

## Programmazione ad Oggetti

L'ambiente Java SDK



V 1.4 © Marco Torchiano 2006

## Classi

- La sintassi/semantica Java è simile al C
- In Java il concetto di primo livello sono le classi

```
public class prova {  
}
```
- Ogni classe sta in un file *.java* con lo stesso nome
  - ♦ Quindi un file per ogni classe

2

## Metodi

---

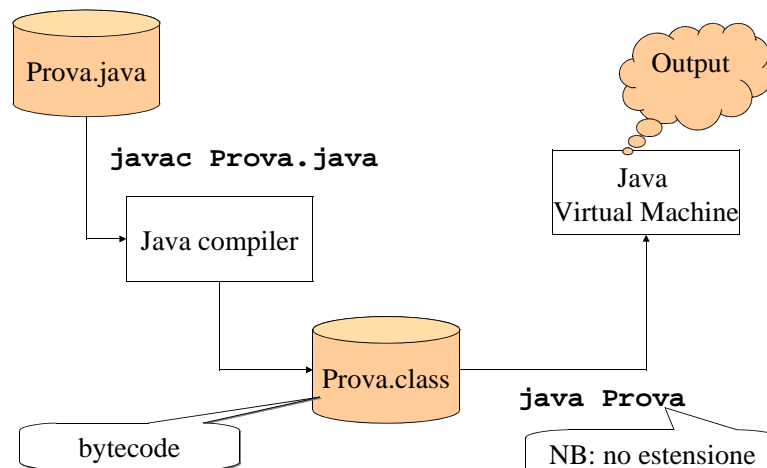
- In Java non esistono le funzioni, solo **metodi** dentro le classi
  - L'esecuzione inizia da un metodo speciale:

```
public static void main(String[] args)
```
  - Si noti che
    - ♦ Il tipo di ritorno è `void`
    - ♦ `args[0]` è il primo argomento
      - Non il nome del programma come in C
- 

3

## Compilazione

---



4

## Esempio

---

- File Prova.java:

```
public class Prova {  
    public static void main(String[] args){  
        int a;  
        a = 3;  
        System.out.println(a);  
    }  
}
```

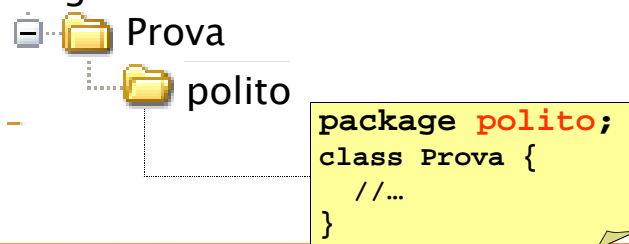
---

5

## Package

---

- I package sono rappresentati da folder.
- Il nome (path) del folder deve corrispondere alla dichiarazione package.

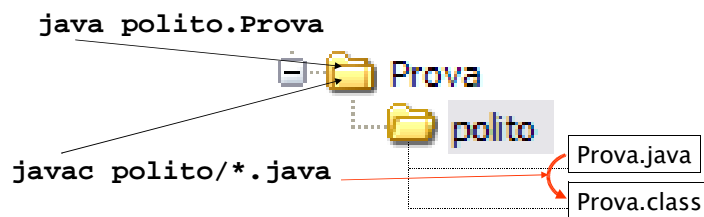


6

## Package

---

- Compilazione ed esecuzione devono essere fatte dal folder radice
  - ♦ La classe `polito.Prova` deve viera mappata nel pathname `polito/Prova.class`



7

## Stringhe

---

- In Java c'è una classe apposta per rappresentare le stringhe: **String**
- Permette di concatenare le stringhe con l'operatore `+`
- Es.

```
String s1 = "A";  
String s2 = "B";  
String s3 = s1 + s2 + "C";
```

8

## Output

---

- L'output viene fatto tramite l'oggetto
  - ♦ `System.out`
- I metodi usati sono
  - ♦ `println()`: stampa l'argomento + a capo
  - ♦ `print()`: stampa l'argomento
- Entrambi i metodi accettano stringhe e tipi di base

---

9

## Domande tipiche

---

- E' piu' "potente" Java o il C?
  - ♦ Performance: C meglio ma non quanto si potrebbe credere (JIT)
  - ♦ Facilità d'uso: Java
  - ♦ Limitazione degli errori: Java
- Come faccio a generare un file ".exe"?
  - ♦ Non lo faccio, per eseguire un programma si usa la JVM installata
  - ♦ GCJ: <http://gcc.gnu.org/java/>

---

10

## Domande tipiche

---

- Ho scaricato Java ma non riesco a compilare i programmi Java
  - ♦ Verificare di aver scaricato Java SDK (che include il compilatore) e non Java RTE (solo VM)
  - ♦ Verificare che il percorso `pathToJava/bin` deve essere nel path

---

11

## Domande tipiche

---

- Java non trova la classe (`ClassNotFoundException`)
  - ♦ Il nome della classe non deve includere l'estensione `.class`:
    - Es. `java Prova`
  - ♦ Verificare che la directory di lavoro sia quella corretta
    - Se la classe è in un package posizionarsi nella radice (relativa ai package)

---

12

## File jar

---

- I programmi Java sono distribuiti in pacchetti inclusi in file **jar**.
- I file jar sono essenzialmente degli archivi compressi
- E' possibile eseguire direttamente un programma contenuto in un file jar

---

13

## Creazione di un file jar

---

- Tramite il comando jar:  
`jar cvf mio.jar *`
- Per eseguire classi contenute in un jar:  
`java -cp mio.jar Prova`
- Per vedere i contenuti:  
`jar tf mio.jar`

---

14

## Jar Main class

---

- Se è definita la classe main il jar può essere eseguito più semplicemente:

```
java -jar mio.jar
```

- Il jar deve essere creato aggiungendo informazioni sul main:

```
jar cvfm mio.jar manifest.txt *.class
```



Main-Class: Prova

---

15