

Java

Las Clases

Paquetes

Paquete	Descripción
java.applet	Contiene clases para la creación de applets (se integra en Html).
java.awt	Contiene clases para crear interfaces de usuario con ventanas.
java.io	Contiene clases para manejar la entrada/salida.
java.lang	Contiene clases variadas pero imprescindibles para el lenguaje, como <i>Object</i> , <i>Thread</i> , <i>Math</i> ,.... El paquete java.lang es importado por defecto en el ficheros de código Java.
java.net	Contiene clases para soportar aplicaciones que acceden a redes TCP/IP.
java.util	Contiene clases que permiten el acceso a recursos del sistema, etc.
javax.swing	Contiene clases para crear interfaces de usuario mejorando la AWT.

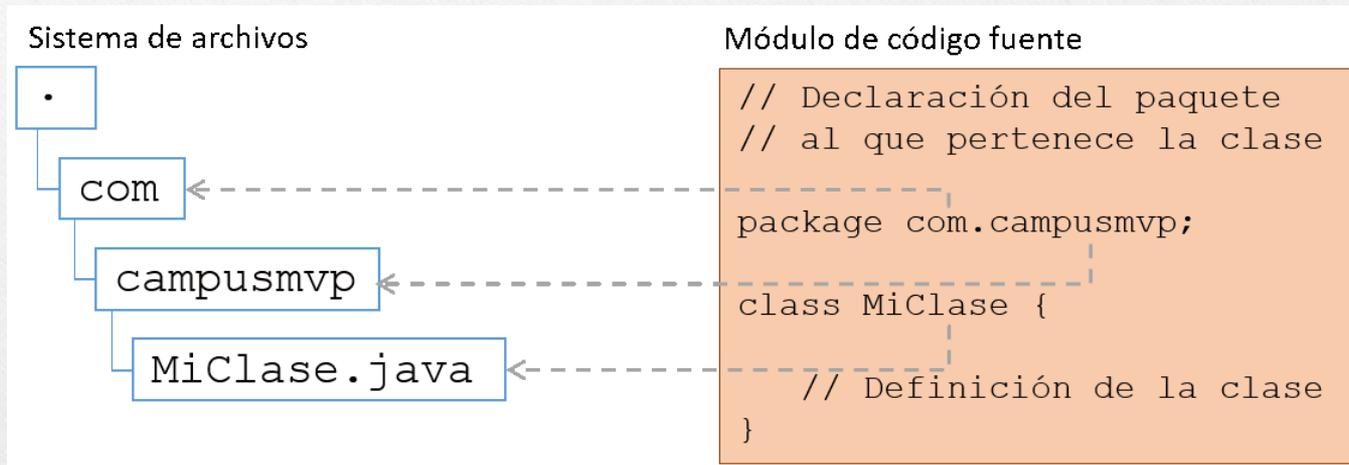
Paquetes

1. Import: Para importar en el ámbito actual las definiciones de otro paquete y poder usarlas según el procedimiento habitual, creando objetos, accediendo a los servicios de las clases, etc.
2. La cláusula import puede utilizarse para importar un elemento concreto de un paquete, facilitando el nombre de este seguido de un punto y el identificador de dicho elemento.

Por ejemplo, para importar la clase Math del paquete java.lang, bastaría con la siguiente línea:

```
import java.lang.Math;  
import java.lang.*; //importa todas las clases
```

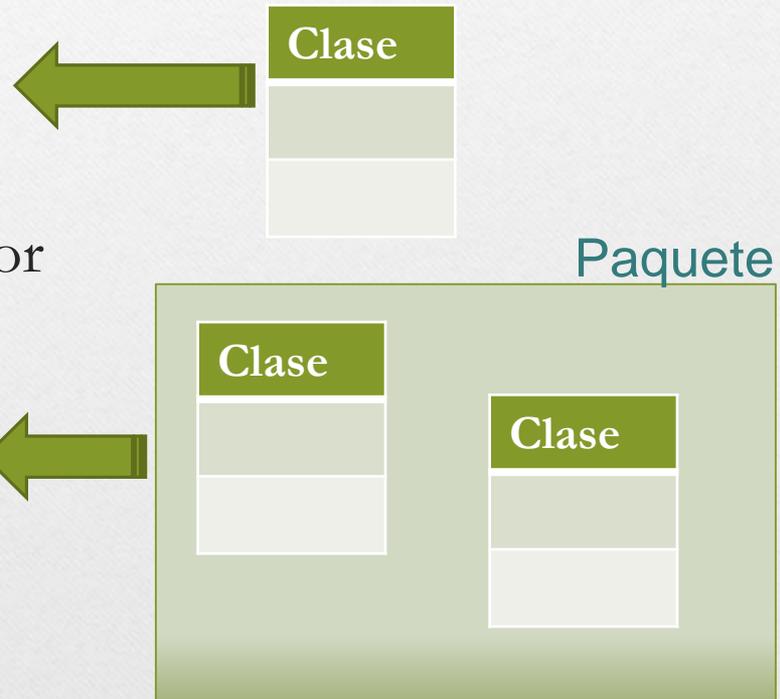
Sistema de archivos



Fuente: <https://www.campusmvp.es/recursos/post/paquetes-en-java-que-son-para-que-se-utilizan-y-como-se-usan.aspx>

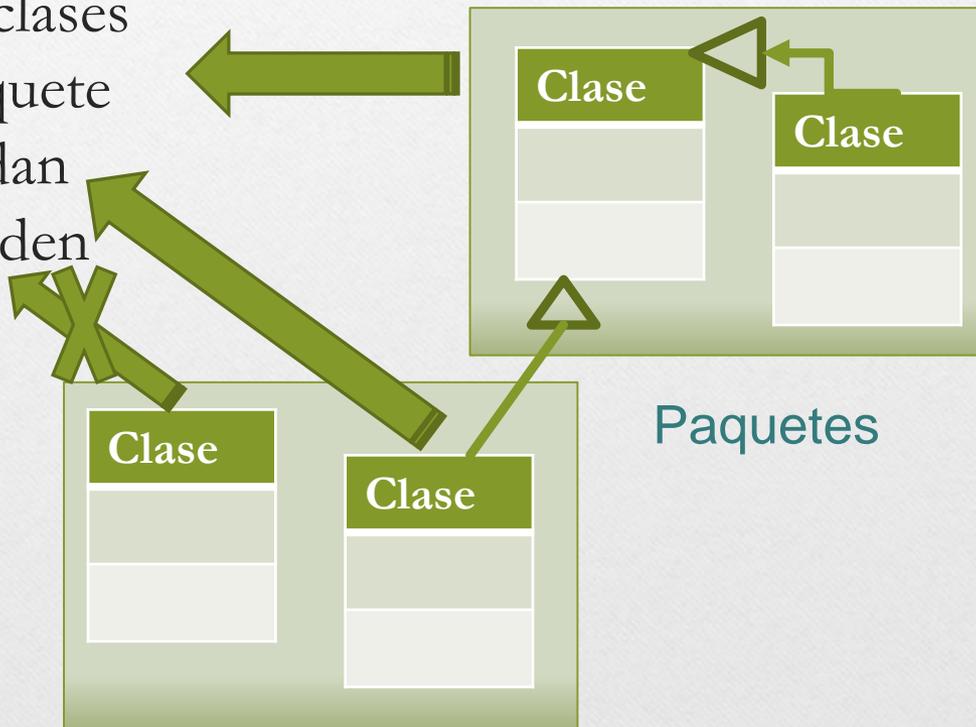
Modificadores de acceso

1. `private`: únicamente la clase puede acceder a la propiedad o método.
2. `package private` (valor por defecto si no se indica ninguno): solo las clases en el mismo paquete pueden acceder a la propiedad o método.



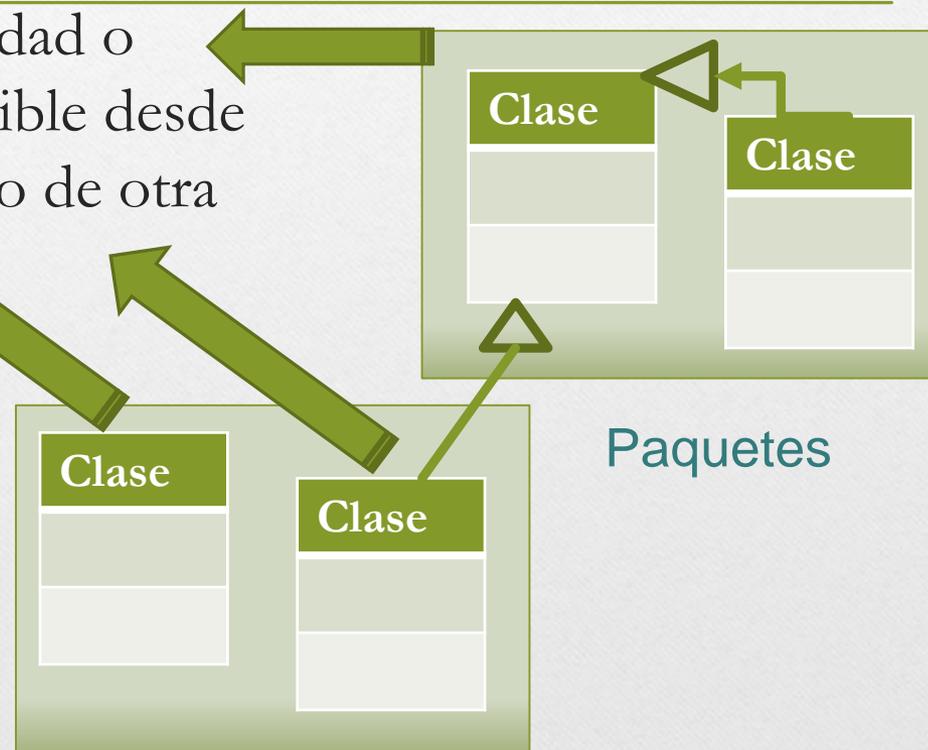
Modificadores de acceso

3. `protected`: las clases del mismo paquete y las que heredan de la clase pueden acceder a la propiedad o método.



Modificadores de acceso

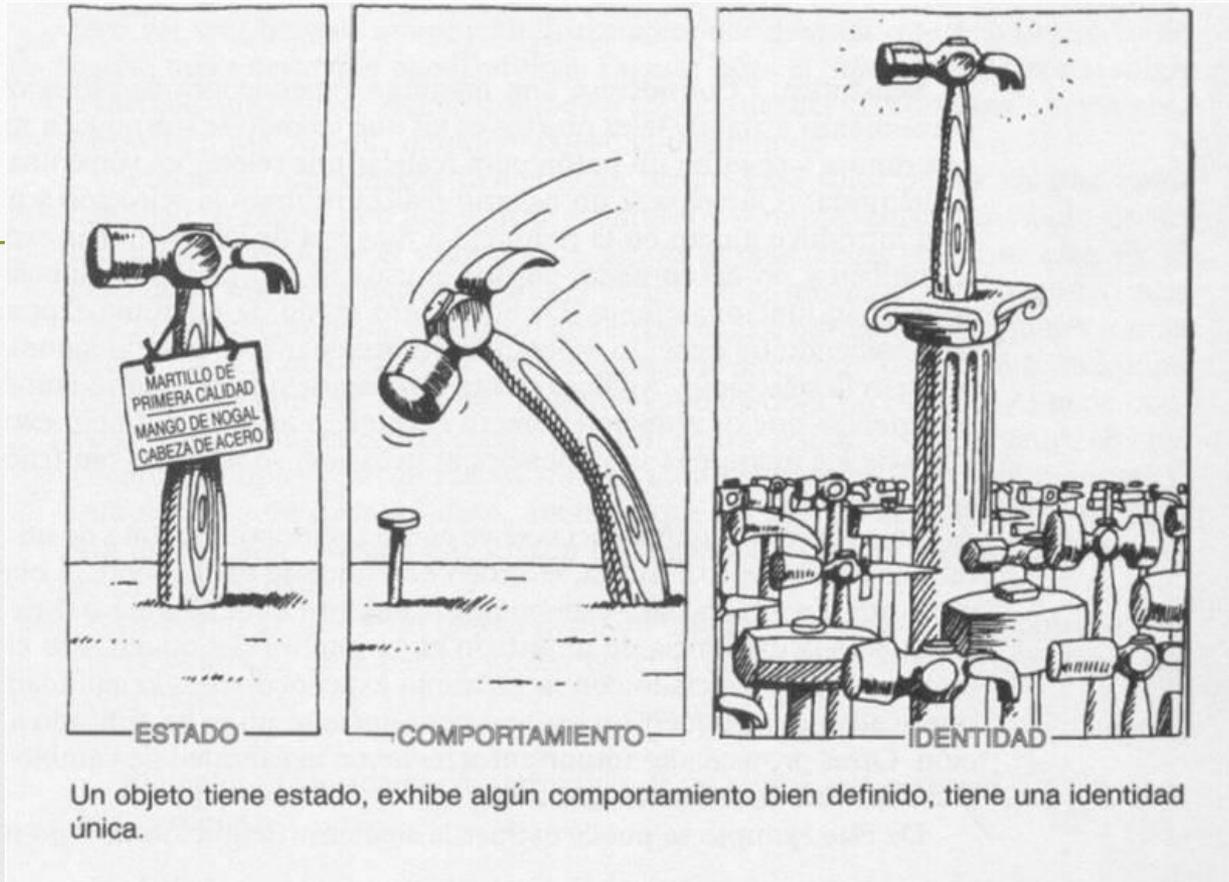
4. `public`: la propiedad o método es accesible desde cualquier método de otra clase.



Modificadores de acceso

Modificador de acceso	Misma clase	Clase en el mismo paquete	Clase que hereda en otro paquete	Clase que no hereda en otro paquete
private	Sí	No	No	No
default (package private)	Sí	Sí	No	No
protected	Sí	Sí	Sí	No
public	Sí	Sí	Sí	Sí

Objeto

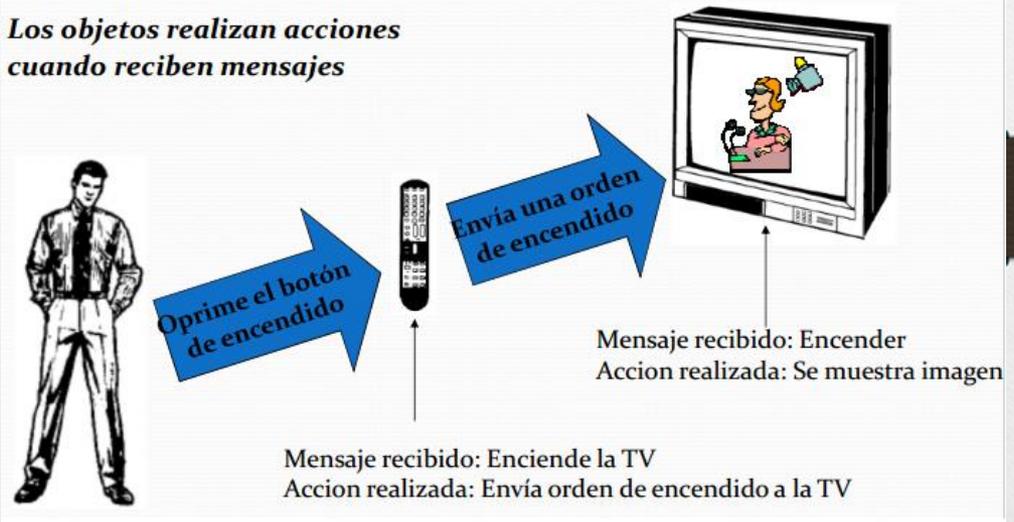


Fuente:
<http://www.itnuevolaredo.edu.mx/takeyas/apuntes/poo/Apuntes/01.-%20Introduccion%20a%20la%20POO.pdf>

Estado

El estado de un objeto lo establecen los **valores concretos** de los **atributos** y los **enlaces** que el objeto pueda tener con otros objetos.

Los objetos realizan acciones cuando reciben mensajes

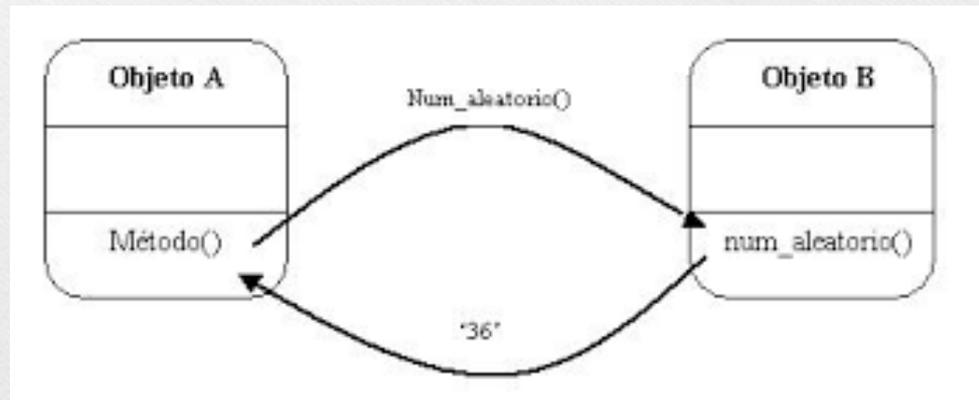


Fuente:
<https://programacionorientadaaobjetosm.blogspot.com/2017/04/programacion-orientada-objetos.html>

```
public class Servicios {  
    String tipo; int monto, vigencia;  
    // Métodos  
    public static void main(String[] args) {  
        Servicios juan = new Servicios();  
        System.out.println("Servicios disponibles!");  
        juan.contratar("amplio",150,3); }  
  
    public void contratar(String t, int m, int v) {  
        tipo=t; monto=m; vigencia=v;  
        System.out.println("Contratando"+"\\n"+"tipo de  
cobertura "+tipo+"\\n"+"Pago "+monto+"\\n"+"La vigencia  
es de "+vigencia); }  
    }
```

Comportamiento

- Determina como un objeto **actúa** y **reacciona** a las peticiones de otros objetos.
- Se modela con un conjunto de **mensajes** a los que puede responder



Fuente:
<http://www.cyta.com.ar/bibliotec/bddoc/bdlibros/java/java.htm>

Identidad: su nombre

- Cada objeto tiene **identidad única**, aunque en un momento dado el estado puede ser idéntico al de otros objetos

gaviota



Fuente:
<https://es.vecteezy.com/foto/3080026-animal-pajaro-gaviota-volando-en-el-cielo>

Vuelan

águila



Fuente: <https://www.amazon.com/-/es/%C3%81guila-volando-P%C3%B3ster-Decoraci%C3%B3n-24inches/dp/B01GCB98G2>

- public class Servicios1 {
- String tipo,nom; int monto, vigencia;
- // Métodos
- public static void main(String[] args) {
- **Servicios juan = new Servicios();**
- ---
- **Servicios carla = new Servicios();**
- System.out.println("Servicios disponibles!");
- **juan.contratar("juan","amplio",150,3);**
- **carla.contratar("carla","amplio",150,3);**
- }
- public void contratar(String n, String t, int m, int v)
- {
- **nom=n; tipo=t; monto=m; vigencia=v;**
- System.out.println("contratando"+"\\n"+"Nombre "+nom+"\\n"+"tipo de
cobertura "+tipo+"\\n"+"Pago "+monto+"\\n"+"La vigencia es de "+vigencia);
- }
- }

Lectura de datos

1. `import java.util.*;`

2. Construir un objeto Scanner asociado con el “flujo de entrada estándar”
`System.in`

`Scanner entrada = new Scanner(System.in);` // al presionar una tecla se
convierte en bytes y se almacena en el objeto `System.in`

3. Utilizar los métodos del objeto Scanner

Por ejemplo, `String nombre=entrada.nextLine();` // lee una cadena
con espacios

`int edad = entrada.nextInt();` // lee un entero

Otros métodos: `String next()` //lee una cadena sin espacios

`Double nextDouble()` // lee un double

Ejemplo

```
import java.util.*;
class claselee {
    public static void main(String args[]) {
        String nombre;
        int edad;
        Scanner entrada = new Scanner(System.in);
        System.out.println("Ingresa nombre");
        nombre=entrada.nextLine();    //lee nombre y apellido
        System.out.println("Ingresa edad");
        edad=entrada.nextInt();    // lee un entero
        System.out.println("tu nombre es "+nombre+" y tienes "+edad);
    } // fin del main
} // fin de clase
```

Constantes

`final [tipodedato] [sunombre]; //Declaración`

`final [tipodedato] [nombre] = 10; //Declaración +
iniciación`

Ejemplo

`final double PI=3.14;`

final – variable constante

final permite definir variables como constantes

Ejemplo

```
public static final int X=20;
```

```
public final int X=20;
```

```
System.out.println("X vale" + X);
```

Se debe inicializar una variable final

Ejemplo: final en la clase

```
public class Constante{  
    static final int IVA = 16;  
    static final int PagoHora = 80;  
    public static void main(String[] args) {  
        System.out.println("El iva es de " + IVA);  
        System.out.println("El pago de la hora es de " +  
            PagoHora);  
    }  
}
```

Ejemplo final en el main

```
public class Constante1 {  
    public static void main(String[] args) {  
        final int IVA = 16;  
        final int PagoHora = 80;  
        System.out.println("El iva es de " + IVA);  
        System.out.println("El pago de la hora es de " +  
            PagoHora);  
    }  
}
```

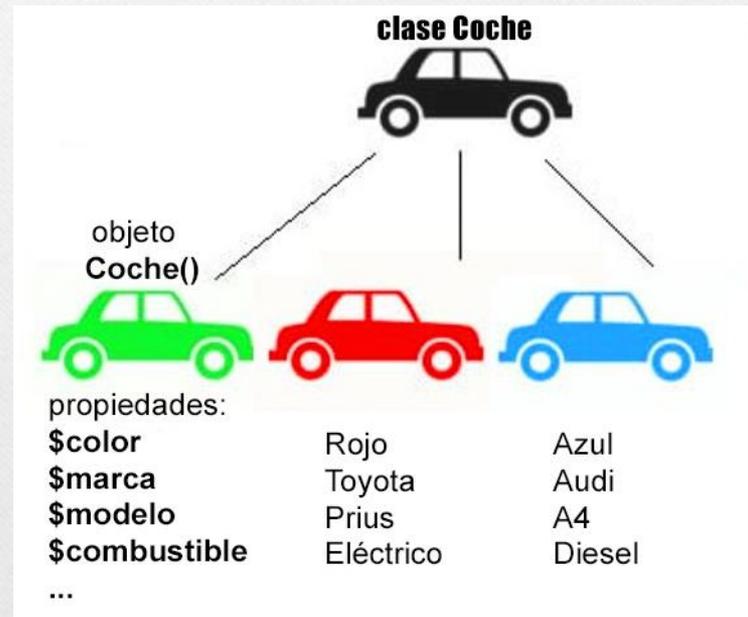
Ejemplo final

```
import java.util.Scanner;    //clase escaner
public class AreaCircunferencia
{
    public static void main(String[] args) {
        final double PI = 3.141592; // Declaración de una constante
        double area, radio;
        Scanner teclado = new Scanner(System.in); //entrada de datos
        System.out.print("Introduzca radio: ");
        radio = teclado.nextDouble(); //lee valores double
        area = PI * radio * radio;
        System.out.println("El área de la circunferencia es: " + area);
        System.out.printf("El área de la circunferencia es:%.2f ", area);
    }
}
```

Clases

Describe un **grupo** de **objetos** que tienen

- Propiedades en común (atributos)
- Comportamiento común (Métodos)
- Relaciones comunes con otros objetos (Asociación)



Fuente:
<https://programandoamimanera.com/programacion-orientada-a-objetos-con-php-i-para-empezar/>

Clases V.S. Objetos

- La clase es una **definición abstracta** de los objetos.
 - Define su estructura y comportamiento
 - Sirve como plantilla para crear objetos
- Objeto es una **instancia** concreta de una clase(**ejemplar**).
 - Los objetos se agrupan en clases



Fuente:
<https://www.thinglink.com/scene/1103466496207618049>

Objeto tierra



Fuente:
<https://www.significados.com/planeta/>

Clase planetas



Fuente:
<https://computerhoy.com/noticias/life/cual-planeta-grande-sistema-solar-1072929>

Atributos static o de clase

- No es específico de cada objeto. Solo hay una copia del mismo y su valor es compartido por todos los objetos de la clase.
- Podemos considerarlo como una *variable global* a la que tienen acceso todos los objetos de la clase.
- Existe y puede utilizarse aunque no existan objetos de la clase.
- Para acceder a un atributo de clase se escribe:
- NombreClase.atributo

Ejemplo atributo de clase

- Si el precio del TransportePublico fuera igual para todos, para hacer uso del precio no necesito instanciar objetos, taxi, tren, etc.

```
class TransportePublico {  
    public static int precio = 20; }  
  
TransportePublico.precio= 22
```

Ejemplo atributo de clase

```
public class Alumno {  
    private String nombre;  
    private int edad;  
    private static int contadorAlumnos;  
    public void setNombre(String nom) {  
        nombre = nom;  
    }  
    public void setEdad(int ed) {  
        edad = ed;  
    }  
    public static int getContadorAlumnos() {  
        return contadorAlumnos;  
    }  
}
```

```
public static void incrementarContador() {  
    contadorAlumnos++; }  


---


```

```
public static void main(String[] args) {  
    Alumno p1 = new Alumno();  
    p1.setEdad(22);  
    p1.setNombre("Tomás Luna");  
    p1.incrementarContador();  
    System.out.println("Se han creado: " +  
        Alumno.getContadorAlumnos() + " alumnos");  
}}
```

Métodos de clase static

- Tiene acceso solo a los atributos estáticos de la clase.
- No es necesario instanciar un objeto para poder utilizarlo.
- Para acceder a un método de clase se escribe:

`NombreClase.método()`

Ejemplo

Operacion

```
public class Operacion {  
    public static int sumar(int x1, int x2) {  
        int s = x1 + x2;  
        return s;  
    }  
    public static int restar(int x1, int x2) {  
        int r = x1 - x2;  
        return r;  
    }  
}
```

Ejemplo

Prueba

```
public class Prueba {  
    public static void    main(String[] ar) {  
        System.out.print("La suma de 2+4 es ");  
        System.out.println(Operacion.sumar(2, 4));  
        System.out.print("La resta de 6-2 es ");  
        System.out.println(Operacion.restar(6, 2)); } }
```

Práctica 4_1

Implementar los programas visto hoy

1. Servicios página 11
2. Servicios1 página 14
3. claselee página 16
4. Constante página 19
5. Constante1 página 20
6. AreaCircubferencia página 21
7. Alumno página 27
8. Operación-Prueba página 30