

Ingredienti sintattici di Java

Eugenio G. Omodeo



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Dip. Matematica e Geoscienze — DMI

Trieste, 23/09/2015

specificare

[spe-ci-fi-cà-re]

(*specifico, -chi, specificano; specificante; specificato*)

v. tr.

1 Dichiarare, indicare, spiegare qualcosa in tutti i suoi particolari: *richiedendo i libri in lettura devi s. l'edizione; ass. non capisco, cerca di specificare*



STAMPA

(Dizionario online tratto da: Grande Dizionario Italiano di Gabrielli Aldo, Ed. Hoepli)

PROGRAMMARE:

In Java si programma *specificando classi*

In effetti:

PROGRAMMARE:

In Java si programma *specificando classi*

'**class**' è il costrutto Java di piú alto livello, che racchiude dettagli di vario genere.

In effetti:

Suoi principali sotto-costrutti:

PROGRAMMARE:

In Java si programma *specificando classi*

In effetti:

'**class**' è il costrutto Java di piú alto livello, che racchiude dettagli di vario genere.

Suoi principali sotto-costrutti:

- 1 dichiarazioni

PROGRAMMARE:

In Java si programma *specificando classi*

In effetti:

'class' è il costrutto Java di piú alto livello, che racchiude dettagli di vario genere.

Suoi principali sotto-costrutti:

- 1 dichiarazioni
- 2 istruzioni

PROGRAMMARE:

In Java si programma *specificando classi*

In effetti:

'class' è il costrutto Java di piú alto livello, che racchiude dettagli di vario genere. Suoi principali sotto-costrutti:

- 1 dichiarazioni
- 2 istruzioni
- 3 o talvolta una miscela delle due, ad es.

```
final double pi = 3.14 ;
```

- 1 Dichiarazioni `di tipo` per le variabili, ad es.:
`double piGreca ;`

- 1 Dichiarazioni `di tipo` per le variabili, ad es.:

`double piGreca ;`

- 2 `Firme` dei metodi, ad es.:

`public static void main(String[] aa)`

- 1 Dichiarazioni `di tipo` per le variabili, ad es.:

`double piGreca;`

- 2 `Firme` dei metodi, ad es.:

`public static void main(String[] aa)`

formano un sotto-linguaggio di Java che dà informazioni essenziali al *compilatore*

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

- ① modifiche del contenuto delle variabili

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

- 1 modifiche del contenuto delle variabili
- 2 'flusso di controllo' (v. qui sotto)

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

- 1 modifiche del contenuto delle variabili
- 2 'flusso di controllo' (v. qui sotto), in particolare:
- 3 avvio di metodi

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

- 1 modifiche del contenuto delle variabili
- 2 'flusso di controllo' (v. qui sotto), in particolare:
- 3 avvio di metodi
- 4 restituz. di *controllo* e *risultato* da parte di un metodo che si sta concludendo

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

- 1 modifiche del contenuto delle variabili
- 2 'flusso di controllo' (v. qui sotto), in particolare:
- 3 avvio di metodi
- 4 restituz. di *controllo* e *risultato* da parte di un metodo che si sta concludendo

'PER NOI' LE ISTRUZIONI SI SUSSEGUONO:

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

- 1 modifiche del contenuto delle variabili
- 2 'flusso di controllo' (v. qui sotto), in particolare:
- 3 avvio di metodi
- 4 restituz. di *controllo* e *risultato* da parte di un metodo che si sta concludendo

'PER NOI' LE ISTRUZIONI SI SUSSEGUONO:

- una alla volta,

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

- 1 modifiche del contenuto delle variabili
- 2 'flusso di controllo' (v. qui sotto), in particolare:
- 3 avvio di metodi
- 4 restituz. di *controllo* e *risultato* da parte di un metodo che si sta concludendo

'PER NOI' LE ISTRUZIONI SI SUSSEGUONO:

- una alla volta,
- a partire dalla prima istr. del metodo **main**,

LE ISTRUZIONI

dirigono l'*esecuzione* del programma disponendo

- 1 modifiche del contenuto delle variabili
- 2 'flusso di controllo' (v. qui sotto), in particolare:
- 3 avvio di metodi
- 4 restituz. di *controllo* e *risultato* da parte di un metodo che si sta concludendo

'PER NOI' LE ISTRUZIONI SI SUSSEGUONO:

- una alla volta,
- a partire dalla prima istr. del metodo **main**,
- tendenzialmente nello stesso ordine indicato dal testo del programma

... ma qui il '*tendenzialmente*' adombra cose importanti ...

- assegnamento

- assegnamento
- blocco di istruzioni

- assegnamento
- blocco di istruzioni
- istruzioni iterative
 - **while**
 - **for**
 - **do-while**

- assegnamento
- blocco di istruzioni
- istruzioni iterative
 - **while**
 - **for**
 - **do-while**
- istruzioni di salto condizionato
 - **if**
 - **if-else**
 - **switch**

- `assegnamento`
- blocco di istruzioni
- istruzioni iterative
 - **while**
 - **for**
 - **do-while**
- istruzioni di salto condizionato
 - **if**
 - **if-else**
 - **switch**
- `invocazione` di metodo, **return ;**

- assegnamento
- blocco di istruzioni
- istruzioni iterative
 - **while**
 - **for**
 - **do-while**
- istruzioni di salto condizionato
 - **if**
 - **if-else**
 - **switch**
- invocazione di metodo, **return ;**
- istruzione vuota

ESPRESSIONE

È il principale sotto-costrutto dell'istruzione —
abbraccia, come caso particolare, la *condizione*

Così come le variabili, anche le espressioni composte hanno un **tipo**,
che *di solito* non viene dichiarato dal programmatore, ma inferito
dal compilatore.

... ma qui il '*di solito*' adombra molte cose ...