

Práctica No. 4. Classpath.

Objetivo:

Aprender a establecer el `classpath` para poder compilar una aplicación Java en donde la clase principal y las clases secundarias están en archivos y ubicaciones distintas.

Palabras clave:

Clases, Compilación, `Classpath`.

Introducción:

Las clases en Java están organizadas en paquetes, los cuales están organizados en directorios en el sistema de archivos. Pero, a diferencia del sistema de archivos, cuando tu especificas un nombre de paquete, especificas, completamente, el nombre del paquete y nunca una parte de él. Por ejemplo, el nombre del paquete para la clase `java.awt.Button` siempre es `java.awt`.

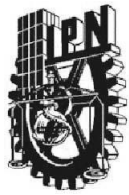
Por ejemplo, supongamos que deseamos que Java en tiempo de ejecución encuentre una clase llamada `Cool.class` en el paquete `utility.myapp`. Si la ruta a ese directorio es `C:\java\MyClasses\utility\myapp` se tiene que establecer la ruta de la(s) clase(s) (`classpath`) para que contenga el directorio `C:\java\MyClasses`.

Para ejecutar esa aplicación, tendrás que usar el siguiente comando de la máquina virtual de Java (JVM):

```
C:> java -classpath C:\java\MyClasses utility.myapp.Cool
```

Cuando se ejecuta la aplicación, la máquina virtual de Java usa los valores de ruta de la(s) clase(s) para buscar cualquier otra clase definida en el paquete `utility.myapp` usada por la clase `Cool`.

Note que el nombre completo del paquete es especificado en el comando. No es posible, por ejemplo, establecer la ruta de la(s) clase(s) para que tenga la ruta Práctica 2. `Classpath`.



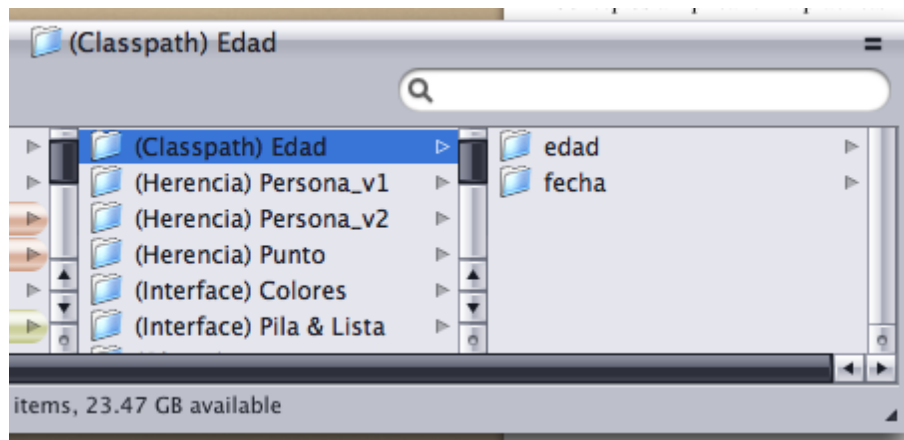
C:\java\MyClasses\utility y usar el comando `Java myapp.Cool`. La clase no sería encontrada.

Uno se puede preguntar que define el nombre del paquete para una clase. La respuesta es que el nombre del paquete es parte de la clase y no puede ser modificada, excepto al recompilar la clase.

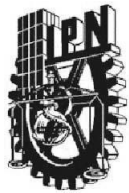
Note que una consecuencia interesante del mecanismo de especificación de paquetes es que los archivos que son parte de un mismo paquete pueden existir en diferentes directorios. El nombre del paquete será el mismo para cada clase, pero la ruta a cada archivo puede comenzar desde un directorio diferente en la ruta de la(s) clase(s).¹

Desarrollo:

Cree una aplicación que calcule la edad de una persona sabiendo su fecha de nacimiento y la fecha actual. La clase principal llamada `edad.java`, la cual se encontrará en un directorio llamado `edad`, pedirá al usuario la fecha de nacimiento y calculará la edad de la persona en cuestión. Por otra parte, en un directorio llamado `fecha`, se encontrará una clase llamada `fecha.java` la cual se encargará de obtener la fecha actual, que será usada en la clase principal. Así se ordenarán los directorios:



¹ Tomado de la documentación de Classpath para Java 1.3
Práctica 2. Classpath.



Este archivo se llamara **edad.jar** y será un jar ejecutable. Por otra parte, se pide un segundo archivo **edad2.jar** ejecutable en el cual se pondrá la clase fecha en un jar interno llamado **date** el cual será llamado por **edad2.jar** para poder ser ejecutada la aplicación.

Recomendaciones:

1. Programe las clases en archivos Java separados, pero en el mismo directorio. Ejecute la aplicación para comprobar el funcionamiento y observar la salida.
2. Conociendo la salida esperada para la aplicación, proceda a separar las clases y realizar las modificaciones pertinentes.

Preguntas:

1. ¿Qué modificaciones requirió realizar en el código fuente para que la aplicación funcionará?
2. ¿Cuáles son las dos formas que existe para modificar el `classpath`?, explique.
3. Almacene la clase secundaria en un archivo JAR (la forma de realizarlo se puede encontrar en la documentación de Java) y describa las formas de compilar la aplicación para obtener los mismos resultados.
4. ¿Cuáles son los beneficios que el formato de archivos JAR proporciona?

Duración:

Una sesión de laboratorio.

Fechas:

Desarrollo de la práctica en laboratorio:

_____.

Entrega del reporte:

_____.