

JAVA desde consola

Una vez tenemos el JDK de JAVA instalado en nuestro sistema junto con las variables de entorno podemos empezar a escribir nuestro primer programa JAVA.

Antes de ponernos a compilar empezaremos definiendo lo que es un paquete en JAVA

Ejercicio 1: Paquetes y directorios

A continuación pondré la declaración de cuatro clases diferentes en paquetes diferentes, cada clase se llamará "**Clase1**", "**Clase2**", "**Clase3**" y "**Clase4**" respectivamente. Voy a jugar un poco con las carpetas donde quedará alojada cada una de las clases para que afiances un poco más el concepto y comprendas bien el funcionamiento de los paquetes en Java y quizá soluciones alguna duda.

Debes descifrar la ubicación exacta de cada clase en el interior del proyecto según la declaración de su respectivo paquete. Indica la estructura de carpetas y la ubicación de cada fichero **.java**

Veamos:

1. Clase número uno:

```
package mis_clases.clases_publicas.clase1;  
public class Clase1  
{  
  
}
```

2. Clase número dos:

```
package mis_clases.clase2;  
class Clase2  
{  
  
}
```

3. Clase número tres:

```
package mis_clases;  
class Clase3  
{  
  
}
```

4. Clase número cuatro:

```
class Clase4  
{  
  
}
```

Para aules:

*Sube a aules el directorio del proyecto para este ejercicio, debe estar comprimido con extension .zip(ejercicio1.zip), dentro del mismo se encontrará toda la estructura de directorios del programa, junto con los ficheros fuente (escribir **Clase1.java**, **Clase2.java**, **Clase3.java** y **Clase4.java**).*

¿Como se compila desde consola?

Antes de realizar la compilación, nos situamos en el directorio base de nuestro proyecto (o el directorio donde hemos decidido que van a ir nuestros fuentes java) y ahí creamos nuestros fuentes **.java** dentro de la estructura de directorios definida en los paquetes(**package**) de cada fichero fuente.

Es importante tener en cuenta lo siguiente:

- Los ficheros **.java** que **NO** tienen definido su **package**, deben colocarse en el directorio base del proyecto JAVA.
En este ejemplo, el directorio base del proyecto es **c:\directorioBase**, por lo tanto el fichero fuente **Fuente1.java** debería ubicarse en la ruta **c:\directorioBase\Fuente1.java**

```
public class Fuente1 {  
  
}
```

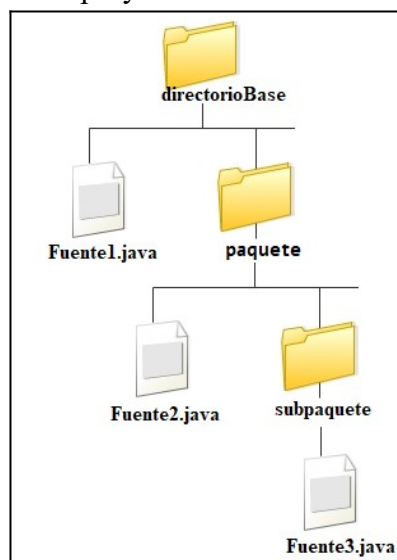
- Si un fichero **.java** **SÍ** tiene definido un **package paquete**; debemos crear en el directorio base del proyecto un subdirectorio **paquete** y meter dentro dicho fichero **.java**. El siguiente fuente debería ir en **c:\ directorioBase\paquete\Fuente2.java**

```
package paquete;  
public class Fuente2 {  
  
}
```

- Si un fichero **.java** tiene definido el paquete **package paquete.subpaquete**; debemos crear en el directorio base del proyecto un subdirectorio **paquete** y dentro de este otro subdirectorio **subpaquete** y dentro de este nuestro fuente **.java**. El siguiente fuente debería ir en **c:\ directorioBase\paquete\subpaquete\Fuente3.java**

```
package paquete.subpaquete;  
public class Fuente3 {  
    public static void main(String args[]){  
    }  
}
```

Es decir, la estructura de los **package** definidos dentro de los ficheros fuente debe coincidir con la estructura de directorios del proyecto JAVA.



Para compilar, una vez situados en el directorio base de nuestro proyecto, debemos compilar así:

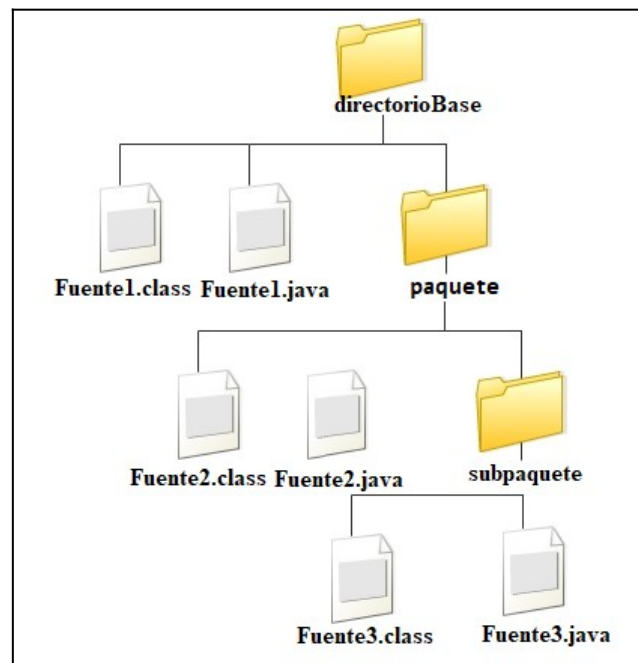
```
javac ruta\Fichero1.java ruta\Fichero2.java ruta\Fichero3.java (... resto de ficheros)
```

En el ejemplo que estamos trabajando tenemos 3 ficheros en nuestro proyecto **JAVA**, para compilar ejecutaríamos desde el directorio base del proyecto la siguiente orden a través de la consola:

```
javac Fuente1.java paquete\Fuente2.java paquete\subpaquete\Fuente3.java
```

es decir, debemos ejecutar el comando **javac** (compilador) desde el directorio base de nuestro proyecto indicando las rutas para llegar desde dicho directorio a los ficheros fuente. Si no hay errores y la compilación se realiza de manera exitosa, se generarán los ficheros **Fuente1.class**, **paquete\Fuente2.class** y **paquete\subpaquete\Fuente3.class**

Después de compilar la estructura de directorios y ficheros **.java** y **.class** nos quedaría así:



¿Como se ejecuta desde consola?

Una vez esté todo compilado y se hayan generado los ficheros **byteCode** con extensión **.class** es hora de ejecutar nuestro programa. Para ello debemos lanzar a la máquina virtual de JAVA la clase que contiene el método **main**. Todos los programas escritos en JAVA inician su ejecución desde la clase que contiene el método **main**.

El fichero **Fuente3.java** contiene el método **main**, por lo tanto, situados en el directorio base de nuestro proyecto ejecutamos:

```
java paquete.subpaquete.Fuente3
```

Fijate que para la ejecución indicamos el paquete de la clase que queremos ejecutar separando los directorios mediante **puntos (.)**, para compilar empleábamos la ruta mediante las **barras (\)**:

¿Compilación y clases en un directorio distinto?

Al compilar como lo hemos hecho hasta ahora, las clases resultantes de la compilación(**ficheros .class**) se almacenan en los mismos directorios en los que se ubican los ficheros fuente(**ficheros .java**).

En los proyectos **JAVA** normalmente se tienen almacenados los fuentes en una ruta diferente de las clases, de modo que si después de realizar la compilación queremos empaquetar las clases del programa en un fichero **.jar** podamos fácilmente prescindir de incluir los archivos fuente.

Para indicar al compilador que deje las clases en una ruta diferente se emplea la opción:

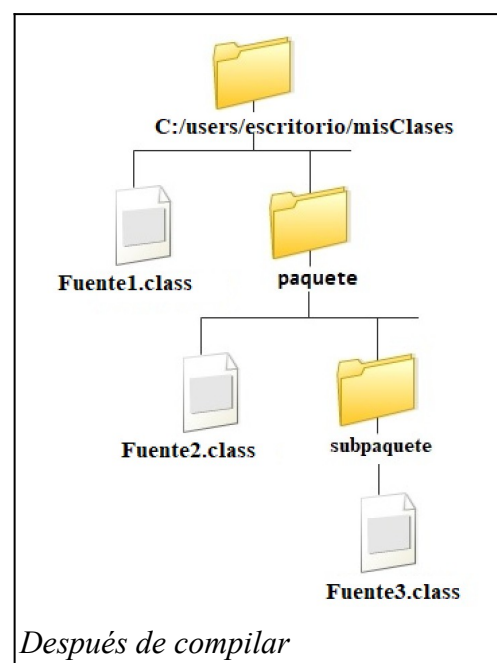
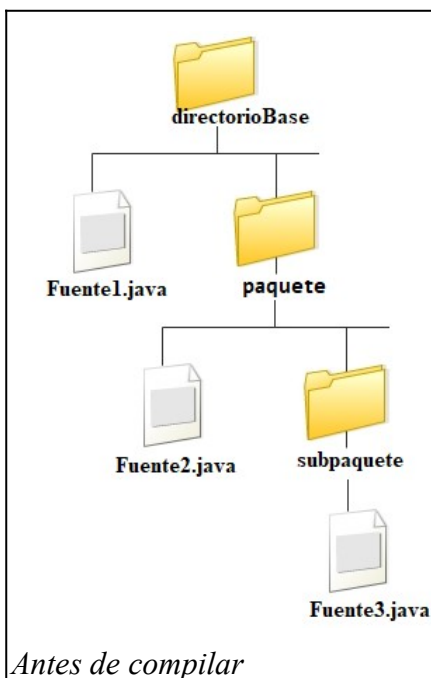
-d ruta_de_las_clases

La ruta de las clases **puede ser una ruta absoluta o una ruta relativa** al directorio desde donde se realiza la compilación. Vamos a imaginar que queremos compilar los ficheros fuente (**.java**) del ejemplo anterior de manera que las clases resultantes se almacenen en la ruta absoluta **C:/users/escritorio/misClases**. Este objetivo lo lograríamos ejecutando desde consola **dentro del directorio base del proyecto** la siguiente orden:

```
javac -d C:/users/escritorio/misClases Fuente1.java paquete\Fuente2.java paquete\subpaquete\Fuente3.java
```

Ruta donde queremos que se generen las clases después de la compilación

La compilación es exactamente igual que la que hemos realizado en el ejemplo anterior pero esta vez indicando mediante la opción **-d** el directorio donde queremos que se almacenen las clases.



Una vez realizada la compilación deberemos ejecutar el programa mediante la orden **java**, en este caso las clases resultantes de la compilación se encuentran en un directorio diferente que los ficheros fuente. Por lo tanto deberemos situarnos desde consola en el directorio **C:/users/escritorio/misClases** y lanzar la orden:

```
java paquete.subpaquete.Fuente3
```

Ejercicio 2: Compilación de clase sin package

En un editor de texto escribimos el siguiente código y lo guardamos como **HolaMundo.java**, con ello obtendremos nuestro primer fichero fuente escrito en lenguaje JAVA. Para que este simple fichero de texto se convierta en código **ByteCode** ejecutable por la máquina virtual de JAVA(**JVM**), este deberá ser compilado. Para ello haremos uso del compilador de JAVA(**javac**), esta herramienta de la JDK es la que se encarga de analizar nuestro código y detectar los posibles errores que hayamos cometido, si no hay errores y el código está bien escrito se generará un fichero con extensión **.class**.

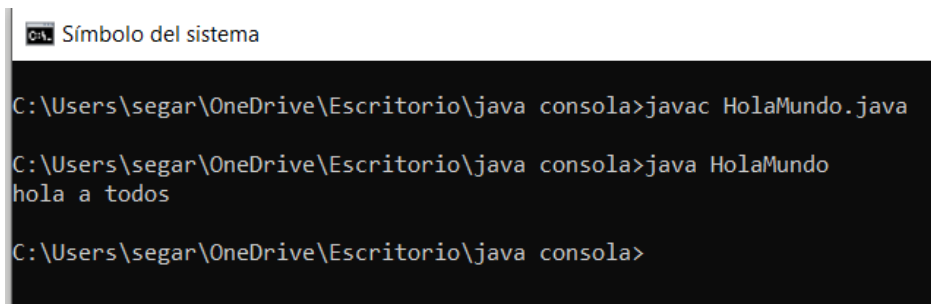
Este fichero **.class** se puede ejecutar en cualquier máquina que tenga instalada una JVM haciendo buena la filosofía de JAVA de **Write once, run anywhere**.

```
public class HolaMundo{  
    public static void main(String[] args){  
  
        System.out.println("hola a todos");  
    }  
}
```

Compilación

Para compilar desde la línea de comandos nos situamos en la carpeta donde hayamos guardado nuestro fuente **HolaMundo.java** y ejecutamos el comando:

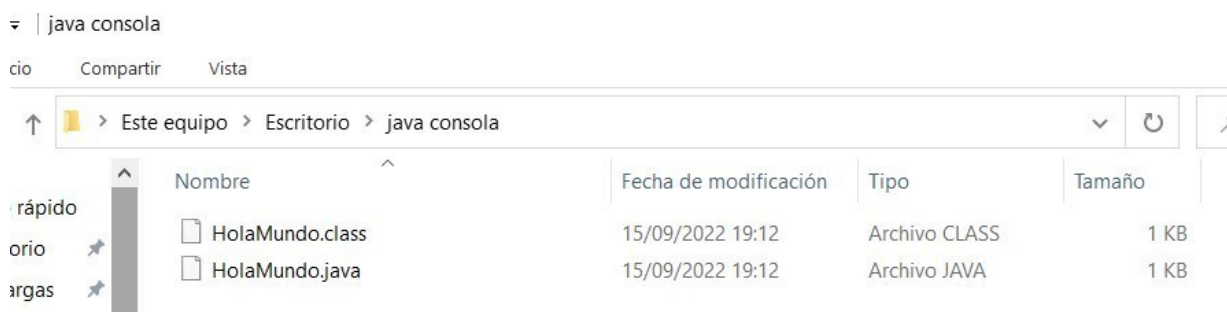
javac HolaMundo.java.



```
C:\Users\segar\OneDrive\Escritorio\java consola>javac HolaMundo.java  
  
C:\Users\segar\OneDrive\Escritorio\java consola>java HolaMundo  
hola a todos  
  
C:\Users\segar\OneDrive\Escritorio\java consola>
```

Ejecución

Si el código del programa está bien escrito se generará en la misma carpeta donde está el fuente el fichero **HolaMundo.class**.



Para ejecutar el programa simplemente escribimos **java HolaMundo**, tened en cuenta que a la hora de ejecutar el programa debemos indicar unicamente el nombre de la clase sin ninguna extensión.

Para aules:

Sube a aules el directorio del proyecto para este ejercicio, debe estar comprimido con extension .zip(ejercicio2.zip), dentro del mismo se encontrará toda la estructura de directorios del programa, junto con los ficheros fuente y sus respectivos .class.

Ejercicio 3: Una clase con package definido

En un editor de texto escribimos el siguiente programa y lo guardamos como

CalculadoraSimple.java. Este programa define su **package** como **org.iesalvarofalomir.calc** por lo tanto deberemos crear la estructura de carpetas dentro del directorio del proyecto y compilar con la orden:

```
javac org\iesalvarofalomir\calc\CalculadoraSimple.java
```

Código **CalculadoraSimple.java**:

```
package org.iesalvarofalomir.calc;

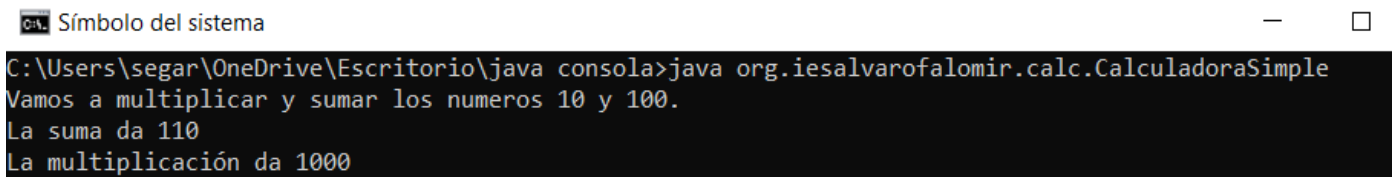
public class CalculadoraSimple{
    public static void main(String[] args){
        System.out.println("Vamos a multiplicar y sumar
los numeros 10 y 100.");

        int a = 10;
        int b = 100;

        System.out.println("La suma da "+ (a+b));
        System.out.println("La multiplicación da "+ (a*b));
    }
}
```

Para la ejecución del programa desde la carpeta del fuente debemos indicar la ruta del paquete, para ellos ejecutamos el siguiente comando:

```
java org.iesalvarofalomir.calc.CalculadoraSimple
```



```
Símbolo del sistema
C:\Users\segar\OneDrive\Escritorio\java consola>java org.iesalvarofalomir.calc.CalculadoraSimple
Vamos a multiplicar y sumar los numeros 10 y 100.
La suma da 110
La multiplicación da 1000
```

Para aules:

Sube a aules el directorio del proyecto para este ejercicio, debe estar comprimido con extension .zip(ejercicio3.zip), dentro del mismo se encontrará toda la estructura de directorios del programa, junto con los ficheros fuente y sus respectivos .class.

Ejercicio 4: Clases en packages diferentes

Ahora vamos a compilar un programa que consta de varias clases definidas en diferentes ficheros y cada uno de ellas esta contenida en su propio **package**. Veremos que desde una clase se llama a las funciones definidas en la otra.

Escribimos el fichero **Calculadora.java**, el cual está definido en el **package org.iesalvarofalomir.calc**. Este programa para realizar correctamente su propósito hace uso de las funciones definidas en la clase **Operacion**. Como la clase **Operacion** se encuentra en un paquete distinto es necesario el uso de la sentencia **import org.iesalvarofalomir.calc.oper.Operacion**. De esta manera le decimos al compilar que las funciones **suma** y **mult** se encuentran definidas en la clase **Operacion** la cual esta en un paquete distinto.

Código Calculadora.java:

```
package org.iesalvarofalomir.calc;

import org.iesalvarofalomir.calc.oper.Operacion;

public class Calculadora{
    public static void main(String[] args){
        System.out.println("Vamos a multiplicar y sumar los numeros
10 y 100.");

        int a = 10;
        int b = 100;

        int suma = Operacion.suma(a, b);
        int multiplicacion = Operacion.mult(a, b);

        System.out.println("La suma da "+suma);
        System.out.println("La multiplicación da "+multiplicacion);

    }
}
```

Después escribimos en un edito de texto el fichero **Operacion.java**.

Código Operacion.java:

```
package org.iesalvarofalomir.calc.oper;

public class Operacion{

    public static int suma(int operA, int operB){
        return operA + operB;
    }
    public static int mult(int operA, int operB){
        return operA * operB;
    }
}
```

En este punto ya deberías de ser capaz de crear la estructura de carpetas del proyecto, realizar la compilación del programa y a continuación ejecutarlo.

En este ejercicio se te pide que en el momento de la compilación le indiques al compilador que quieres que las clases se almacenen en la siguiente ruta:

{carpeta escritorio de tu equipo}\clasesEjercicio4

Para aules:

Sube a aules un fichero de texto ejercicio4.txt con las ordenes empleadas para compilar y ejecutar el ejercicio4. También debes subir el directorio del proyecto para este ejercicio con las fuentes y el directorio donde se han generado las clases después de la compilación. Debe estar todo comprimido con extension .zip (ejercicio4.zip), dentro del mismo se encontrará toda la estructura de directorios del programa, junto con los ficheros fuente y sus respectivos .class.

Ejercicio 5: Clases en packages diferentes

En este ejercicio vamos a descargar desde la tarea de aules el fichero comprimido **ejercicio5fuentes.zip**, después de extraer su contenido veremos que contiene 5 ficheros fuente del lenguaje de programación JAVA, **ClaseA.java**, **ClaseB.java**, **ClaseC.java**, **ClaseD.java** y **ClasePrincipal.java**.

El objetivo de este ejercicio es que crees un directorio base para este nuevo proyecto JAVA y además crees la estructura de directorios en función de los paquetes definidos en cada una de las clases.

Una vez hayas creado toda la estructura de directorios y situado en ellos los ficheros fuente tienes que:

- Realizar la compilación de las 5 clases.
- Ejecutar el programa teniendo en cuenta que el método **main** se encuentra definido en la clase **ClasePrincipal.java**.
- La ejecución correcta del programa mostrará por pantalla el mensaje **“Enhorabuena lograste compilar el programa.”**

Para aules:

Sube a aules un fichero de texto ejercicio5.txt con las ordenes empleadas para compilar y ejecutar el ejercicio5. También debes subir el directorio del proyecto para este ejercicio, debe estar comprimido con extension .zip (ejercicio5.zip), dentro del mismo se encontrará toda la estructura de directorios del programa, junto con los ficheros fuente y sus respectivos .class.

Ejercicio 6: Clases en packages diferentes

En este ejercicio vamos a tomar como punto de partida los ficheros fuente del ejercicio anterior (ejercicio 4), es decir **Calculadora.java** y **Operacion.java** y vamos a ampliar el programa.

A pesar de aun no comprender en profundidad los entresijos del lenguaje de programación JAVA, simplemente estudiando como están definidas las funciones y clases del **ejercicio 4** y aplicando el sentido común te reto a realizar las siguientes mejoras:

- Definir una nueva clase **OperacionBis** la cual estará definida en el paquete **org.iesalvarofalomir.calc.nuevasOperaciones**
- Definir en la clase **OperacionBis** las funciones **div** y **resta**, las cuales aceptaran 2 valores como parámetros y devolverán el resultado de la división y la resta de estos valores respectivamente (fijate como están definidas las funciones **suma** y **mult** del **ejercicio 4**).
- Desde la clase **Calculadora** debemos importar la clase **OperacionBis** para poder usar las funciones **div** y **resta**.

- En la clase **Calculadora** se llamará a las funciones **div** y **resta**(creadas por ti) de la clase **OperacionBis** con el propósito de calcular el resultado de la división y la resta, los resultados se guardaran en 2 variables de nombre **division** y **resta** respectivamente.
- En la clase **Calculadora**, adicionalmente a los mensajes de la suma y la multiplicación, se mostraran los siguientes mensajes:
 - **La división da {resultado división}**
 - **La resta da {resultado resta}**

Para aules:

Sube a aules un fichero de texto ejercicio6.txt con las ordenes empleadas para compilar y ejecutar el ejercicio6. También debes subir el directorio del proyecto para este ejercicio, debe estar comprimido con extension .zip (ejercicio6.zip), dentro del mismo se encontrará toda la estructura de directorios del programa, junto con los ficheros fuente y sus respectivos .class.

¿Uso de librerías externas en ficheros JAR?

Cuando programamos en Java a veces es necesario incluir algunas librerías de terceros, por ejemplo un conector de bases de datos o algo simple como para parser de JSON.

Normalmente estos paquetes se pueden descargar en formato JAR, algo así como un compilado que contiene las clases desarrolladas por otros programadores; y justamente ahí viene una pregunta: ¿Cómo incluir un paquete JAR dentro de un proyecto JAVA mediante consola?

Debemos indicar la ruta donde se ubican las librerías **.jar** en la compilación (**comando javac**) así como en el momento de la ejecución(**comando java**) de la clase que contiene el método **main**. Para indicar la ruta donde se hallan las librerías debemos emplear la opción **-classpath ruta_de_las_clases** tanto en la compilación como en la ejecución.

Con la opción **-classpath** indicamos dónde están las clases externas a las de la API que necesitemos para compilar o ejecutar nuestro programa. Cualquier clase necesaria que no pertenezca a la API(ni a las clases pertenecientes a **java.lang**) debe estar incluida en el **classpath** para poder compilar o ejecutar (aunque se encuentre en el mismo directorio que la que compilamos).

Compilación:

Si las clases están empaquetadas en un fichero **JAR**, se tendrá que hacer referencia a dicho fichero. Por ejemplo si el fichero **misclases.jar** se encuentra en **C:\misclases** la compilación se realizaría tal que:

```
javac -classpath C:\misclases\misclases.jar rutaFicherosFuente
```

Si tenemos varias rutas donde están ubicadas las clases que se necesitan para la compilación, estas se pueden incluir de la siguiente manera:

- **En linux** (se usa **:(dos puntos)** como separador) :
 - **javac -classpath rutaClase1:rutaClase2:rutaClase3 rutaFicherosFuente**
- **En Windows** (se usa **;(punto y coma)** como separador):
 - **javac -classpath rutaClase1;rutaClase2;rutaClase3 rutaFicherosFuente**

Si la clase que contiene el **main** hace uso de clases que se encuentran en un JAR y además utiliza clases que forman parte del mismo proyecto, a la hora de compilar y ejecutar se debe incluir en el **classpath** el directorio base y el directorio donde se encuentra el **jar**. Como en el momento de la compilación nos encontramos en el directorio base, para incluirlo en el **classpath** lo hacemos mediante el **símbolo punto(.)**.

Ejemplo en Windows (en Linux la compilación sería exactamente igual pero utilizando como separador del **classpath** los **dos puntos (:)**):

- **javac -classpath .;C:\misclases\misclases.jar paquete1\UnaClase.java**
 - La sentencia anterior realiza la compilación de la clase **UnaClase.java** contenida en el paquete **paquete1**, dicha compilación se realiza desde el directorio base del proyecto, indicando en el **classpath** que las clases que se importan desde **UnaClase.java** se encuentran:
 - A partir del directorio base (**se indica mediante el punto (.)**)
 - En la ruta donde se encuentra el fichero **misclases.jar**: **C:\misclases\misclases.jar**
- **javac -classpath .;C:\misclases*.jar paquete1\UnaClase.java**
 - La sentencia anterior realiza la compilación de la clase **UnaClase.java** contenida en el paquete **paquete1**, dicha compilación se realiza desde el directorio base del proyecto, indicando en el **classpath** que las clases que se importan desde **UnaClase.java** se encuentran:
 - A partir del directorio base (**se indica mediante el punto (.)**)
 - Dentro del directorio **C:\misclases** en cualquiera de los ficheros **.jar** que allí se encuentran.
 - Se emplea el **asterisco (*)** para indicar que queremos que se busquen las clases en cualquier fichero con extensión **.jar**.

Ejecución:

Vamos a suponer que tenemos un programa que:

- Utiliza clases que se encuentran dentro de librerías externas con extensión **.jar**, estos JAR están dentro del directorio **C:\misclases**
- Utiliza clases que se hallan dentro del directorio base
- La clase que contiene el método **main** se ubica en el paquete **paquete1** y se denomina **UnaClase.class**

Para ejecutar este programa, como siempre, nos colocamos en el directorio base y lanzamos la orden:

java -classpath .;C:\misclases*.jar paquete1.UnaClase

Ejemplo:

Veamos un ejemplo en el que la aplicación hace uso de 3 clases:

- **Saludos**: contiene 2 funciones que devuelven respectivamente 2 textos diferentes.
- **JSONObject**: Clase perteneciente a la librería **json.jar**.
- **ClaseMain**: Contiene 2 sentencias **import** que importan:
 1. La clase **Saludos** perteneciente al proyecto.
 2. La clase **JSONObject** que se encuentra en la librería **json.jar**.

Definición Saludos:

```
package paquete1;

public class Saludos {

    // Devuelve la cadena Buenos días
    public static String saludoMañana() {
        return "Buenos días.";
    }

    // Devuelve la cadena Buenas tardes
    public static String saludoTarde() {
        return "Buenas tardes.";
    }
}
```

Definición ClaseMain:

```
package paquete2;

import org.json.JSONObject;
import paquete1.Saludos;

public class ClaseMain {

    public static void main(String[] args) {

        /* Creación de un objeto JSON y llamada al metodo put
        del objeto recién creado.
        La clase JSONObject esta definida en el fichero json.jar.
        */

        JSONObject objeto = new JSONObject();
        objeto.put("nombre", "Pablo Segarra");

        // Se almacena en la variable saludo el resultado de llamar
        // a la función saludoMañana de la clase Saludos.

        String saludo = Saludos.saludoMañana();

        // Se muestra por pantalla el contenido de las variables

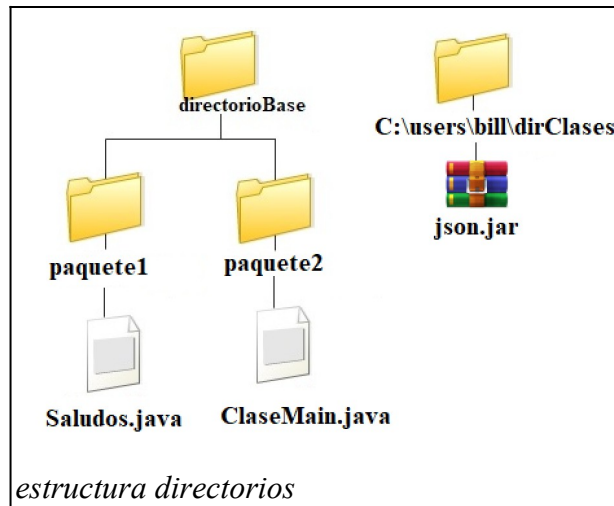
        System.out.println(saludo);
        System.out.println(objeto);
    }
}
```

Si nos fijamos en la clase **ClaseMain** podemos apreciar que contiene 2 sentencias import:

```
import org.json.JSONObject;
import paquete1.Saludos;
```

Por lo tanto para que **ClaseMain.java** se compile con éxito va a necesitar que se importe correctamente tanto la clase **Saludos** como la clase **JSONObject**. La primera(**Saludos**) forma parte del mismo proyecto y por tanto se ubicará dentro del directorio base, sin embargo la clase **JSONObject** está situada en un fichero externo con extensión **.jar**, y es por tanto una librería externa, entonces ¿como compilamos la aplicación? ¿Como la ejecutamos?

- Vamos a suponer que el fichero **json.jar** se encuentra en la ruta **C:\users\bill\dirClases**



Compilacion (desde el directorio base):

```
javac -classpath .;C:\users\bill\dirClases\json.jar paquete1\Saludos.java paquete2\ClaseMain.java
```

ruta donde se hallan las clases

fichero Saludos.java

fichero ClaseMain.java

Ejecución correcta (desde el directorio base):

```
java -classpath .;C:\users\bill\dirClases\json.jar paquete2.ClaseMain
```

Las clases se hallan a partir del directorio base(.) y en el fichero json.org

clase con metodo main

```
Buenos días.
{"nombre":"Pablo Segarra"}
```

Ejecución errónea, sin incluir el punto(directorio base)

```
java -classpath C:\users\bill\dirClases\json.jar paquete2.ClaseMain
```

```
Error: Could not find or load main class paquete2.ClaseMain
Caused by: java.lang.ClassNotFoundException: paquete2.ClaseMain
```

Ejecución errónea, sin incluir la ruta donde se hallan las librerías jar

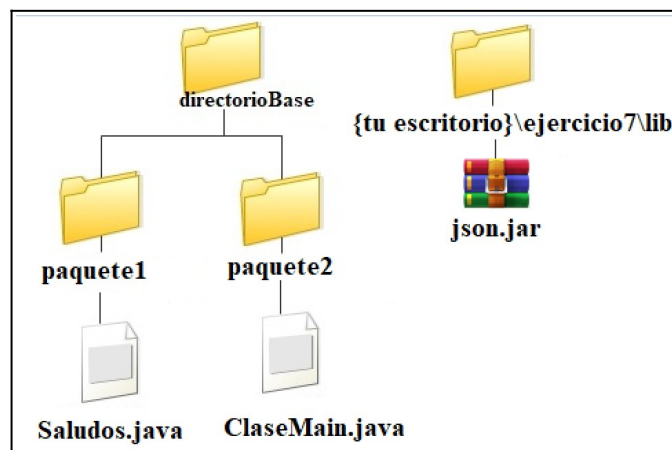
```
java -classpath . paquete2.ClaseMain
```

```
Exception in thread "main" java.lang.NoClassDefFoundError: org/json/JSONObject
    at paquete2.ClaseMain.main(ClaseMain.java:11)
Caused by: java.lang.ClassNotFoundException: org.json.JSONObject
    at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.java:641)
    at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.java:188)
    at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521)
    ... 1 more
```

Ejercicio 7: Realiza el ejemplo anterior

Este ejercicio consiste en implementar el ejemplo que acabamos de ver en el apartado anterior, es decir, debes realizar lo siguiente:

- Escribir los ficheros fuente **Saludos.java** y **ClaseMain.java**, la definición de estos ficheros la tienes descrita en este documento en el apartado anterior, simplemente copia y pega.
- La librería **json.jar** debe estar ubicada en la ruta **{tu escritorio}\ejercicio7\lib**
- Crea la estructura de directorios necesaria para realizar la compilación de forma correcta. Fíjate en la imagen:



- La librería **json.jar** necesaria para compilar y ejecutar el programa se encuentra en aules dentro del fichero **ejercicio7.zip**
- Para realizar la compilación y la ejecución del programa de forma correcta fíjate en las indicaciones que se dan en el ejemplo anterior que tienes explicado en este documento.

Para aules:

Sube a aules un fichero de texto *ejercicio7.txt* con las ordenes empleadas para compilar y ejecutar el ejercicio7. También debes subir el directorio del proyecto para este ejercicio, debe estar comprimido con extensión .zip (*ejercicio7.zip*), dentro del mismo se encontrará toda la estructura de directorios del programa, junto con los ficheros fuente y sus respectivos .class.

Ejercicio 8: Programa con librerías externas

En este ejercicio vamos a descargar desde la tarea de aules el fichero comprimido **ejercicio8.zip**, después de extraer su contenido veremos que contiene 2 ficheros fuente del lenguaje de programación JAVA, **ClaseConLibreriasExternas.java**, **Principal.java** y 2 librerías con extensión .jar, **commons-lang3-3.12.0.jar** y **joda-time-2.0.jar**.

El objetivo de este ejercicio es crear un directorio base para albergar este proyecto JAVA, crear la estructura de directorios según la definición de los **package** dentro de las clases, compilar el código fuente y finalmente ejecutar correctamente el programa resultante de la compilación.

Existen comentarios dentro de los ficheros fuentes que te ayudaran a entender la estructura del programa, si tienes alguna duda pregunta a tu profesor.

Una vez hayas creado toda la estructura de directorios y situado en ellos los ficheros fuente tienes que:

- Ubicar el fichero **commons-lang3-3.12.0.jar** en el directorio: **{tu escritorio}\rutaClases**
- Ubicar el fichero **joda-time-2.0.jar** en el directorio: **{tu escritorio}\otraRutaClases**
- Realizar la compilación de las 2 clases teniendo en cuenta que los **.class** generados deben situarse en un directorio diferente al directorio en el que se encuentran los ficheros fuente (**.java**). La ruta donde deben generarse los **.class** es la siguiente: **{tu escritorio}\ejercicio8\clases**
- Ejecutar el programa JAVA teniendo en cuenta que el método **main** se encuentra definido en la clase **Principal.java**. Ten en cuenta que para ejecutar por consola este programa, debes utilizar como directorio base el directorio donde se han generado las clases durante la compilación. Si la compilación la has realizado correctamente este directorio será **{tu escritorio}\ejercicio8\clases**
- La ejecución correcta del programa mostrará por pantalla el mensaje:

```
Texto repetido = telefono-telefono-telefono-telefono-telefono-  
La fecha de hoy = 2022-10-28
```

* La fecha que aparece en el resultado del programa dependerá del día en que se realice la ejecución.

Para aules:

*Sube a aules un fichero de texto **ejercicio8.txt** con las ordenes empleadas para compilar y ejecutar el ejercicio8. También debes subir el directorio del proyecto para este ejercicio con las fuentes y el directorio donde se han generado las clases después de la compilación. Debe estar todo comprimido con extensión .zip (**ejercicio8.zip**), dentro del mismo se encontrará toda la estructura de directorios del programa, junto con los ficheros fuente y sus respectivos **.class**.*