06/02/2008 06/02/2008

# La classe Counter

```
public class Counter {
  private int val;

public Counter() { val = 1; }
  public Counter(int v) { val = v; }
  public void reset() { val = 0; }
  public void inc() { val++; }
  public void inc(int k) { val += k; }
  public int getValue() { return val;}

public String toString(){
    return "Counter di valore " + val;
  }
}
```

#### IL CONCETTO DI PACKAGE

- Una applicazione è spesso composta di molte classi (eventualmente correlate)
- Un package è un gruppo di classi che costituiscono una unità concettuale.
  - un package può comprendere parecchie classi
  - anche definite in file separati
- Una dichiarazione di package ha la forma:
   package <nomepackage>;
   Se presente, deve essere all'inizio di un file.

```
package pippo; File Counter.java

public class Counter {
    ...
}

package pippo; File Esempio4.java

public class Esempio4 {
    public static void main(String args[]) {
        ...
}
```

#### **PACKAGE E FILE SYSTEM**

- Esiste una corrispondenza biunivoca fra
  - nome del package
  - posizione nel file system delle classi del package
- Un package di nome pippo richiede che tutte le sue classi si trovino in una cartella (directory) di nome pippo

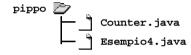
```
pippo 👉 Counter.java
```

#### **COMPILAZIONE ...**

Per compilare una classe Counter che fa parte di un package pippo occorre:

- porsi nella cartella superiore a pippo
- e lì invocare il compilatore con il percorso completo della classe:

javac pippo/Counter.java



### **COMPILAZIONE ...**

Per compilare una classe Counter che fa parte di un package pippo occorre:

• por Non c'è scelta: ogni altro modo di invocare il compilatore è errato!

compieto della classe:

javac pippo/Counter.java



#### ... ED ESECUZIONE

Per eseguire una classe Esempio4 che fa parte di un package pippo occorre:

- porsi nella cartella superiore a pippo
- e lì invocare l'interprete con il <u>nome</u> assoluto della classe:

pippo Counter.class

Esempio4.class

#### **PACKAGE DI DEFAULT**

- Se una classe non dichiara di appartenere ad alcun package, è automaticamente assegnata al package di default
- Per convenzione, questo package fa riferimento alla cartella (directory) corrente
  - è l'approccio usato in tutti i precedenti esempi
  - si possono compilare ed eseguire i file nella cartella in cui si trovano, senza premettere percorsi o nomi assoluti

06/02/2008 06/02/2008

#### SISTEMA DEI NOMI DEI PACKAGE

- Il sistema dei nomi dei package è strutturato
- Perciò, sono possibili *nomi di package* strutturati, come:

```
java.awt.print
pippo.pluto.papero
```

 Conseguentemente, le classi di tali package hanno un nome assoluto strutturato:

```
java.awt.print.Book
pippo.pluto.papero.Counter
```

#### SISTEMA DEI NOMI: DIFETTO

 Ogni volta che si usa una classe, Java richiede che venga denotata con il suo nome assoluto:

```
java.awt.print.Book b;
b = new java.awt.print.Book();
```

- Questo è chiaramente <u>scomodo</u> se il nome è lungo e la classe è usata frequentemente.
- Per tale motivo si introduce il concetto di importazione di nome.

#### IMPORTAZIONE DI NOMI

 Per evitare di dover riscrivere più volte il nome assoluto di una classe, si può importarlo:

```
import java.awt.print.Book;
```

- Da questo momento, è possibile scrivere semplicemente Book invece del nome completo java.awt.print.Book
- Per importare in un colpo solo tutti i nomi pubblici di un package, si scrive

```
import java.awt.print.*;
```

#### **IMPORTAZIONE DI NOMI**

#### Attenzione:

l'istruzione import non è una #include!

- in C, il pre-processore gestisce la #include copiando il contenuto del file specificato nella posizione della #include stessa
- <u>in Java non esiste alcun pre-processore</u>, e non si include assolutamente nulla
- <u>si stabilisce solo una "scorciatoia</u>" per scrivere un nome corto al posto di uno lungo.

06/02/2008

## **PACKAGE E VISIBILITÀ**

- Oltre a pubblico / privato, in Java esiste un terzo tipo di visibilità: la visibilità package
- È il default per classi e metodi
- Significa che dati e metodi sono accessibili solo <u>per le altre classi dello stesso package</u> in qualunque file siano definite
- Altre classi, definite in altri package, non possono accedere a dati e metodi di questo package qualificati a "visibilità package", esattamente come se fossero privati.

## **PACKAGE E VISIBILITÀ**

A differenza del C, <u>il file rimane solo un</u> <u>contenitore fisico</u>, non definisce più un

• ambiente (scope) di visibilità!

 Significa che dati e metodi sono accessibili solo per le altre classi dello stesso package in qualunque file siano definite

Non è quindi possibile, <u>né sensato</u>, pensare di definire una classe *visibile in* un solo file: la visibilità si esprime solo con riferimento ai package. on uesto e",

te un

kage

## IL PACKAGE java.lang

- Il nucleo centrale dal linguaggio Java è definito nel package java.lang
- È sempre importato automaticamente: import java.lang.\* è sottintesa
- Definisce i tipi primitivi e una bella fetta della classi di sistema
- Molte altre classi standard sono definite altrove: ci sono più di cinquanta package!!
  - java.awt, java.util, java.io, java.text,...
  - javax.swing, ...