

DISTRIBUIRE UN'APPLICAZIONE

Problema

- Dopo aver scritto un'applicazione, bisogna ***distribuirla (deployment)***
- Ma un'applicazione Java è fatta di *molte classi*

Come distribuirle insieme in modo efficiente?

Prima idea: ***zipparle insieme***

OK, però...

**così non si saprebbe da dove iniziare
(dov'è il main?)**

IL FORMATO JAR

JAR (Java ARchive) *è una forma evoluta di ZIP*

- È fisicamente un file compresso ZIP
(si apre con Winzip, PKZip, etc)...
- ... però può contenere *informazioni extra*
(il file *MANIFEST*)
- Grazie ad esse, è possibile *eseguire un'applicazione Java semplicemente facendo doppio clic sul file JAR*, senza decomprimerla.

Lo strumento `jar` fa parte del JDK.

CREARE UN FILE JAR

Per creare un file JAR "base"

```
jar cf nomearchivio.jar classi
```

- *classi* è l'elenco di classi (*.class*) da includere
- la forma **.class* permette di aggiungere in un colpo solo tutte le classi di una data cartella
- l'elenco può comprendere cartelle diverse, separate da punti e virgola (esempio: **.class;pippo/*.class*)
- Questo JAR però ***non può essere ancora eseguito facendoci semplicemente "doppio clic" sopra***, perché manca l'indicazione dalla classe che contiene il main.

CREARE UN FILE JAR

Per creare un file JAR completo:

```
jar cmf info.txt nomearchivio.jar classi
```

- info.txt è un file di testo che contiene la riga:
Main-Class: *nomeclasseMain*
- Questa riga viene aggiunta al file MANIFEST e consente all'interprete di sapere quale classe contiene il main
(NB: è indispensabile una riga vuota dopo questa)
- IN ALTERNATIVA si può produrre il file jar come nel caso precedente, e *modificare manualmente* il file MANIFEST contenuto nel jar, aggiungendo la riga richiesta (*metodo non consigliato e poco flessibile*)

ESEGUIRE UN FILE JAR

Per eseguire un'applicazione in file JAR

```
java -jar nomefile.jar
```

- Questo modo di procedere si applica a Java2 (jdk 1.2 o successivi): richiede che il file jar contenga nel file *manifest* il nome della classe che contiene il main
- Nelle versioni precedenti di Java (jdk1.1) si adottava una diversa sintassi, che richiedeva di *specificare esplicitamente* il nome della classe contenente il main:

```
java -cp nomefile.jar MainClass
```

ESEMPIO

Applicazione **App1** costituita da

- `Counter.class`
- `Esempio.class` *(classe con il main)*

CREAZIONE FILE JAR "base"

```
jar cf App1.jar Counter.class Esempio.class
```

oppure, se si trovano in una cartella:

```
jar cf App1.jar *.class
```

Contiene la riga:

Main-Class: Esempio

CREAZIONE FILE JAR completo

```
jar cmf infoApp1 App1.jar Counter.class Esempio.class
```

oppure, se si trovano tutti e soli nella stessa cartella:

```
jar cmf infoApp1 App1.jar *.class
```

ESEMPIO

Il risultato

Contiene la riga:
Main-Class: Esempio

Applicazione eseguibile
(basta fare doppio clic!)



Attenzione:

- Facendo doppio clic sul jar viene lanciata **una versione dell'interprete Java priva di console** → **non si vede niente!**
- È un metodo pensato per lanciare applicazioni grafiche, che creano personalmente le loro finestre.

ESEMPIO

E allora?

- Per vedere qualcosa di più, conviene *lanciare l'applicazione dalla linea di comando*:

java -jar App1.jar

```
C:\temp> java -jar App1.jar
2
11
C:\temp>
```

Alternativa:

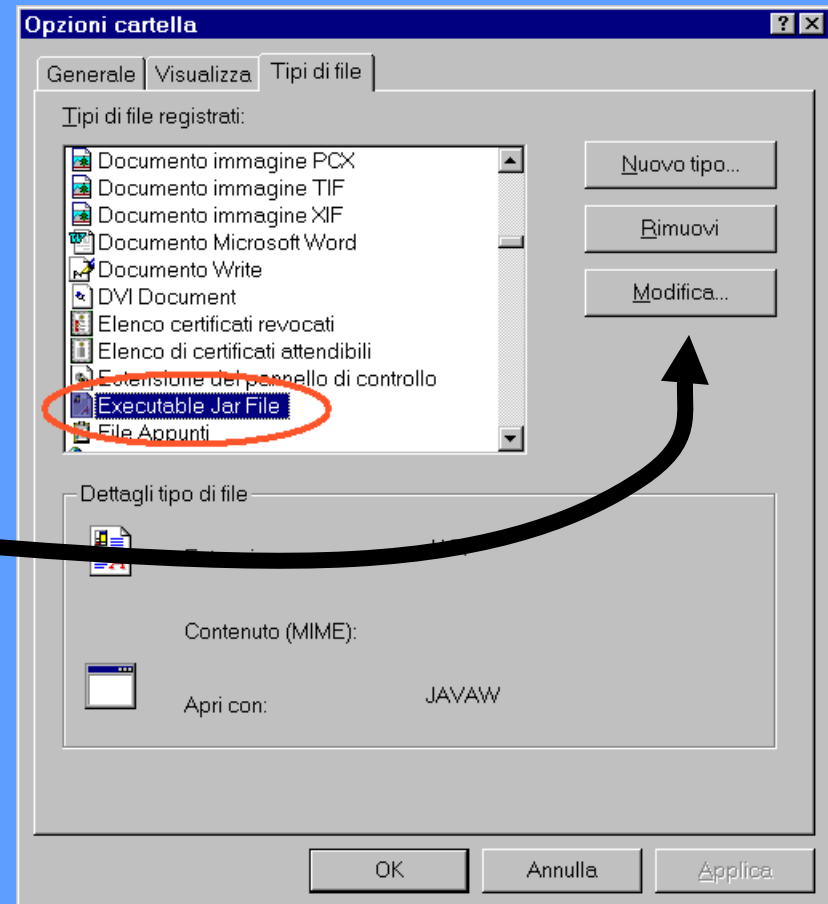
- **Cambiare le impostazioni di Windows** in modo che lanci sempre e comunque l'interprete standard (java.exe) anziché la versione priva di console (javaw.exe)

**** Approccio consigliato solo ad utenti esperti ****

CAMBIARE LE IMPOSTAZIONI DI WINDOWS

Passo 1

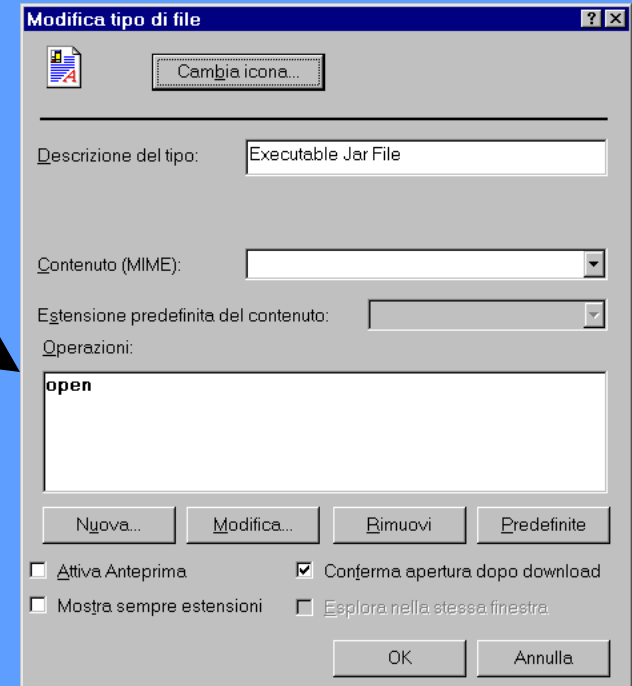
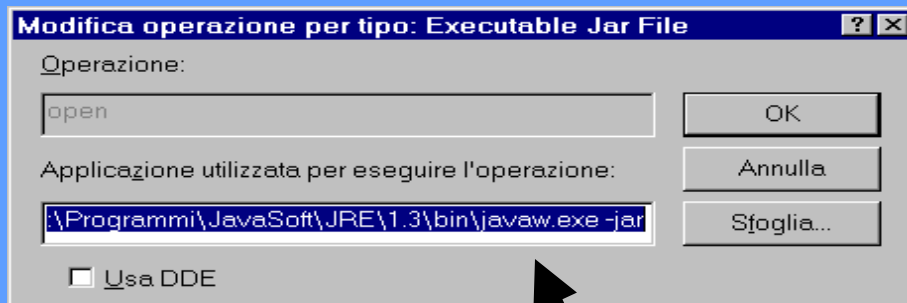
- Da una qualsiasi finestra di Explorer, scegliere Visualizza, Opzioni Cartella, Tipi di file
- Selezionare la riga denominata *"Executable Jar File"* (vedi figura a lato)
- Fare clic sul bottone "Modifica..." (segue)



CAMBIARE LE IMPOSTAZIONI DI WINDOWS

Passo 2

- Si apre la finestra a lato.
- Fare doppio clic sulla voce "open": appare la finestra sottostante



Passo 3

- Cambiare la scritta JAVAW.EXE in JAVA.EXE, e confermare (OK)
- Chiudere, premendo OK, anche tutte le altre finestre, nell'ordine.
- **Per ripristinare le impostazioni iniziali, ripetere la procedura rimettendo JAVAW.EXE al posto di JAVA.EXE**