

Paquetes y Documentación en Java

ELO329: Diseño y Programación Orientados a Objetos

Departamento de Electrónica

Universidad Técnica Federico Santa María

Manejo Creación de Paquetes en Java

Paquetes en Java (package)

- ❑ Existen para garantizar unicidad en los nombres de clases.
- ❑ También permiten organizar y controlar el acceso a clases e interfaces relacionadas.
- ❑ Como un proyecto puede contener muchas clases, hace sentido poner clases e interfaces relacionadas en un mismo directorio; es decir un mismo paquete.
- ❑ Si queremos referirnos a la clase Date, ya definida por Java, podemos usar:

```
java.util.Date hoydia = new java.util.Date();
```

- Una forma reducida es:

```
import java.util.Date; // al inicio del archivo  
Date hoydia = new Date();
```

- ❑ Si deseamos usar varias clases de un mismo paquete:

```
import java.util.*; // al inicio del archivo  
import java.*.*; // es error, ídem con import java.*
```

Uso de clases presentes en dos paquetes

- ❑ Si usamos la clase Date y tenemos:

```
import java.util.*;  
import java.sql.*;
```

- ❑ Hay problema porque ambos definen la clase Date.
- ❑ En este caso, además de lo previo, debemos agregar:

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

- Así el compilador sabe a qué clase Date nos referimos.

- ❑ ¿Qué pasa si necesitamos ambas versiones?

- Debemos usar el nombre completo cada vez:

```
java.util.Date plazo = new java.util.Date();  
java.sql.Date hoy = new java.sql.Date();
```

Creación de Paquetes en Java

- ❑ Para incluir una clase en un paquete, al inicio del archivo indicar:
`package cl.utfsm.elo.elo329 // este es un package`
- ❑ Esto implica que debe existir los directorios: cl, dentro de él utfsm, dentro de éste elo y finalmente el directorio elo329. En este último ponemos los archivos del paquete.
- ❑ Para compilar las clases del paquete, estando en el directorio superior - aquel donde cl es un directorio-, usamos:
`javac cl/utfsm/elo/elo329/unaClase.java`
- ❑ Para correr el main de una clase, ubicada en el directorio superior, usamos:
`java claseConMain // clase que ocupa clases del paquete.`
- ❑ Revisar ejemplo [PackageTest](#)
- ❑ Aquí puedes aprender más sobre paquetes.

Documentación de programas Java

Documentación (1/2)

- ❑ En java contamos con el utilitario **javadoc** para extraer documentación directamente desde el código fuente.
- ❑ Javadoc analiza el código y todos los comentarios entre **/**** y ***/**. Notar partida con dos “*” en lugar de uno.
- ❑ Poner la documentación de una clase inmediatamente antes de la clase.
- ❑ Para los métodos: usar los rótulos
 - @param variable descripción
 - @return descripción
 - @throws descripción de clase
- ❑ Para los datos públicos: **/** ...*/**
- ❑ Comentarios Generales:
 - @author nombre
 - @version texto
 - @since texto
 - @see link
 - Ejemplo: @see cl.utfsm.elo.Employee#raiseSalary(double)

Documentación (2/2)

❑ Se pueden usar todo tipo de rótulos html incrustados en los comentarios.

❑ ¿Cómo generar la documentación?:

`javadoc -d docDirectory *.java`

❑ Para la documentación de un paquete:

`javadoc -d docDirectory nameOfPackage`

❑ Ejemplo: [Account.java](#)

- index.html es generado con

`javadoc -d AccountDoc *.java`

❑ Hay varios otros rótulos, ver:

<https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html#tag>

❑ Revisar ejemplo [Account](#)