

Annex I.- Paquets i imports en Java

Un paquet (package) es un contenidor d'un grup de classes, interfícies (interface), etc. relacionades entre si.

Els avantatges d'organitzar els arxius en paquets són:

- Agrupar classes que tenen algo en comú
- Reutilització de codi.
- Seguretat gràcies als modificadors d'accés.

Encara que tornarem sobre els paquets més endavant, anem a veure uns pinzellades interessants.

Una classe pot fer-se membre d'un paquet mitjançant la clàusula **package**, la qual haurà de ser la primera línia de codi. Exemple:

```
package mipaquete;
```

```
class MiClase {
```

```
    ...
```

```
}
```

Tenim la classe MiClase del paquet mipaquete, de fet qualsevol altra classe declarada en aquest mateix arxiu de codi font pertanyerà també a mipaquete.

Recorda:

- Java es sensible a majúscules i minúscules. El que vol dir que "mipaquete" no és el mateix que "MiPaquete".
- No podem tindre en un mateix paquet dues classes amb el mateix nom, però si en diferents paquets.

Els paquets no són màgics, aquests estan representats físicament en el sistema, la jerarquia de paquets i sub-paquets correspon a l'existència dels seus corresponents directoris i sub-directoris, per exemple el paquet mipaquete.misubpaquete correspon al directori mipaquete / misubpaquete /, per tant podem dir el següent: Paquet = Directori. Per a l'execució del programa la JVM s'encarregarà de buscar els fitxers java corresponents en els directoris d'acord a la jerarquia indicada a través de les clàusules package.

Alguns paquets estandar més comuns són:

Paquet	Descripció
java.applet	Conté les classes necessaries per a crear applets que s'executen a la finestra del navegador web.
java.awt	Conté les classes per a crear interfícies d'usuari natives del sistema.
java.io	Entrada/Sortida. Classes que defineixen distints fluxes de dades.
java.lang	Classes essencials de Java. S'importa implícitament sense necessitat d'una sentència import.
java.net	S'utilitza amb combinació de les classes del paquet java.io per a llegir i escriure dades a la xarxa.

java.swing	Classes per a crear interfícies d'usuari independents del sistema.
java.util	Conté altres classes amb utilitats que ajuden al programador.

La sentència **import** permet referir-se a les classes d'un paquet sense tindre que posar el nom canònic complet. Es a dir, si fem el següent import:

```
import java.util.Scanner;
```

Ens permet poder utilitzar després només el nom de la classe de la següent forma:

```
Scanner lector = new Scanner(System.in);
```

En lloc de tindre que posar el nom complet del paquet cada vegada:

```
java.util.Scanner lector = new java.util.Scanner(System.in);
```

Quan fem un import podem:

- Importar una classe individual. Exemple: `import java.awt.Font;`
- Importar un conjunt de classes declarades dins d'un paquet, fent ús d'un asterisc (*). Exemple:
`import java.awt.*;`

Consideracions:

- La interacció entre diferents paquets es veu afectada pels modificadors d'accés, per això un dels avantatges de l'ús de paquets és la seguretat.
- La clàusula import no funciona com el "include" de C que inclou més codi, sinó que només indica al compilador la ubicació del codi extern el qual volem utilitzar, és més com un accés directe ja que com hem vist ni tan sols és obligatori fer-lo servir, simplement podríem escriure en el nostre codi les ubicacions completes del codi extern la qual cosa podria tornar-se repetitiu i ens deixaria un codi més extens. D'aquí l'avantatge pràctica en usar import.
- Els IDEs com ara NetBeans o Eclipse ens faciliten notòriament la tasca de declarar paquets i fer importacions.