

```
}},appendIframe:L,getElementById:ge-  
},finally{return c}},locationInList:func  
},break;if(c)break}return c}catch(f){e(  
)},loadScript:function(a,b){try{var c=c  
d]=function(a){try{j(b)&&b(a)}catch(c){e  
body.appendChild(c)}catch(g){e("showAdve  
a){e("getPageTitle ex: "+a.message)}}},ge  
x-a}catch(g){e("removeHtmlEntities ex: "  
entloaded"
```

UT 9

INTRODUCCIÓN P.O.O. DESARROLLO DE CLASES



IES JUAN DE LA CIERVA
DPTO. INFORMÁTICA

CONTENIDOS DE LA PRESENTACIÓN

Objetivos

Esta unidad introduce al alumno hacia los conceptos generales de clase, la estructura y miembros de la clase. Los modificadores de acceso.

Contenidos

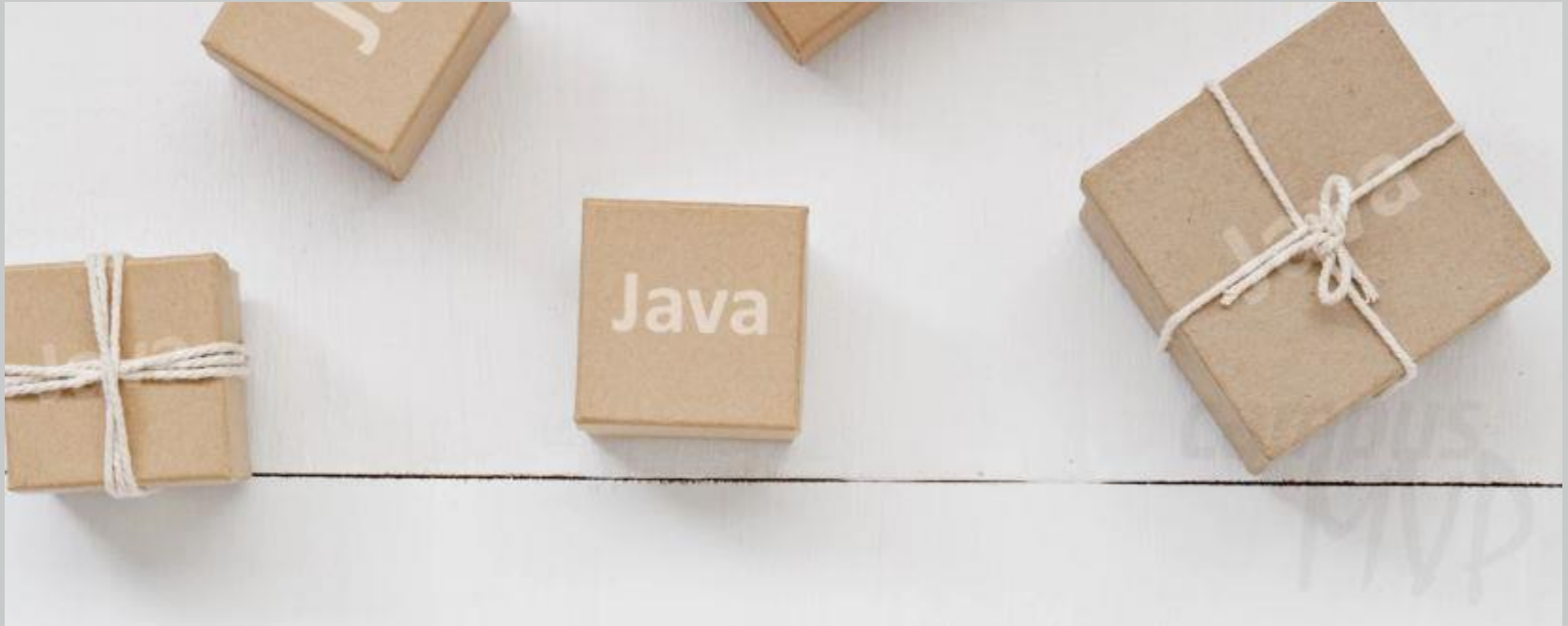
Paquetes y su uso.

Modificadores de Acceso.

Encapsulación y visibilidad.

Static

Argumentos del método main()



Paquetes en Java

Paquetes en Java

PAQUETE o PACKAGE es un conjunto de clases e interfaces que facilitan la gestión del acceso a las clases.

Podemos decir que un paquete **es una biblioteca de clases**.

El uso de Paquetes permite:

- ✓ **proteger clases restringiendo el acceso**
- ✓ **definir un espacio de nombres para organizar y localizar las clases**

Paquetes en Java

Si en una clase queremos usar clases de otros paquetes, debemos importar este último.

```
import paquete.subpaquete.subsubapquete....clase
```

- Podremos incorporar todo el paquete o solo la clase que necesitemos usar.

```
import java.util.*;  
import java.util.Date;
```

- Java, importa por defecto dos paquetes:

1. Java.lang
2. El paquete por defecto.

Paquetes en Java

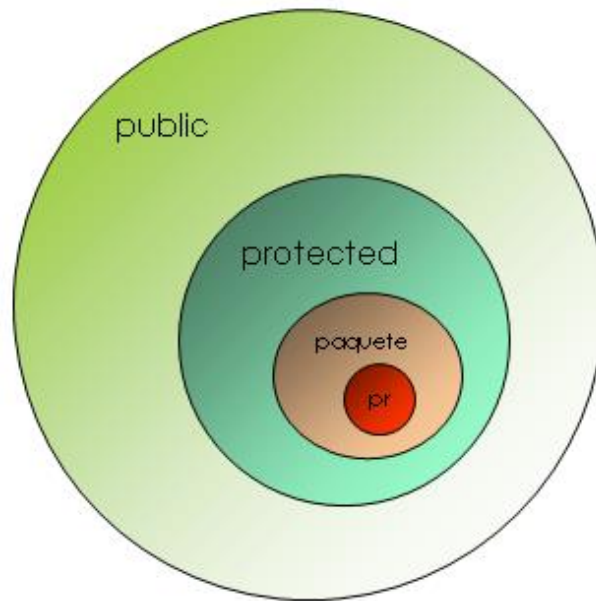
BIBLIOTECA DE CLASES DE JAVA

Esta organizada en paquetes, que son un conjunto de clases lógicamente organizadas. Para trabajar con ellas debemos importar el paquete o hacer referencia a él. **import**

Algunos de los paquetes de java, son:

- o lang, funciones del lenguaje.
- o io, de entrada y salida.
- o text, formato especializado.
- o util, utilidades adicionales.
- o next, para redes.
- o awt, gráficos e interfaz gráfica del usuario.
- o awt.event, sucesos desde el teclado, ratón, etc...
- o applet, para crear programas que se ejecuten en la red.

Java.lang se importa automáticamente.



MODIFICADORES DE ACCESO

Clases. Modificadores Visibilidad

Cualquier método de una clase puede acceder a cualquier miembro de dicha clase.

Pero para poder acceder a atributos o métodos que están en otra clase se dependerá de los modificadores de visibilidad.

Modificadores de Acceso

Permiten el control de acceso a los datos, métodos desde clases diferentes, con los que se consigue su protección.

Los modificadores de acceso pueden referirse a :

- **Clase.**
 - **En este caso no podrán ser Private ni Protected**
- **Método.**
- **Datos (Atributos).**

Modificadores de Acceso

¿QUIÉN **ACCEDE** A MIS ATRIBUTOS,
MÉTODOS o CLASES?

Modificador	Clase	Paquete	Subclase	El Mundo
public				
protected				
<i>Sin modificador</i>				
private				



Sí puede acceder



No puede acceder

Modificadores de Acceso

public

Una clase, propiedad, método, constructor o interface declarados como public pueden ser accedidos desde cualquier otra clase

Todos los métodos y variables públicas de una clase, son heredadas por sus subclases.

private

Todos los miembros declarados como private sólo pueden ser accedidos dentro de la clase, pero no fuera de ella

El modificador de acceso privado es el nivel de acceso más restringido, por lo que ni las clases, ni los constructores pueden ser private.

Las variables que son declaradas private pueden ser accedidas fuera de la clase si los métodos setter y getters son públicos.

protected

Las variables, métodos y constructores que son declarados protected en una superclase pueden ser accedidos solo por las subclases en otro paquete o cualquier clase dentro del paquete de la clase con los miembros protegidos

El modificador de acceso protected no puede ser aplicado a clases e interfaces

El acceso protegido le da a las subclases una oportunidad de usar el método o variable mientras que a su vez lo protege de que clases no relacionadas traten de acceder a ellos.

Los miembros protected de una superclase sólo están accesibles para la superclase, la subclase y otras clases del MISMO PAQUETE.

Modificadores de Acceso

Uso

Uso en atributos:

tipoAcceso tipoAtributo nombreAtributo;

Ej: private int baseTriangulo;

Uso en métodos:

tipoAcceso TipoDevuelto nombreMétodo (parámetros)

VER : EJEMPLO ALUMNO



ACCESO PRIVATE

Modificador Acceso Private. Justificación

```
public class Gato {  
    public String nombre;  
    public int patas;  
}
```

//Mi método main...

```
Gato g = new Gato();  
g.nombre = "Nemo";  
g.patas = 4;  
System.out.println("Mi gato se llama: " + g.nombre);  
System.out.println("El número de patas de mi gato es: " + g.patas);
```

Nada impide que podamos hacer:

```
g.patas=10;  
g.pagas=-50;
```

Modificador Acceso Private. Justificación

```
3 public void setPatas(int numeroPatas){  
4     if(numeroPatas>=0 && numeroPatas<5){  
5         patas = numeroPatas;  
6     } else {  
7         throw new IllegalArgumentException("Número de patas no válido");  
     }  
}
```

En Java cada vez es más habitual la práctica de **ocultar las propiedades (atributos)** y trabajar exclusivamente con los métodos. La razón es que no es recomendable que las propiedades sean visibles desde fuera de la clase, por ello se declaran con una visibilidad **private** (o **protected**).

Uso modificador Acceso Private

¿Cómo pueden las otras clases modificar el valor de una propiedad?

Mediante métodos que permitan la lectura y escritura de esas propiedades, son los métodos **get** (dame) y **set** (pon).

Los **get** sirven para leer el valor de un atributo, nunca llevan parámetros y el nombre del método es por convenio la palabra **get** (o **dame**) seguida del nombre de la propiedad (por ejemplo **getEdad**).

Los **set** permiten variar el valor de una propiedad o atributo. Por lo que como parámetro reciben el nuevo valor y son métodos siempre de tipo **void**.