

Struttura delle classi

1 Membri di una classe

I principali tipi di **membri** di una classe, cioè di elementi che possono essere presenti nel suo corpo, sono:

- campi
- campi statici
- costruttori
- metodi
- metodi statici

2 Campi

I **campi** (o **variabili di istanza**) sono variabili associate a ciascun oggetto:

- definiscono lo *stato* dell'oggetto
- lo spazio necessario a memorizzarli viene riservato all'interno dell'oggetto al momento della sua creazione
- i dati memorizzati nei campi vivono insieme all'oggetto, cioè esistono finché l'oggetto è accessibile
- se non sono **inizializzati esplicitamente** (es. `private int x = 0;`), vengono **inizializzati automaticamente** con un valore di default al momento della creazione dell'oggetto (ma è sempre bene dare valori espliciti)

Valori di default per l'inizializzazione dei campi

Tipo	Default
Interi	0
Virgola mobile	0.0
char	\u0000
boolean	false
Riferimento	null

Per accedere ai campi si utilizza la sintassi:

```
referimento.nome_campo
```

Il **referimento** può anche essere la pseudo-variabile **this**, che all'interno del codice di una classe si riferisce all'oggetto che esegue il metodo (o che si sta costruendo, nel caso di un costruttore). **this** può anche essere sottinteso, scrivendo solo il nome del campo.

Ai campi è possibile applicare alcuni **modificatori di visibilità**:

private: il campo non è accessibile all'esterno del codice della classe

public: il campo è accessibile ovunque sia visibile la classe

3 Campi statici

I **campi statici** (o **variabili di classe**), che si dichiarano utilizzando la parola riservata **static**, contengono informazioni comuni a tutta la classe:

- esistono indipendentemente dagli oggetti
- vengono creati dalla JVM al momento del caricamento della classe (che si verifica quando viene eseguita la prima istruzione contenente il nome della classe)
- come per i campi non statici, l'inizializzazione può essere automatica o esplicita

Per l'accesso, si utilizza la sintassi:

```
nome_classe.nome_campo_statico
```

Anche in questo caso, all'interno del codice di una classe è possibile accedere ai suoi campi statici scrivendo anche solo il nome del campo.

Ai campi statici possono essere applicati i modificatori di visibilità **private** e **public**.

4 Costruttori

Un **costruttore** è una porzione di codice usata per inizializzare in modo consistente un oggetto di una classe.

Se in una classe non sono presenti costruttori, il compilatore ne fornisce uno privo sia di argomenti che di codice.

Nel codice di un costruttore, è possibile richiamarne un altro della stessa classe, scrivendo come prima istruzione **this(lista_argumenti)**.

5 Metodi e metodi statici

I metodi definiscono il *comportamento* dell'oggetto.

Essi vengono invocati con la sintassi:

```
nome_oggetto.nome_metodo(lista_argomenti)
```

I **metodi statici**, dichiarati con la parola riservata `static`, appartengono invece all'intera classe (e non ai singoli oggetti). Per questo, all'interno del loro codice non è possibile accedere a membri non statici dell'oggetto corrente (che non esiste), cioè non si può utilizzare (neanche implicitamente) il riferimento `this`.

Al contrario, nel corpo dei metodi non statici è possibile accedere anche a membri statici.

La sintassi per l'invocazione di un metodo statico è:

```
nome_classe.nome_metodo(lista_argomenti)
```

All'interno del codice di una classe, è possibile invocare metodi (statici e non) della classe stessa scrivendo solo `nome_metodo(lista_argomenti)`, cioè omettendo il nome dell'oggetto o della classe.

Anche ai metodi (di entrambi i tipi) si possono applicare i modificatori di visibilità `private` e `public`.