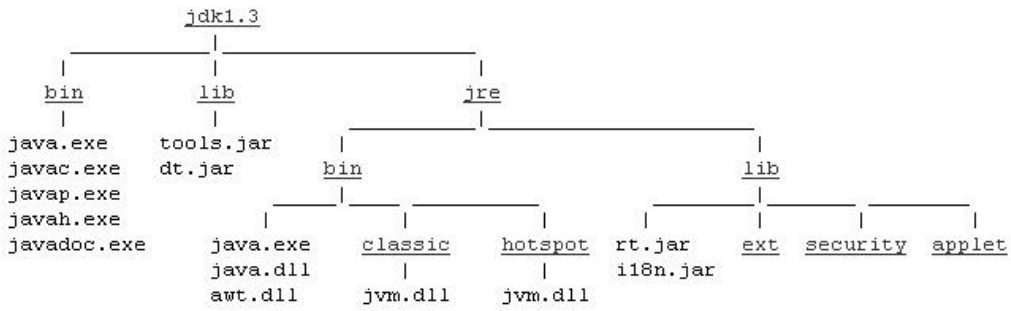
Esercitazione n° 1

Obiettivi

- Introduzione all'utilizzo di ***Java Development Kit*** (**JDK**) versione 1.3
- Sviluppare programmi Java tramite linea di comando
  Es: *javac, java, jdb, javadoc…*
- Primo esempio di programma Java con due classi:
  - variante in un solo file
  - variante in due file distinti

JDK1.3: il direttorio

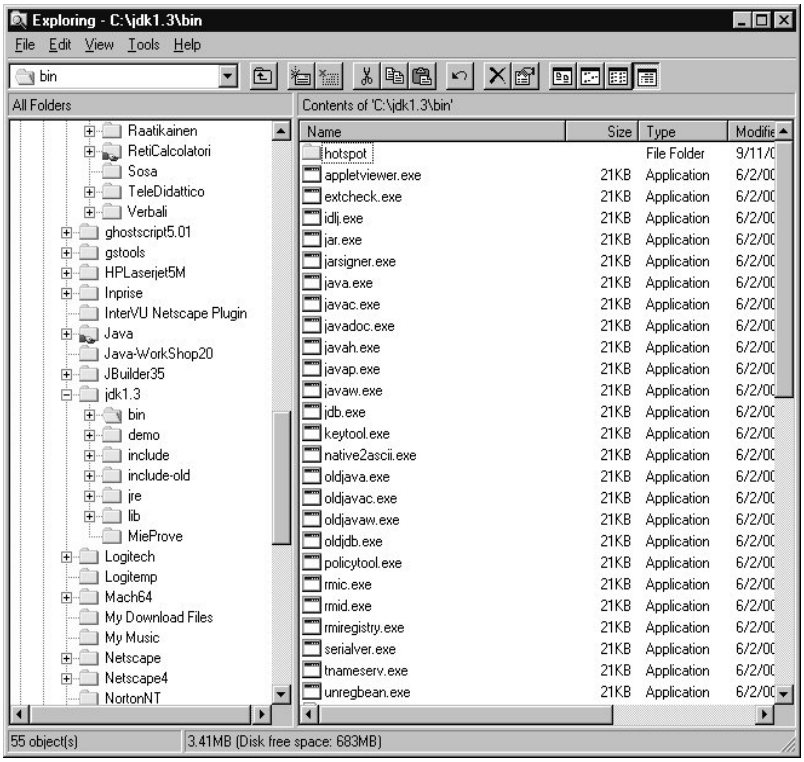


.\jdk1.3\lib Contiene i file usati dai tool di sviluppo

.\jdk1.3\jre Java Runtime Environment. Librerie ed eseguibili per la Java Virtual Machine

.\jdk1.3\bin Contiene i file eseguibili del tool di sviluppo

JDK1.3: il direttorio /bin



Strumenti a linea di comando

- ***Compilatore*** - javac
- ***Macchina virtuale*** - java
- ***Debugger*** - jdb
- Generatore automatico di ***documentazione ipertestuale*** - javadoc
- Altri ***comandi*** - appletviewer, javap, rmic, …

Documentazione completa:
http://www.java.sun.com/j2se/1.3/docs/tooldocs/tools.html

javac

- Legge *sorgenti Java* di classi e interfacce e li ***compila*** in *classfile* in formato *bytecode*
- Permette di compilare ***classi singole*** e ***gruppi di classi***, anche mantenendo direttori separati per i file sorgenti e compilati

javac [opzioni] [sorgenti] [@ElencoSorg]

possibili opzioni:

- **classpath**: classi bootstrap, extension, poi *classpath* di utente (*variabile di ambiente* e *opzione*)

-**d** (direttorio per classi), -**g** (debugging abilitato), -**verbose** (info estese sulla compilazione), **...**

java

- Mette in ***esecuzione*** una applicazione Java
- Avvia una macchina virtuale Java, ***carica*** una classe specificata e invoca il suo metodo ***main*** (*pubblico* e *statico*)

java [opzioni] File.class [parametri]

java [opzioni] -jar File.jar [parametri]

varianti: ***javaw***, ***oldjava***, ***oldjavaw***

opzioni: -**classpath**/-**cp**, -**jar**, -**verbose**, -**version**, -**?**, ...

Esempio: L'EURO-CONVERTITORE (1)

☞ Scopo

Definire una astrazione per un semplice Euro-Convertitore (EuroConverter.java)

Realizzare un'applicazione che stampa una tabella di conversione (Main.java)

EuroConverter.java

Specifica

Componente caratterizzato in ogni istante da un accumulatore che contiene la quantità corrente di Euro da convertire

Componente a cui si accede tramite le operazioni di:

- **add**: Somma Euro alla quantità in accumulatore
- **sub:** Toglie Euro alla quantità in accumulatore
- **convert:** restituisce il corrispondente valore in Lire della quantità di Euro in accumulatore
- **currentValue:** restituisce l'attuale valore dell'accumulatore

Main.java

Specifica

Si vuole produrre in uscita la stampa di una tabella di conversione Euro -> Lire per i valori 1,5,10,15,20,….100.

ES:

```
Euro: 1 Lire: 1936
Euro: 5 Lire: 9681
Euro: 10  …..
  …….
```


Esercitazione n°1: variante su due file(1)

Scopo

Definire le classi su più file

Esercitazione n°1: variante su due file(2)

Come procedere?

1. scaricare le classi **EuroConverter.java** e **Main.java**

2. Salvare il file **EuroConverter.java** sotto la directory C:\TEMP

   Salvare il file **Main.java** sotto la directory C:\TEMP

Ricorda : Il nome del file deve corrispondere al nome della classe!!!!

Esercitazione n°1: variante su due file(3)

3. ***Compilare…***

```
C:\TEMP> javac EuroConverter.java
```

? *quali e quanti file .class produce? In quale directory?*

4. ***Eseguire…***

```
C:\TEMP> java EuroConverter
```

ha senso? ?

Esercitazione n°1: variante su due file(3)

3 ***Compilare…***

C:\TEMP> `javac Main.java`

? *quali e quanti file .class produce? In quale directory?*

4 ***Eseguire…***

C:\TEMP> `java Main`

ha senso? ?

? *Qual è il vantaggio di avere le classi in file separati?*

Maggiore RIUTILIZZABILITA', tutti i cambiamenti sulla classe **EuroConverter** sono confinati alla classe **EuroConverter** dentro il file **EuroConverter.java**

Esercitazione n°1: variante in un file (1)

Come procedere?

1 Scaricare il file **Esempio1.java** dal sito Web del corso e salvarlo in **C:\TEMP**

2 Il file **Esempio1.java** contiene due classi: una classe **public** di nome completo ***Esempio1*** e una classe innestata di nome ***EuroConverter***.

Esercitazione n°1: variante in un file (2)

4 La classe ***EuroConverter*** definisce il tipo di dato astratto 'Euro-Convertitore'.

4.1 **Parte Nascosta (stato):** definisce in una variabile protetta l'accumulatore

4.2 **Parte Visibile (operazioni):**

Costruttore: EuroConverter(double euro)

Trasformatori: void add(double euro)
void sub (double euro)

***Selettori*:** double currentValue()
double convert()

Per l'arrotondamento si possono usare i metodi di utilità predefiniti nella classe java.lang.Math di Java

Esercitazione n°1: variante in un file (3)

5 La classe **Esempio1** contiene il metodo ***main()*** nel quale viene utilizzato il tipo di dato astratto **EuroConverter**

5.1 In particolare, crea due oggetti di tipo ***EuroConverter*** referenziati dalle variabili uno e due ed inizializzati al valore iniziale comune di € 100,00.

5.2 Quindi somma 10 Euro al convertitore uno e ne toglie 10 dal convertitore due. [istruzioni "`uno.add(10.00);`" e "`due.dec(10.00);`"].

5.3 Infine stampa a video i valori di conversione in Lire degli accumulatori dei due convertitori usando "`System.out.println`" per la stampa. Il valore nel convertitore `uno` sarà £ 212.990, quello del convertitore `due` £ 174.264.

Esercitazione n°1: variante in un file (4)

Una possibile soluzione. Contenuto del file Esempio1.java:

```
public class Esempio1 {
  public static void main(String[] args) {
    EuroConverter uno = new EuroConverter (100.00);
    EuroConverter due = new EuroConverter (100.00);
    uno.add(10.00);
    due.sub(10.00);
    System.out.println("Uno: Importo Lire " + uno.convert());
    System.out.println("Due: Importo Lire " + due.convert());}
}
public class EuroConverter{
  private double i=0;      //ACCUMULATORE
  public EuroConverter(double j) {i=j;}
  public void add(double j) {i=i+j;} //Somma i di j Euro
  public void sub(double j){i=i-j;}  // Sottrazione...
  public double currentValue(){return i;}
  public double convert() {return Math.rint(i*1936.27);}
 }
```


Esercitazione n°1: variante in un file (5)

Compilazione

C:\TEMP> `javac Esempio1.java`

? *quali e quanti file .class produce? In quale directory?*

C:\TEMP> `javac -g Esempio1.java`

? *cosa cambia?*

L'opzione -g avvia una compilazione con informazione di debugging abilitate

Esecuzione

C:\TEMP> `java Esempio1`

Documentazione

C:\TEMP> `javadoc Esempio1.java`