

CURSO DE FUNDAMENTOS DE JAVA

PAQUETES EN JAVA



Por el experto: Ing. Ubaldo Acosta



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema de paquetes en Java.

¿Estás listo? Ok, ¡Vamos!

PAQUETES EN JAVA

**ORGANIZACIÓN
CLASES**

**LIMITAN EL
ACCESO A SUS
CLASES**



**EVITAN
CONFLICTOS
ENTRE
NOMBRES**

**HAY UNA
BIBLIOTECA
DE PAQUETES
EXISTENTES**

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

Los paquetes en Java permiten organizar nuestras clases. Podemos pensar en los paquetes como cualquier carpeta o folder que creamos para organizar documentos, archivos, fotos o cualquier tipo de información. En Java lo que organizamos son clases, y básicamente nos permite agrupar clases ya sea por su función, por herencia, o por cualquier característica que deseemos, lo importante es que tengamos una organización de nuestras clases, ya que solo en la versión estándar de Java SE encontramos más de 4000 clases, y por lo tanto es importante definir la forma en que organizaremos nuestras clases.

De igual manera, un proyecto en Java suele tener cientos o miles de clases, por lo tanto debemos organizar nuestras clases de la mejor forma, normalmente por la función que estas clases desempeñan en nuestro programa.

Al tener tantas clases, es normal que existan clases con el mismo nombre, sin embargo como cada clase pertenece a un paquete distinto, los paquetes también nos servirán para evitar problemas de nombres entre clases. Por ejemplo existen muchas clases Date en la versión estándar de Java, y para poder distinguir qué clase queremos utilizar debemos especificar el paquete Java donde se encuentra esta clase. Más adelante veremos un ejemplo de cómo hacer uso de los paquetes, y cómo Java nos permite utilizar las clases que pertenecen a dichos paquetes.

Otro de los usos de los paquetes es que limitan el acceso a los métodos o atributos de nuestras clases, ya que se limita según los modificadores de acceso que se utilicen. Este concepto lo veremos a detalle en el siguiente curso.

En la versión estándar de Java existen más de 4000 clases, por lo que ya existe un gran número de paquetes definidos, además de los que nosotros definamos. Todas estas clases están disponibles para ser utilizadas en nuestros programas Java, y más adelante veremos cómo podemos acceder a estas clases.

CONVENCIÓN EN NOMBRE DE PAQUETES EN JAVA

- Todo el nombre debe estar en minúsculas.
- Se acostumbra escribir el nombre del dominio web de manera invertida. Ej. Si el nombre del dominio es: `globalmentoring.com.mx`, entonces el nombre del paquete será: `mx.com.globalmentoring`
- Ejemplo de un proyecto: `mx.com.globalmentoring.miproyecto`
- Ejemplo de subpaquetes:

```
mx.com.globalmentoring.contabilidad.miproyecto  
mx.com.globalmentoring.administracion.miproyecto
```

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

El nombre de los paquetes en Java siguen un estándar.

El nombre del dominio no considera `www`, sólo es el nombre del dominio. Esto asegura que este nombre utilizado en internet es único. Si no tuviéramos un dominio, podemos asignar un nombre al proyecto y crear imaginariamente nuestro nombre de internet. Ej. `com.miproyecto`, y de esta manera si hubiera algún conflicto de nombres con otro proyecto, basta con modificar el nombre del proyecto a alguno que no tenga conflicto. La idea es que sigamos este estándar aunque no tengamos un dominio en internet, ya que este dominio web no es necesario que exista en internet, es sólo para evitar conflictos de nombre lo más posible.

Sin embargo podemos observar en la lámina un ejemplo del dominio `globalmentoring.com.mx`, el cual si queremos utilizarlo en nuestros proyectos Java podríamos usarlo, en primer lugar invirtiendo el orden del nombre del dominio, y posteriormente agregando el nombre de nuestro proyecto, con el objetivo de tener un nombre de paquete único. También vemos algunos ejemplos si tuviéramos un proyecto, y si ese proyecto tuviera subpaquetes, para distintas partes del mismo proyecto.

EJEMPLO DE CLASE DENTRO DE PAQUETE

Ejemplo de paquetes en Java:

```
package com.gm; //Definición del paquete
```

```
public class Utileria {  
    public static void imprimir(String s){  
        System.out.println("Imprimiendo mensaje: " + s);  
    }  
}
```

```
import com.gm.*; //Se importa el paquete a utilizar
```

```
public class EjemploPaquetes {  
  
    public static void main(String[] args) {  
        Utileria.imprimir("Hola"); //Se utiliza la clase importada  
    }  
}
```

Podemos observar en el ejemplo que primeramente creamos una clase llamada Utileria, esta clase la estamos agregando a un paquete llamado com.gm, pero puede ser cualquier nombre que deseen.

Una vez que hemos agregado una o más clases a este paquete, entonces creamos una clase en un paquete distinto. En este caso creamos la clase EjemploPaquetes en el paquete por default de Java, es decir, en ningún paquete. Cabe señalar que esta práctica no se recomienda, solo es para realizar el ejercicio. Java recomienda que todas las clases que creamos deben estar dentro de un paquete.

Ahora, para poder utilizar la clase Utileria definida en el paquete com.gm, lo que debemos hacer es utilizar la palabra import, la cual la podemos utilizar de dos maneras, una es importando todas las clases usando el *, o la otra es especificando el nombre de la clase, es decir: import com.gm.Utileria. Sin embargo esta última opción nos obliga a hacer un import por cada clase que deseemos utilizar del paquete com.gm, y si fueran muchas clases serian muchas líneas de código, por lo que en muchas ocasiones la notación import com.gm.* es la más utilizada. Cabe mencionar que el import no afecta la memoria, ya que no es que se carguen todas las clases de un paquete en memoria, una clase de carga en memoria hasta que se usa el nombre de la misma dentro del programa y no antes.

Finalmente podemos ver cómo estamos utilizando el método estático llamado imprimir, el cual pertenece a la clase Utileria. Observamos que ya no estamos indicando a qué paquete corresponde dicha clase, ya que hemos hecho el import. Sin embargo si hubiera más de una clase llamada Utileria, podríamos eliminar el import e indicar en la línea de código donde vamos a utilizar la clase, directamente el paquete al cual corresponde la clase, quedando com.gm.Utileria.imprimir("Hola"); con esto sabríamos exactamente a qué paquete corresponde la clase que estamos utilizando.

IMPORT ESTÁTICO

Ejemplo de import estático en Java:

```
package com.gm; //Definición del paquete

public class Utileria {
    public static void imprimir(String s){
        System.out.println("Imprimiendo mensaje: " + s);
    }
}

//Importamos el método estático a utilizar
import static com.gm.Utileria.imprimir;

public class EjemploPaquetes {

    public static void main(String[] args) {
        imprimir("Hola");
    }
}
```

En Java es posible importar los métodos estáticos a utilizar, de esta manera simplificamos la sintaxis en los métodos estáticos dentro de nuestro código.

En el ejemplo mostrado podemos observar que la clase Utileria y el paquete al que pertenecen son los mismos que en el ejemplo anterior, pero debido a que el método imprimir es un método estático, podemos aprovechar la sintaxis de import static que nos brinda java para importar el método estático a utilizar, y de esta manera se simplifica la sintaxis del uso del método imprimir, ya que como podemos observar el método imprimir ya no debe indicar a qué clase pertenece, sino que es suficiente con indicar el nombre del método, ya que en el import static ya se ha indicado que pertenece a la clase Utileria.

PAQUETES MÁS IMPORTANTES DE JAVA SE

Nombre del Paquete	Contenido
java.lang	Contiene clases esenciales, se importa implícitamente sin necesidad de una sentencia import
java.util	Contiene clases como la clase Date, entre muchas clases de utilerías más comunes en Java
java.io	input/output-Entrada/Salida. Clases que definen distintos flujos de datos
java.net	Se usa en combinación con las clases del paquete java.io para leer y/o escribir datos en la red.
java.applet	Contiene las clases necesarias para crear applets y se ejecutan en la ventana del navegador
java.awt	Contiene clases para crear una aplicación GUI (Graphic User Interface) independiente de plataforma

En la tabla mostrada, podemos observar algunos de los paquetes Java más comunes de la versión estándar. El más importante es el paquete java.lang, el cual contiene las clases core del lenguaje Java, tales como la clase Object, que es la clase de la cual descienden todas las clases en Java, entre varias clases más. Cabe señalar que no es necesario hacer import de ningún de las clases de este paquete, ya que es el único que se importa de manera automática por la importancia que tienen estas clases en la compilación y ejecución de nuestras clases Java.

Existen otros paquetes los cuales son también muy importantes, tales como el paquete java.util, java.net, y java.awt, java.ui y java.applet, entre muchos paquetes más. Conforme vayamos avanzando a lo largo del curso y cursos posteriores estaremos estudiando varias de las clases de estos paquetes.

EJERCICIOS CURSO FUNDAMENTOS DE JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Paquetes en Java.

CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

CURSO ONLINE

FUNDAMENTOS DE JAVA

Por: Ing. Ubaldo Acosta



CURSO DE FUNDAMENTOS DE JAVA

www.globalmentoring.com.mx

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- ✔ Lógica de Programación
- ✔ Fundamentos de Java
- ✔ Programación con Java
- ✔ Java con JDBC
- ✔ HTML, CSS y JavaScript
- ✔ Servlets y JSP's
- ✔ Struts Framework
- ✔ Hibernate Framework
- ✔ Spring Framework
- ✔ JavaServer Faces
- ✔ Java EE (EJB, JPA y Web Services)
- ✔ JBoss Administration
- ✔ Android con Java
- ✔ HTML5 y CSS3

Datos de Contacto:

Sitio Web: www.globalmentoring.com.mx

Email: informes@globalmentoring.com.mx

