Universidad César Vallejo

Pregrado

SESIÓN 14:
Despliegue de Aplicaciones

Empaquetar y desplegar

- Ya tenemos nuestra aplicación compilada y funcionando.
- ¿Cómo se la damos a los usuarios?
- ¿Qué le damos exactamente a los usuarios?
- ¿Y si no conocemos a los usuarios?
- Luego necesitamos una forma de empaquetar los aplicativos Java y de desplegarlos a las plataformas cliente.

Empaquetado

- La primera idea es separar el código fuente de los bytecode.
- El código fuente no es útil para el usuario final, y además, ocupa espacio aumentando el tamaño del aplicativo.
- Existen dos técnicas para realizar esta separación:
- Manual cada vez que queramos empaquetar (desaconsejada).
- Una buena estructura de directorios y la opción –d del compilador.

Estructura de directorios

- Existen múltiples posibilidades a la hora de organizar los directorios. Esta es una sugerencia:

1. Un primer directorio raíz. Por ejemplo c:\trabajo
2. Un directorio para el código fuente. Por ejemplo c:\trabajo\src
3. Un directorio para los bytecode. Por ejemplo c:\trabajo\bin

- En el directorio src iremos creando todos los paquetes y código fuente (*.java).

La opción –d de javac

- Si compilamos tal cual el código, los bytecode se crearían en la estructura del código fuente.
- Para evitar esto, existe la opción –d con la que indicaremos el directorio donde queremos que nos genere el bytecode (*.class).

  javac –d ..\bin *.java
- Es muy importante que el directorio de los fuentes (c:\trabajo\src) esté en el CLASSPATH para evitar posibles problemas.

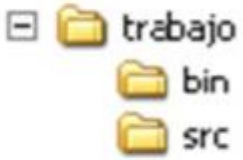
La opción –d de javac

- Al analizar el contenido del directorio especificado en la opción –d veremos que no solo crea allí los bytecode, sino que también genera la estructura de directorios correspondiente a los paquetes Java.
- El procedimiento de ejecución de la aplicación no se ve modificado por esta forma de compilación.
- Simplemente habrá que seguir teniendo en cuenta que:

1. Si hay paquetes, el nombre de la clase incluye el paquete.
2. Añadir al CLASSPATH el directorio de los bytecode para evitar problemas (c:\trabajo\bin).

Ejemplo

- Estructura inicial:



- Desarrollo de los fuentes:

Ejemplo

- Ajustamos el CLASSPATH:

```
set CLASSPATH=c:\trabajo\src;c:\trabajo\bin;%CLASSPATH%
```

- Compilamos:

```
cd c:\trabajo\src
javac –d ..\bin org\javahispano\Test.java
```

Ejemplo

- Ejecutamos:

  java org.javahispano.Test

Nota: Recuerda que ambos directorios: src y bin fuerona ñadidos al CLASSPATH.

Empaquetado

- Ya tenemos los fuentes separados de los bytecode.
- Ahora debemos empaquetar todos los bytecode para que sea más fácil el despliegue.
- Para ello vamos a usar los ficheros JAR:

  JAR (Java Archive).
- Los ficheros JAR están basados en el formato de los ficheros ZIP y nos permiten empaquetar todas las clases en un único fichero.

Ficheros JAR

- Para trabajar con ficheros JAR, el SDK contiene una herramienta: jar.exe
- Si la ejecutamos tal cual, nos muestra todas sus opciones:

Ficheros JAR

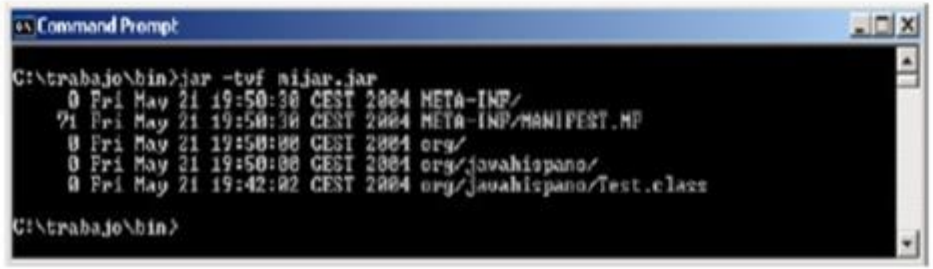
- Las opciones mas comunes son:

1. -c: crear un fichero JAR nuevo.
2. -t: listar el contenido de un fichero JAR.
3. -x: extraer el contenido de un fichero JAR.
4. -f: especificar el fichero JAR (en combinación con las anteriores opciones).
5. -m: especificar un fichero “manifest”.
6. -v: mostrar información del proceso por pantalla.

Ejemplo

- Crear el fichero JAR:

- Listar el contenido del fichero JAR:



Ejecución

- Tanto la JVM como el compilador de Java saben buscar clases dentro de los ficheros JAR.
- Para poder ejecutar aplicaciones en ficheros JAR tenemos tres opciones:

Añadir el fichero JAR al CLASSPATH:

```
set CLASSPATH=c:\temp\mijar.jar
java org.javahispano.Test
```

Añadir el fichero JAR al CLASSPATH en línea de ejecución:

```
java –cp c:\temp\mijar.jar org.javahispano.Test
```

Crear un fichero JAR ejecutable.

Fichero JAR ejecutable

- Los ficheros JAR contienen un fichero de descripción llamado el fichero “manifest”.
- Al listar el contenido del fichero JAR en el ejemplo pudimos observar la entrada:

  META-INF/MANIFEST.MF
- Si extraemos el contenido del fichero JAR y editamos el fichero “manifest” veremos algo como:

  Manifest-Version: 1.0

  Created-By: 1.4.2_02 (Sun Microsystems Inc.)

Fichero JAR ejecutable

- Para que un fichero JAR sea ejecutable necesitamos añadir la siguiente entrada al fichero "manifest":

  Main-Class: nombre_de_clase_principal (Nota: sin .class)

  Main-Class: org.javahispano.Test

- Para añadir dicha línea al fichero "manifest", crearemos un fichero texto con la línea y al crear el fichero JAR utilizaremos la opción –m para referenciarlo:

  jar cvmf manifest.txt mijar.jar *.class

Fichero JAR ejecutable

- Nota: es muy importante añadir un “Intro” al final de la línea “Main-Class” para que se añada correctamente.
- Para ejecutarlo, utilizaremos la opción -jar de la JVM:

  java –jar mijar.jar
- En las últimas versiones de los Sistemas Operativos: Windows y Mac, se pueden ejecutar este tipo de fichero JAR directamente pulsando dos veces sobre el fichero con el ratón.

Despliegue

- Existen distintos tipos de despliegue, dependiendo de dónde resida el código:

1. **Local:** La aplicación reside y se ejecuta en la máquina cliente (usuario final).
2. **Remoto:** La aplicación reside y se ejecuta en una máquina servidora.
3. **Mixto:** La aplicación reside en una máquina servidora pero se ejecuta en la máquina cliente. O parte de la aplicación reside y se ejecuta en la máquina cliente mientras que otra parte reside y se ejecuta en la máquina servidora.

Despliegue local

- La aplicación reside y se ejecuta en la máquina cliente.
- Normalmente se distribuye a través de disco, CD, FTP, e-Mail, o cualquier otro medio en forma de uno o varios ficheros JAR.
- El usuario final necesita tener instalada la JVM (el JRE – Java Runtime Environment) para poder ejecutar código Java.

Despliegue remoto

- La aplicación reside y se ejecuta en la máquina servidora.
- Los clientes son cualquier dispositivo de acceso a la red: navegador web, móvil WAP, etc….
- Para el desarrollo de la aplicación servidora se utilizan tecnologías del J2EE como Java Servlets o Java Server Pages.

Despliegue mixto

- Se trata de una solución intermedia entre el despliegue local y el despliegue remoto.

1. **Applets Java:** las aplicaciones Java residen en un servidor y se ejecutan en un navegador (embebidas en una página HTML) en la máquina cliente.
2. **Programación distribuida:** parte de la aplicación reside en la máquina cliente y parte en la máquina servidora y se comunican mediante tecnologías como: RMI, IIOP, TCP/IP.
3. **Java Web Start (JWS):** las aplicaciones Java residen en un servidor pero se descargan en la máquina cliente a través de un navegador (normalmente) y ya quedan instaladas localmente.

Universidad César Vallejo

Licenciada por Sunedu
para que puedas salir adelante