

Detalles técnicos sobre Java

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

27 de abril de 2007

Variables de entorno

JAVA_HOME Directorio donde está la distribución Java que queremos usar.

CLASSPATH Directorio donde están las clases que queremos usar. Puede haber más de un directorio separado por **:** en sistemas Unix y **;** en sistemas Windows.

¿Dónde se definen las variables de entorno?

- Fichero **/etc/environment**, válidas para todo el sistema
- Ficheros de configuración de la **shell**:
 - sh-Shells no login (terminales en entorno de ventanas):
/etc/bash_bashrc, **~/.bashrc**.
 - En la línea de comandos.
 - **Scripts**.

Esqueleto aplicación Java I

Ficheros a nivel superior `.` . Formato libre, ficheros de configuración, pruebas, etc.

Directorio `./src`

- Código fuente de la aplicación.
- Estructura de directorios debe representar la estructuración en paquetes.
- No son necesarios para la ejecución del programa.

La clase `geometria.figuraPlana.Circulo` debe estar en el fichero

`./src/geometria/figuraPlana/Circulo.java`

Esqueleto aplicación Java II

Directorio `./classes`

- Ficheros *ejecutables* (`.class`).
- Estructura de directorios debe representar la estructuración en paquetes.
- Son necesarios para la ejecución del programa.

La clase `geometria.figuraPlana.Circulo` debe estar en el fichero

```
./classes/geometria/figuraPlana/Circulo.class
```

La clase `PrColeccion` debe estar en el fichero

```
./classes/PrColeccion.class
```

Compilación

Programa `$JAVA_HOME/bin/javac`

```
~/Java$ $JAVA_HOME/bin/javac 1
Usage: javac <options> <source files> 2
where possible options include: 3
-g Generate all debugging info 4
-g:none Generate no debugging info 5
-g:{lines,vars,source} Generate only some debugging info 6
-nowarn Generate no warnings 7
-verbose Output messages about what the compiler is doing 8
-deprecation Output source locations where deprecated APIs are used 9
-classpath <path> Specify where to find user class files 10
-cp <path> Specify where to find user class files 11
-sourcepath <path> Specify where to find input source files 12
-bootclasspath <path> Override location of bootstrap class files 13
-extdirs <dirs> Override location of installed extensions 14
-endorseddirs <dirs> Override location of endorsed standards path 15
-d <directory> Specify where to place generated class files 16
-encoding <encoding> Specify character encoding used by source files 17
-source <release> Provide source compatibility with specified release 18
-target <release> Generate class files for specific VM version 19
-version Version information 20
-help Print a synopsis of standard options 21
-X Print a synopsis of nonstandard options 22
-J<flag> Pass <flag> directly to the runtime system 23
```

Máquina virtual

Programa \$JAVA_HOME/bin/java

```
~/Java$ $JAVA_HOME/bin/java
Usage: java [-options] class [args...]
           (to execute a class)
  or java [-options] -jar jarfile [args...]
           (to execute a jar file)

where options include:
  -d32          use a 32-bit data model if available
  -d64          use a 64-bit data model if available
  -client      to select the "client" VM
  -server      to select the "server" VM
  -hotspot     is a synonym for the "client" VM [deprecated]
               The default VM is client.

  -cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
               A : separated list of directories, JAR archives,
               and ZIP archives to search for class files.
  -D<name>=<value>
               set a system property
  -verbose[:class|gc|jni]
               enable verbose output
  -version      print product version and exit
  -version:<value>
               require the specified version to run
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

Compilación y ejecución *a mano*

El directorio `~/Java` es donde tengo todo lo relacionado con la aplicación:

- Ficheros
- Directorio `src`
- Directorio `classes`

```
~/Java$ $JAVA_HOME/bin/javac -classpath ./classes -sourcepath ./src 1
-d ./classes -deprecation -target 1.5 2
./src/geometria/figurasPlanas/Paralelogramo.java 3
~/Java$ $JAVA_HOME/bin/javac -classpath ./classes -sourcepath ./src 4
-d ./classes -deprecation -target 1.5 ./src/PruebaGeometria.java 5

$JAVA_HOME/bin/java -classpath ./classes PruebaGeometria puntos.txt figuras.txt 1
```

Empaquetado de programas: jar

```
~/Java$ $JAVA_HOME/bin/jar Sintaxis: jar {ctxu}[vfmOMi] [archivo-jar] [archivo-manifest]
Opciones:
  -c crear nuevo contenedor
  -t mostrar contenido de contenedor
  -x extraer archivos nombrados (o todos) del contenedor
  -u actualizar contenedor existente
  -v generar salida detallada en salida estándar
  -f especificar nombre de archivo contenedor
  -m incluir información de manifest del archivo manifest especificado
  -O solo almacenar; no utilizar compresión ZIP
  -M no crear un archivo manifest para las entradas
  -i generar información de índice para los archivos jar especificados
  -C cambiar al directorio especificado e incluir el archivo siguiente
Si alguno de los archivos es un directorio, se procesará de forma recursiva.
Se deben especificar los nombres del archivo manifest y del archivo contenedor
en el mismo orden en que se especifiquen los indicadores 'm' y 'f'.

Ejemplo 1: para archivar dos archivos de clase en un contenedor llamado classes.jar
jar cvf classes.jar Foo.class Bar.class
Ejemplo 2: utilizar un archivo manifest existente, 'mymanifest', y archivar todos
archivos del directorio foo/ en 'classes.jar':
jar cvfm classes.jar mymanifest -C foo/ .
```

Empaquetado de programas: jar

```
~/Java$ cd classes/
~/Java/classes$ jar cvf geometria.jar geometria/ PruebaGeometria.class
adding: META-INF/ (in=0) (out=0) (stored 0%)
adding: META-INF/MANIFEST.MF (in=56) (out=56) (stored 0%)
adding: geometria/ (in=0) (out=0) (stored 0%)
adding: geometria/figurasPlanas/ (in=0) (out=0) (stored 0%)
adding: geometria/figurasPlanas/Recta.class (in=3558) (out=1689) (deflated 52%)
adding: geometria/figurasPlanas/FiguraPlana.class (in=250) (out=163) (deflated 34%)
adding: geometria/figurasPlanas/Circulo.class (in=2630) (out=1277) (deflated 51%)
adding: geometria/figurasPlanas/Superficie.class (in=346) (out=238) (deflated 31%)
adding: geometria/figurasPlanas/Segmento.class (in=2233) (out=1135) (deflated 49%)
adding: geometria/figurasPlanas/Vector.class (in=2675) (out=1330) (deflated 50%)
adding: geometria/figurasPlanas/Paralelogramo.class (in=3990) (out=1863) (deflated 53%)
adding: geometria/figurasPlanas/Punto.class (in=1806) (out=884) (deflated 51%)
adding: geometria/figurasPlanas/FiguraPlanaIncorrecta.class (in=273) (out=213) (deflated 50%)
adding: PruebaGeometria.class (in=4968) (out=2440) (deflated 50%)
Total:
-----
(in = 22777) (out = 13246) (deflated 41%)
~/Java/classes$ mv geometria.jar /tmp
~/Java$ cd ..
~/Java$ cp puntos.txt figuras.txt /tmp
~/Java/classes$ cd /tmp/
/tmp$ $JAVA_HOME/bin/java -classpath geometria.jar
PruebaGeometria puntos.txt figuras.txt
```

Generalización del clásico `make` para Java

- Es independiente de la máquina subyacente.
- Guiado por objetivos.

Fichero `build.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<project default="prueba" basedir=". ">
  <property name="src" value="./src"/>
  <property name="build" value="./classes"/>
  <property name="lib" value="./lib"/>

  <target name="init">
    <mkdir dir="${build}"/>
    <mkdir dir="${lib}"/>
  </target>

  <target name="prueba" depends="init" >
    <javac destdir="${build}" classpath="${classpath}" listfiles="yes"
      optimize="on" srcdir="${src}">
      <classpath>
        <pathelement location="${build}"/>
        <fileset dir="${lib}">
          <include name="*.jar"/>
        </fileset>
      </classpath>
      <include name="pr1/**" />
      <include name="pr2/**" />
    </javac>
  </target>
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

Fichero `build.xml`

```
<target name="clean"
  description="clean up" >
  <delete verbose="true" quiet="false">
    <fileset dir="." includes="**/*.class"/>
    <fileset dir="." includes="**/*.jar"/>
    <fileset dir="." includes="**/semantic.cache"/>
    <fileset dir="." includes="semantic.cache"/>
  </delete>
</target>
</project>
```

1
2
3
4
5
6
7
8
9
10

Para compilar y empaquetar geometria

```
<?xml version="1.0" encoding="utf-8"?>
<project default="geometria" basedir=".>
  .....
  .....
  <target name="geometria" depends="init" description="Compilar las clases de la practica"
    <javac destdir="${build}" classpath="${classpath}" listfiles="yes"
      optimize="on" srcdir="${src}">
      <classpath>
        <pathelement location="${build}"/>
        <fileset dir="${lib}">
          <include name="*.jar"/>
        </fileset>
      </classpath>
      <include name="geometria/**" />
      <include name="PruebaGeometria.java" />
    </javac>
  </target>

  <target name="distGeometria" depends="geometria">
    <jar jarfile="geometria.jar" basedir="${build}" compress="yes"
      includes="geometria/**/*.class, PruebaGeometria.class"/>
  </target>
</project>
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```
~/Java$ ant distGeometria
Buildfile: build.xml

init:

geometria:
  [javac] Compiling 10 source files to /home/luis/facultad/luis/docencia/Java06-07/Java
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/PruebaGeometria.java
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP9anas
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP10anas
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP11anas
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP12anas
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP13anas
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP14anas
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP15anas
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP16anas
  [javac] /home/luis/facultad/luis/docencia/Java06-07/Java/src/geometria/figurasP17anas

distGeometria:
  [jar] Building jar: /home/luis/facultad/luis/docencia/Java06-07/Java/geometria.jar

BUILD SUCCESSFUL
Total time: 6 seconds
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

Scripts

Fichero `geometria.sh`

```
#!/bin/sh
1
2
if [ -z "$JAVA_HOME" ]; then
3
4   JAVA_HOME=/opt/java/jdk1.5
5
6 fi
7
8 JAVA=$JAVA_HOME/bin/java
9
10 $JAVA -cp geometria.jar PruebaGeometria $*
```

con permiso de ejecución

```
~/Java$ chmod +x geometria.sh
~/Java$ ls -al geometria.sh
-rwxrwx--- 1 luis luis 159 2007-04-25 17:56 geometria.sh
```

```
~/Java$ ./geometria.sh puntos.txt figuras.txt
Puntos:
Elemento 1 Punto: (4.0,5.0)
Elemento 2 Punto: (7.0,3.0)
Elemento 3 Punto: (1.0,1.0)
Elemento 4 Punto: (0.5,1.0)
.....
```