico (800,  
            TA9Ej1_Electrodomestico.colorE.NEGRO, TA9Ej1_Electrodomestico.consumoE.A, 50);  
        System.out.println(elec1.ElectrodomesticoTA9_toString()+"\n");  
        arrayE[0] = elec1;
```

```
Electrodomestico{precio = 800.0, color = NEGRO, consumo = 'A', peso = 50}
```

```
// --- LAVADORAS ---  
TA9Ej1_Lavadora lavadora1 = new TA9Ej1_Lavadora (800, TA9Ej1_Lavadora.colorE.AZUL,  
    TA9Ej1_Lavadora.consumoE.C, 30, 15);  
System.out.println(lavadora1.TA9Ej1_Lavadora_toString()+"\n");  
arrayE[4] = lavadora1;
```

```
Lavadora{precio = 800.0, color = AZUL, consumo = 'C', peso = 30}
```

## 1

# MainApp - Consola

```
//      --- TELEVISION ---  
TA9Ej1_Television tv1 = new TA9Ej1_Television (800, TA9Ej1_Television.colorE.NEGRO,  
        TA9Ej1_Television.consumoE.C, 30, 15);  
    System.out.println(tv1.TA9Ej1_Television_toString()+"\n");  
    arrayE[7] = tv1;
```

```
TV {precio = 800.0, color = NEGRO, consumo = 'C', peso = 30}
```

```
double totalTV = 0;  
double totalLavadoras = 0;  
double totalElec = 0;  
  
for (TA9Ej1_Electrodomestico elec : arrayE) {  
    if (elec instanceof TA9Ej1_Television) {  
        totalTV += ((TA9Ej1_Television) elec).precioFinal();  
    } else if (elec instanceof TA9Ej1_Lavadora) {  
        totalLavadoras += ((TA9Ej1_Lavadora) elec).precioFinal();  
    } else {  
        totalElec += elec.precioFinal();  
    }  
}  
  
double total = totalElec + totalTV + totalLavadoras;
```

```
La suma de todas las televisiones es: 3500.0  
La suma de todas las lavadoras es: 3190.0  
La suma de todos los electrodomesticos es: 10010.0
```

2

Crea la interface Entregable y aplícala a las clases Serie y Videojuego.

Sigue las relaciones requeridas.

```
public interface TA9Ej2_Entregable {  
    void entregar();  
    void devolver();  
    boolean isEntregado();  
    int compareTo(Object a);  
  
}
```

```
public class TA9Ej2_Serie implements TA9Ej2_Entregable {  
    // Atributos  
    private String titulo;  
    private int numeroTemporadas;  
    private boolean entregado;  
    private String genero;  
    private String creador;
```

```
public class TA9Ej2_Videojuego implements TA9Ej2_Entregable{  
    // Atributos  
    private String titulo;  
    private int horasEstimadas;  
    private boolean entregado;  
    private String genero;  
    private String company;
```

## 2

# MainApp

```
public class TA9Ej2_MainApp {  
    public static void main(String[] args) {  
        TA9Ej2_Serie[] series = new TA9Ej2_Serie[5];  
        TA9Ej2_Videojuego[] videojuegos = new TA9Ej2_Videojuego[5];  
  
        // Agregamos las series y los videojuegos:  
        series[0] = new TA9Ej2_Serie("Doctor Who", 26, "Ciencia ficción",  
                                     "Sydney Newman, C. E. Webber, Donald Wilson");  
  
        videojuegos[0] = new TA9Ej2_Videojuego("Genshin Impact", 80,  
                                                "RPG /Mundo abierto", "Hoyoverse");  
  
        // "Entregamos" las series y videojuegos que queremos para compararlas:  
        series[0].entregar(); // Seran true  
        series[2].entregar();  
        videojuegos[1].entregar();  
        videojuegos[2].entregar();  
        videojuegos[3].entregar();  
        videojuegos[4].entregar();  
    }  
}
```

## 2

## MainApp – Consola

```
int seriesEntregadas = contarEntregados(series);  
System.out.println("Total de SERIES entregadas: " + seriesEntregadas);  
  
int videojuegosEntregados = contarEntregados(videojuegos);  
System.out.println("Total de VIDEOJUEGOS entregados: " + videojuegosEntregados);
```

```
Total de SERIES entregadas: 2  
Total de VIDEOJUEGOS entregados: 4
```

```
TA9Ej2_Videojuego videojuegoMasHoras = obtenerVideojuegoMasHoras(videojuegos);  
System.out.println("\nEl videojuego con más horas estimadas es:"  
    + "\n" +videojuegoMasHoras);  
  
TA9Ej2_Serie serieMasTemporadas = obtenerSerieMasTemporadas(series);  
System.out.println("\nLa serie con más temporadas es:"  
    + "\n" + serieMasTemporadas);
```

```
El videojuego con más horas estimadas es:  
Videojuego{titulo='Honkai Impact 3rd', horasEstimadas=100, entregado=true, genero='Acción'}  
  
La serie con más temporadas es:  
Serie{titulo='Doctor Who', numeroTemporadas=26, entregado=true, genero='Ciencia ficción'}
```



```
public class TA9Ej3_Libro {  
  
    private String isbn;  
    private String titulo;  
    private String autor;  
    private int nPaginas;  
  
    public TA9Ej3_Libro() {}  
  
    public TA9Ej3_Libro(String isbn, String titulo, String autor, int nPaginas) {  
        this.isbn = isbn;  
        this.titulo = titulo;  
        this.autor = autor;  
        this.nPaginas = nPaginas;  
    }  
}
```

```
public TA9Ej3_Libro compararPaginas(TA9Ej3_Libro otroLibro) {  
    int paginas1 = this.nPaginas;  
    int paginas2 = otroLibro.nPaginas;  
  
    if (paginas1 >= paginas2) {  
        return this;  
    }  
    return otroLibro;  
}
```

## 3

## MainApp – Consola

```
public class TA9Ej3_MainApp {  
    public static void main(String[] args) {  
        TA9Ej3_Libro libro1 = new TA9Ej3_Libro("12345", "El señor de los anillos",  
                                                "J.R.R. Tolkien", 1000);  
        TA9Ej3_Libro libro2 = new TA9Ej3_Libro("67890", "Harry Potter y la piedra filosofal",  
                                                "J.K. Rowling", 500);  
  
        System.out.println(libro1.toString());  
        System.out.println(libro2.toString());  
  
        System.out.println("\nEl libro con más páginas:\n"  
                           + libro1.compararPaginas(libro2).toString());  
    }  
}
```

```
El libro El señor de los anillos con ISBN 12345 creado por J.R.R. Tolkien tiene 1000 páginas.  
El libro Harry Potter y la piedra filosofal con ISBN 67890 creado por J.K. Rowling tiene 500 páginas.  
  
El libro con más páginas:  
El libro El señor de los anillos con ISBN 12345 creado por J.R.R. Tolkien tiene 1000 páginas.
```

Crea la class *Raíces* y utilízala para calcular la ecuación de segundo grado según los inputs que se aporten.

```
public class TA9Ej4_Raices {

    private double a;
    private double b;
    private double c;

    public TA9Ej4_Raices(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }
}
```

```
public void calcular() {
    if (tieneRaices()) {
        obtenerRaices();
    } else if (tieneRaiz()) {
        obtenerRaiz();
    } else {
        System.out.println("La ecuación no tiene solución.");
    }
}
```

```
public void obtenerRaices() {
    double resultado1 = 0, resultado2 = 0;

    resultado1 = (-b + Math.sqrt(getDiscriminante())) / (2 * a);
    resultado2 = (-b - Math.sqrt(getDiscriminante())) / (2 * a);

    System.out.println("Sol. 1: " + resultado1);
    System.out.println("Sol. 2: " + resultado2);
}
```

```
public void obtenerRaiz() {
    double resultado = (-b + Math.sqrt(getDiscriminante())) / (2 * a);
    System.out.println("Sol.: " + resultado);
}
```

## 4

# MainApp – Consola

```
import java.util.Scanner;

public class TA9Ej4_MainApp {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Introduce el valor de a:");
        double a = sc.nextDouble();
        System.out.println("Introduce el valor de b:");
        double b = sc.nextDouble();
        System.out.println("Introduce el valor de c:");
        double c = sc.nextDouble();

        TA9Ej4_Raices r = new TA9Ej4_Raices(a, b, c);
        r.calcular();
    }
}
```

```
Introduce el valor de a:
2
Introduce el valor de b:
6
Introduce el valor de c:
-3
Sol. 1: 0.4364916731037085
Sol. 2: -3.4364916731037085
```

5

Crea la super-class Persona y utilízala en sus hijas Alumno y Profesor. Crea la class Aula para relacionarlas.

Instancia los objetos Alumno y Profesor en el main y simula diversos objetos Aula.

```
public abstract class TA9Ej5_Persona {
    protected String nombre;
    protected int edad;
    protected Sexo sexo;
}
```

```
public class TA9Ej5_Profesor extends TA9Ej5_Persona {
    private Materia materia;
    private Sexo sexo;

    public TA9Ej5_Profesor() {
        super();
        this.materia = Materia.matemáticas;
    }
}
```

```
public class TA9Ej5_Alumno extends TA9Ej5_Persona {
    private int nota;

    public TA9Ej5_Alumno() {
        super();
        this.nota = generarNota();
    }
}
```

```
import java.util.Arrays;

public class TA9Ej5_Aula {
    private int id;
    private int maxEstudiantes;
    private Materia materia;
    private TA9Ej5_Alumno[] alumnos;
    private TA9Ej5_Profesor profesor;
```

```
public class TA9Ej5_Materia {
    public enum Materia {
        matemáticas, filosofía, física
    }
}
```

```
public class TA9Ej5_Sexo {
    public enum Sexo {
        H, M
    }
}
```

```
import TA9.TA9Ej5_Materia.Materia;

public class TA9Ej5_MainApp {

    public static void main(String[] args) {

        TA9Ej5_Profesor prof1 = new TA9Ej5_Profesor("Vicente", 50, Sexo.H, Materia.matemáticas);
        TA9Ej5_Profesor prof2 = new TA9Ej5_Profesor("Ana", 42, Sexo.M, Materia.filosofía);
        TA9Ej5_Profesor prof3 = new TA9Ej5_Profesor("Arnau", 38, Sexo.H, Materia.física);

        TA9Ej5_Alumno alumno1 = new TA9Ej5_Alumno("Paco", 20, Sexo.H);
        TA9Ej5_Alumno alumno2 = new TA9Ej5_Alumno("Vanesa", 21, Sexo.M);
        TA9Ej5_Alumno alumno3 = new TA9Ej5_Alumno("Miguel", 22, Sexo.H);
```

```
TA9Ej5_Alumno[] alumnos = { alumno1, alumno2, alumno3, alumno4, alumno5, alumno6,
    alumno7, alumno8, alumno9, alumno10 };

TA9Ej5_Aula aula1 = new TA9Ej5_Aula(1, 10, Materia.matemáticas,
    (TA9Ej5_Alumno[]) alumnos, (TA9Ej5_Profesor) prof1);
TA9Ej5_Aula aula2 = new TA9Ej5_Aula(2, 10, Materia.filosofía,
    (TA9Ej5_Alumno[]) alumnos, (TA9Ej5_Profesor) prof2);
TA9Ej5_Aula aula3 = new TA9Ej5_Aula(3, 10, Materia.física,
    (TA9Ej5_Alumno[]) alumnos, (TA9Ej5_Profesor) prof3);
TA9Ej5_Aula aula4 = new TA9Ej5_Aula(4, 10, Materia.física,
    (TA9Ej5_Alumno[]) alumnos, (TA9Ej5_Profesor) prof1);
```

## 5

## MainApp – Consola

```
System.out.println("Resultados del aula 1:");  
    aula1.entregarNotas();  
System.out.println("Resultados del aula 2:");  
    aula2.entregarNotas();  
System.out.println("Resultados del aula 3:");  
    aula3.entregarNotas();  
System.out.println("Resultados del aula 4:");  
    aula4.entregarNotas();
```

```
Resultados del aula 2:  
Hay 7 alumnos ausentes de 10  
No se ha podido dar clase
```

```
Resultados del aula 3:  
Hay 5 alumnos ausentes de 10  
No se ha podido dar clase
```

```
Resultados del aula 4:  
No se ha podido dar clase
```

Crea la class Cine y utiliza objetos de la clase Espectador para ocupar los objetos Butaca del cine, el cual presenta el Objeto Película. La entrada de los espectadores tiene restricciones (edad y precio entrada según dinero espectador). En el main, deberá de poder mostrarse el Cine y cómo éste queda ocupado.

```
public class TA9Ej6_Espectador {

    private String nombre;
    private int edad;
    private double dinero;

    public TA9Ej6_Espectador() {
        generarTA9Ej6_EspectadorRandom();
    }

    public TA9Ej6_Espectador(String nombre, int edad, double dinero) {}

    public void generarTA9Ej6_EspectadorRandom() {

        TA9Ej6_genteRandom[] nombres = TA9Ej6_genteRandom.values();
        int random = (int) (Math.random() * nombres.length);
        this.nombre = nombres[random].toString();

        random = (int) (Math.random() * 100);
        this.edad = random;

        double money = Math.random() * 100;
        this.dinero = money;

    }
}
```

```
public class TA9Ej6_Cine {

    private TA9Ej6_Peliculas pelicula;
    private double precio;
    private int filas;
    private int columnas;

    public TA9Ej6_Cine() {}

    public TA9Ej6_Cine(TA9Ej6_Peliculas pelicula, double precio,

    public void dibujarTA9Ej6_Cine() {
        for (int i = this.filas; i > 0; i--) {
            for (int j = 0; j < this.columnas; j++) {
                char c = (char) (65 + j);
                System.out.print(Integer.toString(i) + c + " ");
            }
            System.out.println();
        }
        System.out.println();
    }
}
```



```
public class TA9Ej6_Peliculas {  
  
    private String titulo;  
    private int duracion;  
    private int edadMinima;  
    private String director;  
  
    public TA9Ej6_Peliculas() {  
  
    public TA9Ej6_Peliculas(String titulo, int duracion,  
        int edadMinima, String director) {  
        this.titulo = titulo;  
        this.duracion = duracion;  
        this.edadMinima = edadMinima;  
        this.director = director;  
    }  
}
```

```
public class TA9Ej6_Butaca {  
  
    private int fila;  
    private char columna;  
    private boolean ocupada;  
  
    public TA9Ej6_Butaca() {  
  
    public TA9Ej6_Butaca(int fila, char columna,  
        boolean ocupada) {  
        this.fila = fila;  
        this.columna = columna;  
        this.ocupada = ocupada;  
    }  
}
```

## 6

## MainApp

```
import java.util.Scanner;

public class TA9Ej6_MainApp {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        TA9Ej6_Peliculas p = new TA9Ej6_Peliculas("Avatar", 162, 7, "James Cameron");
        TA9Ej6_Cine c = new TA9Ej6_Cine(p, 20, 8, 8);

        int capacidad = c.getColumnas() * c.getFilas();
        TA9Ej6_Butaca[][] asientos = new TA9Ej6_Butaca[c.getFilas()][c.getColumnas()];

        rellenarTA9Ej6_Butacas(asientos);
        dibujarTA9Ej6_Cine(asientos);

        System.out.println("Introduce el número de espectadores que van al cine:");
        int n = sc.nextInt();
        TA9Ej6_Espectador[] espectadores = new TA9Ej6_Espectador[n];

        generarTA9Ej6_Espectadores(n, espectadores);
        sentarTA9Ej6_Espectadores(espectadores, capacidad, c, asientos, p);
        System.out.println();
        dibujarTA9Ej6_Cine(asientos);
    }
}
```

```
public static int generarTA9Ej6_Butaca(int n) {
    // ...
}

public static void generarTA9Ej6_Espectadores(int n, TA9Ej6_Espectador[] espectadores) {
    // ...
}

public static void dibujarTA9Ej6_Cine(TA9Ej6_Cine c, TA9Ej6_Butaca[][] asientos) {
    // ...
}

public static void rellenarTA9Ej6_Butacas(TA9Ej6_Cine c, TA9Ej6_Butaca[][] asientos) {
    // ...
}
```

```
public static boolean puedeSentarse(TA9Ej6_Espectador e, TA9Ej6_Butaca b) {
    // ...
}

public static boolean tieneDinero(TA9Ej6_Espectador e) {
    // ...
}

public static boolean hayEspacio(TA9Ej6_Cine c, TA9Ej6_Butaca[][] asientos) {
    // ...
}

public static boolean tieneEdad(TA9Ej6_Espectador e) {
    // ...
}
```

## 6

## MainApp – Consola

```
8A( ) 8B( ) 8C( ) 8D( ) 8E( ) 8F( ) 8G( ) 8H( )
7A( ) 7B( ) 7C( ) 7D( ) 7E( ) 7F( ) 7G( ) 7H( )
6A( ) 6B( ) 6C( ) 6D( ) 6E( ) 6F( ) 6G( ) 6H( )
5A( ) 5B( ) 5C( ) 5D( ) 5E( ) 5F( ) 5G( ) 5H( )
4A( ) 4B( ) 4C( ) 4D( ) 4E( ) 4F( ) 4G( ) 4H( )
3A( ) 3B( ) 3C( ) 3D( ) 3E( ) 3F( ) 3G( ) 3H( )
2A( ) 2B( ) 2C( ) 2D( ) 2E( ) 2F( ) 2G( ) 2H( )
1A( ) 1B( ) 1C( ) 1D( ) 1E( ) 1F( ) 1G( ) 1H( )
```

Introduce el número de espectadores que van al cine:  
72

JoseAntonio no tiene suficiente dinero.  
Raul no tiene suficiente dinero.  
Adrian no tiene suficiente dinero.  
Jesus no tiene suficiente dinero.  
FranciscoJ no tiene suficiente edad.  
Pablo no tiene suficiente edad.  
Pedro no tiene suficiente dinero.  
FranciscoJ no tiene suficiente edad.  
MariaIsabel no tiene suficiente edad.  
Fernando no tiene suficiente dinero.  
MariaCarmen no tiene suficiente dinero.  
Josefa no tiene suficiente dinero.  
Fernando no tiene suficiente dinero.  
Ana no tiene suficiente dinero.  
Adrian no tiene suficiente dinero.  
Enrique no tiene suficiente dinero.

```
8A(X) 8B( ) 8C(X) 8D(X) 8E(X) 8F(X) 8G(X) 8H(X)
7A(X) 7B( ) 7C(X) 7D(X) 7E( ) 7F(X) 7G(X) 7H(X)
6A(X) 6B(X) 6C(X) 6D( ) 6E(X) 6F(X) 6G( ) 6H(X)
5A(X) 5B( ) 5C(X) 5D(X) 5E(X) 5F(X) 5G( ) 5H(X)
4A(X) 4B(X) 4C(X) 4D(X) 4E(X) 4F(X) 4G(X) 4H(X)
3A(X) 3B(X) 3C(X) 3D(X) 3E(X) 3F(X) 3G(X) 3H(X)
2A(X) 2B(X) 2C(X) 2D(X) 2E(X) 2F(X) 2G(X) 2H(X)
1A(X) 1B(X) 1C(X) 1D(X) 1E( ) 1F(X) 1G(X) 1H(X)
```