



## TEMA 3

# El Concepto De Objeto y Clase

V1.2

Manuel Pereira González

## Agenda



- **Encapsulamiento y Reutilización**
- Introducción a Objetos y Clases
- Atributos y Métodos
- Package e Import
- Instanciación y Destrucción
- Variables y Métodos de clase
- Constantes
- Resumen



# Encapsulamiento y Reutilización



- Nivel de abstracción -> **Encapsulamiento**
- “Caja Negra”, sólo es necesario conocer su **interfaz** con el exterior (atributos y métodos)
  - Facilita el **intercambio** de piezas (clases) por evolución o corrección de errores sin afectar al resto del código.
  - Facilita la **reutilización** de componentes: No es necesario “reinventar la rueda”

# Agenda



- Encapsulamiento y Reutilización
- **Introducción a Objetos y Clases**
- Atributos y Métodos
- Package e Import
- Instanciación y Destrucción
- Variables y Métodos de clase
- Constantes
- Resumen



# Introducción a Objetos y Clases



- Clase = Tipo de Objeto (Conjunto de objetos con el mismo comportamiento)
- Nombre de clase = Nombre de fichero

```
/**
 * Nombre:      Alumno.java
 * Descripción: Representa un alumno de la asignatura de programación
 * Autor:       Manuel Pereira
 */

public class Alumno {
}
```

## Agenda



- Encapsulamiento y Reutilización
- Introducción a Objetos y Clases
- **Atributos y Métodos**
- Package e Import
- Instanciación y Destrucción
- Variables y Métodos de clase
- Constantes
- Resumen



# Atributos y Métodos



- Atributos = Propiedades de la clase

```
/**
 * Nombre: Alumno.java
 * Descripción: Representa un alumno de la asignatura de programación
 * Autor: Manuel Pereira
 */

public class Alumno {
    public String nombre;
    public String apellidos;
    public int edad;

    public float notaTeoria;
    public float notaPractica1;
    public float notaPractica2;
    public float notaPractica3;
    public float notaParcial1;
    public float notaParcial2;
    public float notaParcial3;
}
```

# Atributos y Métodos



- Métodos =  
funciones que se  
pueden invocar  
sobre un objeto de  
la clase

```
public class Alumno {
    public String nombre;
    public String apellidos;
    public int edad;

    public float notaTeoria;
    public float notaPractica1;
    public float notaPractica2;
    public float notaPractica3;
    public float notaParcial1;
    public float notaParcial2;
    public float notaParcial3;
    public float notaJunio;
    public float notaSeptiembre;

    public float notaFinal(boolean estamosEnSeptiembre) {
        float notaParciales = notaPractica1 + notaPractica2
            + notaPractica3 + notaParcial1 + notaParcial2
            + notaParcial3 + (notaJunio * 0.4f);
        if(estaEnSeptiembre) {
            if(notaParciales > notaSeptiembre) {
                return notaParciales;
            } else {
                return notaSeptiembre;
            }
        } else {
            if(notaParciales > notaJunio) {
                return notaParciales;
            } else {
                return notaJunio;
            }
        }
    }
}
```

# Atributos y Métodos



- Acceso a atributos y métodos de una clase a través del operador punto "."

```
public class EjemploAlumnos {  
    public static void main(String[] args) {  
        Alumno pepe = new Alumno();  
        Alumno juan = new Alumno();  
        pepe.nombre = "José";  
        pepe.apellidos = "López Sánchez";  
        juan.nombre = "Juan";  
        juan.apellidos = "Rodríguez García";  
        System.out.println("La nota de " + pepe.nombre  
            + " es: " + pepe.notaFinal(true));  
    }  
}
```

```
public class Alumno {  
    public String nombre;  
    public String apellidos;  
    public int edad;  
  
    public float notaTeoria;  
    public float notaPractica1;  
    public float notaPractica2;  
    public float notaPractica3;  
    public float notaParcial1;  
    public float notaParcial2;  
    public float notaParcial3;  
    public float notaJunio;  
    public float notaSeptiembre;  
  
    public float notaFinal(boolean estamosEnSeptiembre) {  
        float notaParciales = notaPractica1 + notaPractica2  
            + notaPractica3 + notaParcial1 + notaParcial2  
            + notaParcial3 + (notaJunio * 0.4f);  
        if(estaamosEnSeptiembre) {  
            if(notaParciales > notaSeptiembre) {  
                return notaParciales;  
            } else {  
                return notaSeptiembre;  
            }  
        } else {  
            if(notaParciales > notaJunio) {  
                return notaParciales;  
            } else {  
                return notaJunio;  
            }  
        }  
    }  
}
```

# Agenda



- Encapsulamiento y Reutilización
- Introducción a Objetos y Clases
- Atributos y Métodos
- **Package e Import**
- Instanciación y Destrucción
- Variables y Métodos de clase
- Constantes
- Resumen



# Package



- Paquete (package) de la clase = Directorio del fichero
  - Estructura de directorios = Estructura de paquetes
  - Paquetes sirven para agrupar clases
  - Función: Evitar conflictos de símbolos
  - Cada clase pertenece a un package
  - Si no se define ningún package para una clase, pertenece al package por defecto (raíz)

```
package practicas.practical;  
  
public class Pieza {  
}
```

# Import



- Utilización de clases externas:
  - Del mismo paquete que la clase: No es necesario importarlas
  - De otros paquetes: Es necesario importarlas (declarar que se van a utilizar) al principio del fichero. Palabra clave **import**.
  - Se puede importar una clase, o todas las clases de un paquete.
    - Ej: **import java.util.ArrayList;**
    - Ej: **import java.util.\*;**

# Import



```
/**
 * Nombre: Alumno.java
 * Descripción: Representa una pieza del juego del Tetris
 * Autor: Manuel Pereira
 */

package practicas.practical;

import java.util.ArrayList;
import java.util.Hashtable;

public class Pieza {
    public ArrayList casillas;
    public Hashtable tablaInterna;

    public String dameCasilla(int posicion) {
        return (String) casillas.get(posicion);
    }
}
```

# Agenda



- Encapsulamiento y Reutilización
- Introducción a Objetos y Clases
- Atributos y Métodos
- Package e Import
- **Instanciación y Destrucción**
- Variables y Métodos de clase
- Constantes
- Resumen



# Instanciación



- Objeto = Instancia (ejemplar) de una clase
- Necesario crearlos (instanciarlos)

| Clase          | Objeto                            |
|----------------|-----------------------------------|
| Persona        | José Menendez López               |
| Coche          | Citroen Xsara matrícula 4545BBS   |
| Perro          | "popi", el perro de José          |
| Montaña        | Mont Blanc                        |
| Teléfono Móvil | El móvil de José: Nokia 3310 rojo |

# Instanciación



- Crear/Instanciar un objeto de una determinada clase. Operador **new**.

```
public class EjemploAlumnos {  
    public static void main(String[] args) {  
        Alumno pepe = new Alumno();  
        Alumno juan = new Alumno();  
        pepe.nombre = "José";  
        pepe.apellidos = "López Sánchez";  
        juan.nombre = "Juan";  
        juan.apellidos = "Rodríguez García";  
        System.out.println("Nombre: " + pepe.nombre + ", apellidos: " + pepe.apellidos);  
        System.out.println("Nombre: " + juan.nombre + ", apellidos: " + juan.apellidos);  
    }  
}
```



# Instanciación



```
public class Nota {
    public int valor;
    public float desviacion;
}

public class ApuntanDistinto {
    public static void main(String[] args) {
        Nota nt;
        Nota nt2;
        nt = new Nota();
        nt2 = new Nota();
        nt.valor = 4;
        System.out.println("El valor es: " + nt.valor);
        System.out.println("El otro valor es: " + nt2.valor);
    }
}

public class ApuntanIgual {
    public static void main(String[] args) {
        Nota nt;
        Nota nt2;
        nt = new Nota();
        nt2 = nt;
        nt.valor = 4;
        System.out.println("El valor es: " + nt.valor);
        System.out.println("El otro valor es: " + nt2.valor);
    }
}
```

Simbolo del sistema

```
C:\Practicas\lecciones>java ApuntanDistinto
El valor es: 4
El otro valor es: 0
C:\Practicas\lecciones>
```

Simbolo del sistema

```
C:\Practicas\lecciones>java ApuntanIgual
El valor es: 4
El otro valor es: 4
C:\Practicas\lecciones>
```

# Instanciación - Constructores



- Constructores: Métodos que se ejecutan automáticamente al crear los objetos de una clase (al invocar a new).
- Si no se define ningún constructor, se crea uno por defecto (sin argumentos). Si se define alguno, el constructor por defecto no se crea automáticamente.

```
public class Alumno {
    public String nombre;
    public String apellidos;

    public Alumno (String nombreDelAlumno,
                   String apellidosDelAlumno) {
        nombre = nombreDelAlumno;
        apellidos = apellidosDelAlumno;
    }
}

public class EjemploAlumnos {
    public static void main(String[] args) {
        Alumno pepe = new Alumno("José", "López Sánchez");
        Alumno juan = new Alumno("Juan", "Rodríguez García");
        System.out.println("Nombre: " + pepe.nombre +
                           "Apellidos: " + pepe.apellidos);
    }
}
```

## Instanciación - this



- Variable **this**
  - Definida implícitamente en el cuerpo de los métodos
  - Referencia al objeto sobre el que se invoca el método

```
public class Alumno {  
    public String nombre;  
    public String apellidos;  
  
    public Alumno (String nombre, String apellidos) {  
        this.nombre = nombre;  
        this.apellidos = apellidos;  
    }  
}
```

## Instanciación - null



- Antes de asignarse una variable a un objeto creado (con el operador new), la variable no apunta a nada. En java, nada = **null**

```
public class EjemploNull {  
  
    /**  
     * @param args Argumentos recibidos por línea de parámetros  
     */  
  
    static Alumno a;  
  
    public static void main(String[] args) throws IOException {  
        System.out.println("El valor de la variable a es: " + a);  
        System.out.println("Pulsa una tecla...");  
        System.in.read();  
        System.out.println(a.nombre);  
    }  
}
```

Command Prompt

```
C:\Practicas>java EjemploNull  
El valor de la variable a es: null  
Pulsa una tecla...  
Exception in thread "main" java.lang.NullPointerException  
    at EjemploNull.main(EjemploNull.java:21)  
C:\Practicas>
```

# Instanciación – Inicialización de Variables



- Las variables de objeto y de clase pueden inicializarse en la propia declaración
- Las variables de objeto se inicializan a un valor por defecto si no se hace explícitamente
  - 0 para enteros, flotantes y caracteres
  - false para booleanos
  - null para objetos
- Las variables locales (de un método) **deben inicializarse antes de usarse**, si no se produce un error de compilación

# Instanciación – Inicialización de Variables



```
public class PruebaAnimales {  
  
    // Inicialización de variables de objeto  
    public int a = 0;  
    public ArrayList b = new ArrayList();  
    // c por defecto se inicializa a cero  
    public long c;  
  
    // Inicialización de variables de clase  
    public static float d = 1f;  
    public static String e = "Cadena";  
    // f por defecto se inicializa a cero  
    public static int f;  
    // g por defecto se inicializa a false  
    public static boolean g;  
  
    public static void main(String[] args) {  
        (new PruebaAnimales()).metodo();  
    }  
  
    public void metodo() {  
        int h;  
  
        // NO COMPILA!!! No se puede utilizar la variable a  
        // hasta que no se inicialice a algún valor  
        int i = h + 1;  
  
        System.out.println("Variables: " + a + ", " +  
            b + ", " + c + ", " + d + ", " + e + ", " +  
            f + ", " + g);  
    }  
}
```

Command Prompt

```
C:\Practicas\Programacion>java PruebaAnimales  
Variables: 0, [], 0, 1.0, Cadena, 0, false  
C:\Practicas\Programacion>
```

# Destrucción



- No es necesario liberar la memoria explícitamente: **Garbage Collector**
- Posibilidad de ejecutar código cuando el Garbage Collector elimina un objeto: método **finalize**

```
public class Alumno {  
    public String nombre;  
    public String apellidos;  
  
    public Alumno (String nombre, String apellidos) {  
        this.nombre = nombre;  
        this.apellidos = apellidos;  
    }  
  
    protected void finalize () {  
        // Código destructor aquí  
    }  
}
```

# Agenda



- Encapsulamiento y Reutilización
- Introducción a Objetos y Clases
- Atributos y Métodos
- Package e Import
- Instanciación y Destrucción
- **Variables y Métodos de clase**
- Constantes
- Resumen



# Variables y Métodos de clase



- Posibilidad de definir variables y objetos a nivel de clase (compartidos por todos los objetos de la clase). Uso de **static**.

```
public class Alumno {
    public String nombre;
    public String apellidos;
    public static int contadorAlumnos = 0;

    public Alumno (String nombre, String apellidos) {
        this.nombre = nombre;
        this.apellidos = apellidos;
        contadorAlumnos++;
    }

    protected void finalize() {
        contadorAlumnos--;
    }
}

public class EjemploAlumnos {
    public static void main(String[] args) {
        Alumno pepe = new Alumno("José", "López Sánchez");
        System.out.println("Contador: " + pepe.contadorAlumnos + " alumnos");
        Alumno juan = new Alumno("Juan", "Rodríguez García");
        System.out.println("Contador: " + juan.contadorAlumnos + " alumnos");
    }
}
```

```
C:\ Command Prompt
C:\Practicas>java EjemploAlumnos
Contador: 1 alumnos
Contador: 2 alumnos
C:\Practicas>
```

# Variables y Métodos de clase



- Se puede invocar un método de clase desde cualquier objeto de la clase, o sobre la propia clase

```
public class Alumno {
    public String nombre;
    public String apellidos;
    public static int contadorAlumnos = 0;

    public Alumno (String nombre, String apellidos) {
        this.nombre = nombre;
        this.apellidos = apellidos;
        contadorAlumnos++;
    }

    public static final void imprimirContador() {
        System.out.println("Contador: " + contadorAlumnos + " alumnos");
    }
}

public class EjemploAlumnos {
    public static void main(String[] args) {
        Alumno pepe = new Alumno("José", "López Sánchez");
        Alumno.imprimirContador();
        Alumno juan = new Alumno("Juan", "Rodríguez García");
        juan.imprimirContador();
    }
}
```

```
C:\ Command Prompt
C:\Practicas>java EjemploAlumnos
Contador: 1 alumnos
Contador: 2 alumnos
C:\Practicas>
```

## Variables y Métodos de clase



- Desde un método de clase no se puede hacer referencia a variables/métodos de objeto ni a **this**, porque el método no se invoca sobre un objeto determinado sino sobre la clase.

```
public class Alumno {  
    public String nombre;  
    public String apellidos;  
    public static int contadorAlumnos = 0;  
  
    public Alumno (String nombre, String apellidos) {  
        this.nombre = nombre;  
        this.apellidos = apellidos;  
        contadorAlumnos++;  
    }  
  
    public static final void imprimirContador() {  
        System.out.println("Contador: " + contadorAlumnos + " alumnos");  
        // ERROR: No se puede acceder a la variable de objeto "nombre"  
        System.out.println("Nombre del alumno: " + nombre);  
    }  
}
```

## Agenda



- Encapsulamiento y Reutilización
- Introducción a Objetos y Clases
- Atributos y Métodos
- Package e Import
- Instanciación y Destrucción
- Variables y Métodos de clase
- Constantes**
- Resumen



# Constantes



- Utilización de **final** para definir constantes
- Constantes a nivel de clase (**static final**) o a nivel de objeto (**final**)

```
public class Alumno {  
    public final String nombre;  
    public String apellidos;  
  
    public Alumno (String nombre, String apellidos) {  
        // Se inicializa el valor de la constante en el constructor  
        this.nombre = nombre;  
        this.apellidos = apellidos;  
    }  
  
    public void intentarCambiarNombre(String nuevoNombre) {  
        // ERROR: No se puede modificar el valor de la  
        //         constante de objeto "nombre"  
        nombre = nuevoNombre;  
    }  
}
```

# Constantes



- Constantes a nivel de clase (**static final**)

```
public class Colores {  
    public static final int ROJO = 1;  
    public static final int AMARILLO = 2;  
    public static final int VERDE = 3;  
}  
  
public class EjemploColores {  
    public static void main(String[] args) {  
        System.out.println("El número del color rojo es: " + Colores.ROJO);  
        System.out.println("El número del color verde es: " + Colores.VERDE);  
        System.out.println("El número del color amarillo es: " + Colores.AMARILLO);  
    }  
}
```

```
Command Prompt  
C:\Practicas>java EjemploColores  
El numero del color rojo es: 1  
El numero del color verde es: 3  
El numero del color amarillo es: 2  
C:\Practicas>_
```

# Agenda



- Encapsulamiento y Reutilización
- Introducción a Objetos y Clases
- Atributos y Métodos
- Package e Import
- Instanciación y Destrucción
- Variables y Métodos de clase
- Constantes
- **Resumen**



# Resumen



- |   |   |
|---|---|
| <ul style="list-style-type: none"><li>▪ Encapsulamiento y Reutilización<ul style="list-style-type: none"><li>▪ Facilitar intercambio y reutilización</li></ul></li><li>▪ Introducción a Objetos y Clases</li><li>▪ Atributos y Métodos<ul style="list-style-type: none"><li>▪ Atributos</li><li>▪ Métodos</li></ul></li><li>▪ Package e Import<ul style="list-style-type: none"><li>▪ Package<ul style="list-style-type: none"><li>▪ Package por defecto</li></ul></li><li>▪ Import</li></ul></li></ul> | <ul style="list-style-type: none"><li>▪ Instanciación y Destrucción<ul style="list-style-type: none"><li>▪ Instanciación<ul style="list-style-type: none"><li>▪ Constructores</li><li>▪ This</li><li>▪ null</li></ul></li><li>▪ Destrucción<ul style="list-style-type: none"><li>▪ Garbage Collector</li><li>▪ Finalize</li></ul></li></ul></li><li>▪ Variables y Métodos de clase<ul style="list-style-type: none"><li>▪ static</li></ul></li><li>▪ Constantes<ul style="list-style-type: none"><li>▪ final</li><li>▪ static final</li></ul></li></ul> |
|---|---|



## Resumen: Para más información



- **Clases y Objetos:**
  - [http://pisuerga.inf.ubu.es/Isi/Invest/Java/Tuto/II\\_5.htm](http://pisuerga.inf.ubu.es/Isi/Invest/Java/Tuto/II_5.htm)
- **Clases en Java (3 capítulos):**
  - [http://eees.ii.uam.es/alfonso/web\\_poo\\_04/teoria/material/clases\\_1.pdf](http://eees.ii.uam.es/alfonso/web_poo_04/teoria/material/clases_1.pdf)
  - [http://eees.ii.uam.es/alfonso/web\\_poo\\_04/teoria/material/clases\\_2.pdf](http://eees.ii.uam.es/alfonso/web_poo_04/teoria/material/clases_2.pdf)
  - [http://eees.ii.uam.es/alfonso/web\\_poo\\_04/teoria/material/clases\\_3.pdf](http://eees.ii.uam.es/alfonso/web_poo_04/teoria/material/clases_3.pdf)
- **Clases en Java**
  - <http://www.ii.uam.es/%7Ecastells/docencia/poo/3-objetos.pdf>