Corso di Laurea Ingegneria Civile

Fondamenti di Informatica

Variabili e assegnazione

Aprile 2010

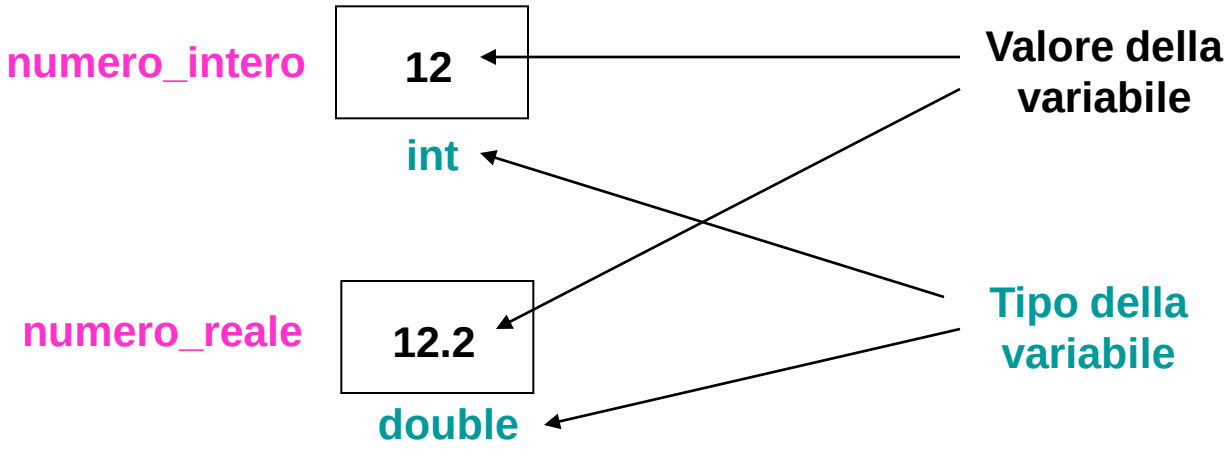
Contenuti

❑Variabili e assegnazione

- Variabili e aree di memoria
- Dichiarazione di variabili
- Assegnazione e accesso
- Costanti
- Variabili riferimento

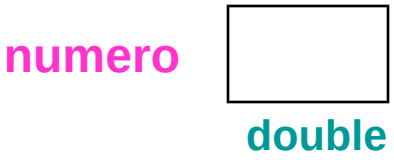
Variabile

- **Una variabile è un contenitore con un nome, che**
  - consente di memorizzare un valore
    - un valore di un certo **tipo** (e solo di quel tipo)
    - in momenti diversi dell'esecuzione, il valore memorizzato da una variabile può cambiare — ovvero, può "variare"
  - consente l'accesso al valore memorizzato



- **Le variabili sono utili per memorizzare valori da usare in modo ripetuto o in momenti successivi**

Assegnazione



- **Per memorizzare un valore in una variabile deve essere usata assegnazione**
  - **numero = 1.44;**

- **L'effetto di una assegnazione consiste nel**
  - calcolare il valore di una assegnazione, scritta a destra del simbolo =
  - memorizzare il valore calcolato nella variabile il cui nome è scritto a sinistra del simbolo =

Accesso al valore di una variabile

- **Il valore di una variabile può essere acceduto scrivendo il nome della variabile in un posto diverso dalla sinistra dell'operatore di assegnazione**

  - **System.out.println(numero);**
    - visualizza il valore di **numero**

- **Il nome di una variabile**
  - in alcuni casi viene utilizzato per accedere al valore memorizzato dalla variabile
  - in altri casi viene utilizzato per modificare il valore memorizzato dalla variabile

Variabili e aree di memoria

- **Una variabile è una astrazione per una "area di memoria"**
  - per area di memoria si intende un gruppo di celle di memoria
    - una variabile è una astrazione per un'area di memoria
  - un'area di memoria viene identificata dall'indirizzo dell'area di memoria (l'indirizzo della prima cella dell'area di memoria)
    - il nome di una variabile è una astrazione per l'indirizzo di un'area di memoria
  - un'area di memoria ha lo scopo di memorizzare un valore, opportunamente codificato mediante una sequenza di bit
    - il valore di una variabile è una astrazione per la sequenza di bit memorizzata da un'area di memoria
  - il valore memorizzato da un'area di memoria può essere interpretato solo conoscendo il tipo della codifica utilizzata per l'area di memoria
    - il tipo di una variabile è una astrazione per il tipo della codifica utilizzata per un'area di memoria
  - una unità di memoria fornisce le operazioni di lettura e scrittura
    - le operazioni di accesso e assegnazione a una variabile sono astrazioni per la lettura e la scrittura di un'area di memoria

Dichiarazione di variabili

- **Per poter usare una variabile, è necessario prima dichiararla**
  - il compilatore riserva così un'area di memoria in cui memorizzare valori del tipo della variabile
- **Una dichiarazione di variabile consiste di**
  - il **nome** della variabile – un identificatore
  - il **tipo** della variabile
    - il tipo dei valori che la variabile può memorizzare
  - prima il **tipo** e poi il **nome**

```
double numero;  // numero di cui si vuole
                // calcolare la radice quadrata
```

è sempre bene commentare la dichiarazione di una variabile per descrivere l'uso che se ne intende fare

- **È anche possibile dichiarare più variabili in un'unica dichiarazione**

```
double a, b, c;    // coefficienti di una equazione di secondo grado
```

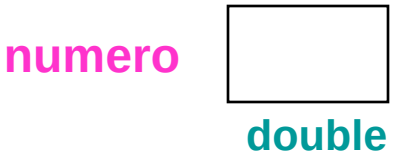
Variabili locali

- **le variabili dichiarate nell'ambito di un metodo si chiamano variabili locali del metodo**
  - **tra le variabili locali, un metodo può accedere solo a quelle che dichiara**
  - **un metodo non può accedere a variabili locali di un altro metodo**

Dichiarazioni di variabili locali

- **L'"esecuzione" di una dichiarazione di variabile locale ha l'effetto di**
  - allocare per la variabile un contenitore in grado di memorizzare un valore del tipo specificato nella dichiarazione
    - allocare vuol dire "riservare un'area di memoria"
  - associare il nome della variabile a tale contenitore

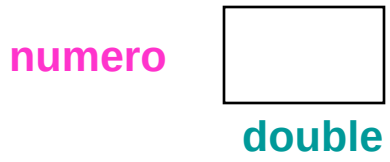
```
double numero;
```



  - effetto locale al metodo che contiene la dichiarazione
  - nessun valore memorizzato nella variabile
    - valore indefinito

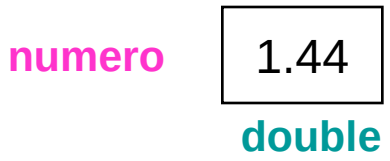
Assegnazione e accesso

```
double numero;
```



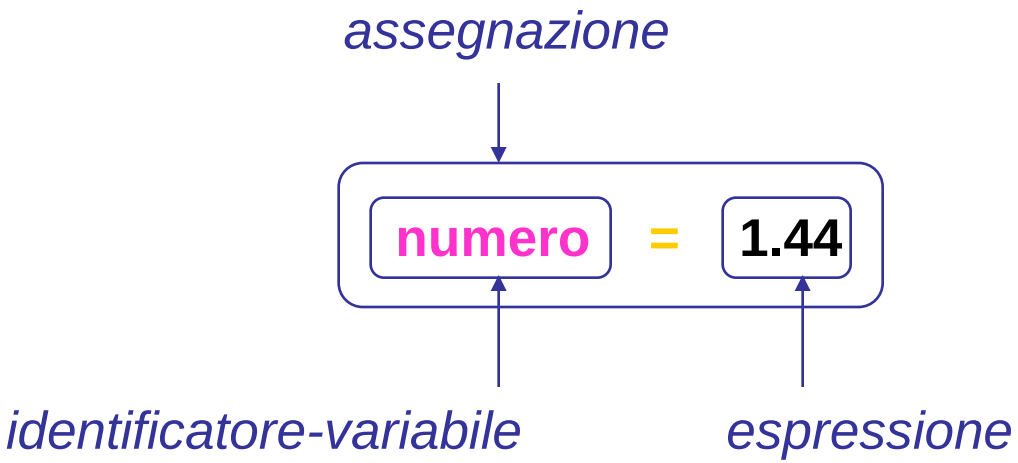
- Per memorizzare un valore in una variabile deve essere usata una istruzione di **assegnazione**

```
numero = 1.44;
```



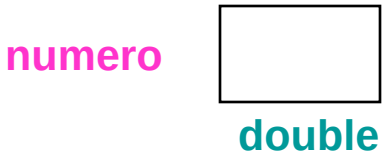
- il simbolo **=** è chiamato l'**operatore di assegnazione**

Vincoli in una assegnazione



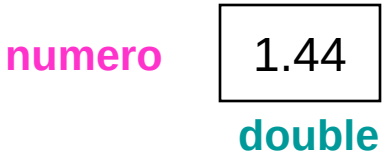
- **la variabile di nome *identificatore-variabile* deve essere stata dichiarata da una istruzione precedente nel metodo**
- **il tipo di *espressione* deve essere consistente con il tipo specificato della variabile di nome *identificatore-variabile* (che è il tipo specificato nella dichiarazione di tale variabile)**
  - in prima approssimazione, la variabile e l'espressione devono essere dello stesso tipo

Semantica dell'assegnazione



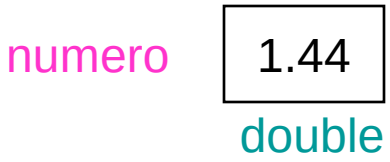

```
numero = 1.44;
```

- **Effetto di una assegnazione**
  - **calcola il valore v dell'espressione *espressione* scritta a destra dell'operatore di assegnazione**
  - **memorizza il valore v calcolato nella variabile *identificatore-variabile* scritta a sinistra dell'operatore di assegnazione**



Accesso a variabili

```
double numero;
numero = 1.44;
```



- **Dopo che è stato memorizzato un valore in una variabile, è possibile accedere al valore associato alla variabile**
  - mediante il nome della variabile usato come operando in una espressione

    System.out.println(numero);

- **È un errore accedere a una variabile a cui non è ancora stato assegnato un valore**

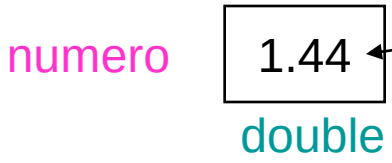
Due funzionalità per le variabili

- **Una variabile fornisce due funzionalità**
  - memorizzazione di un valore
  - accesso al valore memorizzato
- **Funzionalità selezionata sulla base di considerazioni sintattiche**
  - nome scritto nel lato sinistro di una assegnazione
    - memorizzazione
  - nome scritto nel lato destro di una assegnazione o altrove
    - accesso
  - ad esempio **t = s**;
    - la variabile **s** viene usata per accedere al valore che memorizza
    - la variabile **t** viene usata per memorizzare un valore

Assegnazione non è uguaglianza

- **L'operatore di assegnazione = non va confuso con un simbolo di uguaglianza**
  - si consideri ad esempio l'effetto delle seguenti istruzioni

```
double numero;
numero = 1.44;
```



In ciascun istante dell'esecuzione di un metodo o programma, una variabile può memorizzare un solo valore.
Tale valore può cambiare

seguite anche dalla seguente assegnazione

```
numero = numero+1;
```

(equivalente a numero++;)

ora numero vale 2.44

Le variabili sono indipendenti

❑ Le variabili sono indipendenti

- una assegnazione viene usata per modificare il valore di una sola variabile

```
double a, b;
a = 2.71;
b = a;
```


```
a = 3.14;
```

Costanti

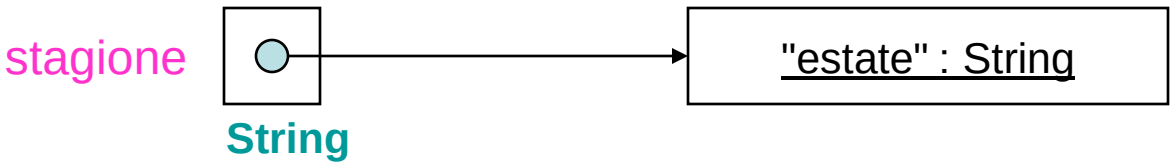
- **Una variabile può essere utilizzata per memorizzare un valore che non cambierà nel corso dell'esecuzione, ovvero un valore costante**
  - dichiarazione preceduta dalla parola chiave **final**
  - assegnazione del valore della costante contestuale alla dichiarazione – dichiarazione con inizializzazione

```
final double RAGGIO_CERCHIO = 10.0;
```

- **Uso delle costanti**
  - è possibile accedere al valore di una costante come a un'altra qualsiasi variabile
  - a parte l'inizializzazione, non è possibile assegnare altri valori alla costante

Variabili riferimento

- **In Java, due categorie di tipi**
  - **tipi primitivi**
    - un numero limitato di tipi predefiniti nel linguaggio Java — ad esempio, **double** e **int**
  - **tipi riferimento**
    - ogni classe definisce un diverso tipo riferimento
- **Una variabile riferimento è una variabile il cui tipo è un tipo riferimento (ovvero è il nome di una classe)**
  - una variabile riferimento di un tipo **C** è in grado di memorizzare un riferimento a un oggetto istanza di **C**



Stringhe

- **Un tipo di dato di uso molto comune è il tipo delle stringhe**
  - una **stringa** è una sequenza finita di caratteri
- **In Java le stringhe sono rappresentate dalla classe String — del package java.lang**
  - un oggetto **String** rappresenta una sequenza finita di caratteri
    - il valore (o contenuto) di un oggetto **String** è la stringa rappresentata da quell'oggetto
    - una stringa costante viene denotata dalla stringa stessa racchiusa tra doppi apici

```
String stagione;
stagione = "estate";
```

- **System.out.println(stagione);**

Cosa abbiamo visto finora

- Variabili e assegnazione
- Variabili e aree di memoria
- Dichiarazione di variabili
- Assegnazione e accesso
- Costanti
- Variabili riferimento

Riferimenti al libro di testo

- **Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al Capitolo 9**