

## 3.E. Librerías y paquetes.

Marcar como hecha

### 1. Librerías de objetos (paquetes).

#### 1.3. Jerarquía de paquetes.

Para organizar mejor las cosas, un **paquete**, en vez de clases, también puede contener otros paquetes. Es decir, podemos hacer **subpaquetes** de los paquetes y subpaquetes de los subpaquetes y así sucesivamente. Esto permite agrupar paquetes relacionados en un paquete más grande. Por ejemplo, si quiero dividir mis clases de ejemplos en ejemplos básicos y ejemplos avanzados, puedo poner más niveles de paquetes separando por puntos:

```
package ejemplos.basicos;
```

```
package ejemplos.avanzados;
```

A nivel de sistema operativo, tendríamos que **crear los subdirectorios** **basicos** y **avanzados** dentro del directorio **ejemplos**, y meter ahí las clases que correspondan.

Para compilar, en el directorio del proyecto habría que compilar poniendo todo el path hasta llegar a la clase. Es decir, el nombre de la clase va con todos los paquetes separados por puntos, por ejemplo **ejemplos.basicos.Bienvenida**.

La estructura de directorios en el sistema operativo cuando usamos subpaquetes sería:

```
/<directorio_usuario>/Proyecto_Bienvenida/ejemplos/basicos/HolaMundo.java
```

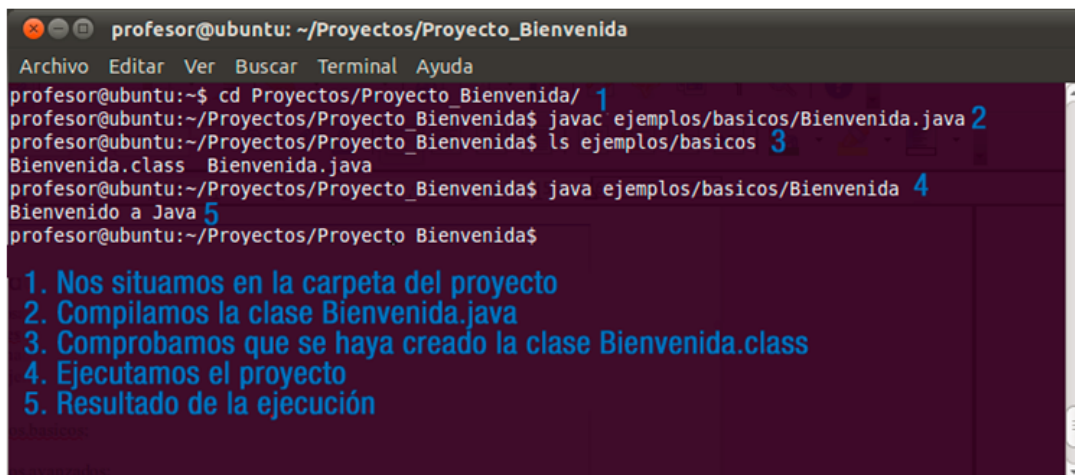
Y la compilación y ejecución sería:

```
$ cd /<directorio_usuario>/Proyecto_Bienvenida
```

```
$ javac ejemplos/basicos/Bienvenida.java
```

```
$ java ejemplos/basicos/Bienvenida
```

```
Hola Mundo
```



```
profesor@ubuntu: ~/Proyectos/Proyecto_Bienvenida
Archivo Editar Ver Buscar Terminal Ayuda
profesor@ubuntu:~$ cd Proyectos/Proyecto_Bienvenida/ 1
profesor@ubuntu:~/Proyectos/Proyecto_Bienvenida$ javac ejemplos/basicos/Bienvenida.java 2
profesor@ubuntu:~/Proyectos/Proyecto_Bienvenida$ ls ejemplos/basicos 3
Bienvenida.class  Bienvenida.java
profesor@ubuntu:~/Proyectos/Proyecto_Bienvenida$ java ejemplos/basicos/Bienvenida 4
Bienvenido a Java 5
profesor@ubuntu:~/Proyectos/Proyecto_Bienvenida$
```

1. Nos situamos en la carpeta del proyecto  
2. Compilamos la clase Bienvenida.java  
3. Comprobamos que se haya creado la clase Bienvenida.class  
4. Ejecutamos el proyecto  
5. Resultado de la ejecución

La Biblioteca de Clases de Java se organiza haciendo uso de esta jerarquía de paquetes. Así por ejemplo, si quiero acceder a la clase **Date**, tendré que importarla indicando su ruta completa, o sea, **java.util.Date**, así:

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

Los paquetes en Java pueden organizarse jerárquicamente de manera similar a lo que puedes encontrar en la estructura de carpetas en un dispositivo de almacenamiento, donde:

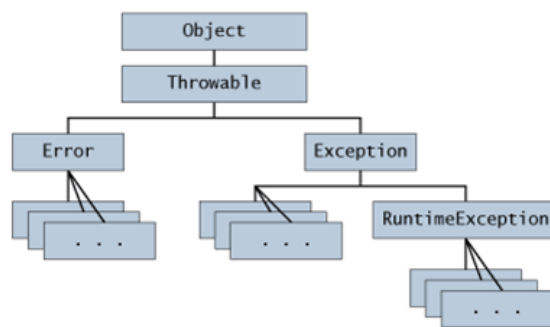
- Las clases serían como los archivos.
- Cada paquete sería como una carpeta que contiene archivos (clases).
- Cada paquete puede además contener otros paquetes (como las carpetas que contienen carpetas).
- Para poder hacer referencia a una clase dentro de una estructura de paquetes, habrá que indicar la trayectoria completa desde el paquete raíz de la jerarquía hasta el paquete en el que se encuentra la clase, indicando por último el nombre de la clase (como el path absoluto de un archivo).

La estructura de paquetes en Java permite organizar y clasificar las clases, **evitando conflictos de nombres y facilitando la ubicación** de una clase dentro de una estructura jerárquica.

Por otro lado, la organización en paquetes permite también el **control de acceso a miembros de las clases desde otras clases** que estén en el **mismo paquete** gracias a los **modificadores de acceso** (recuerda que uno de los modificadores que viste era precisamente el de paquete).

Las clases que forman parte de la jerarquía de clases de Java se encuentran organizadas en diversos paquetes.

Todas las clases proporcionadas por Java en sus bibliotecas son miembros de distintos paquetes y se encuentran organizadas jerárquicamente. Dentro de cada paquete habrá un conjunto de clases con algún tipo de relación entre ellas. **Se dice que todo ese conjunto de paquetes forman la API de Java**. Por ejemplo las clases básicas del lenguaje se encuentran en el paquete `java.lang`, las clases de entrada/salida las podrás encontrar en el paquete `java.io` y en el paquete `java.math` podrás observar algunas clases para trabajar con números grandes y de gran precisión.



#### Para saber más

Puedes echar un vistazo a toda la jerarquía de paquetes de la API de Java en la documentación oficial de Oracle (en inglés):

[Java Platform, Standard Edition 8 API Specification.](#)

◀ 3.D. Métodos.

Ir a...

3.F. Entrada y salida de datos por consola. ▶

