

## Linee guida per la stesura del codice

Pierluigi Crescenzi<sup>1</sup>

<sup>1</sup>Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Firenze ([pierluigi.crescenzi@unifi.it](mailto:pierluigi.crescenzi@unifi.it))

### Sommario

Forniamo, in queste linee guida, un piccolo insieme di regole che andrebbero rispettate per la stesura del codice, in modo da facilitare il compito di chi dovrà, successivamente, leggerlo, comprenderlo e, eventualmente, modificarlo.

### Classi

Ogni file che definisce una classe deve essere organizzato nel modo seguente:

1. Un'eventuale istruzione `package`.
2. Una o più eventuali istruzioni `import`.
3. Un commento che spieghi lo scopo e il ruolo della classe all'interno del programma.
4. La definizione di una classe dichiarata `public`, la quale, a sua volta, deve essere strutturata in due parti: una contenente le componenti pubbliche e una contenente quelle private. Ciascuna di queste due parti deve essere organizzata nel modo seguente.
  - (a) Variabili di istanza.
  - (b) Campi statici.
  - (c) Costruttori.
  - (d) Metodi di istanza (una linea bianca va lasciata dopo ciascun metodo).
  - (e) Metodi statici (una linea bianca va lasciata dopo ciascun metodo).

### Variabili

Tutte le variabili d'istanza non finali devono essere private e tutte le componenti devono essere etichettate come `public` oppure `private`. Ogni qualvolta sia possibile, bisogna evitare variabili statiche (ad eccezione di quelle finali, come quelle che corrispondono a costanti). Le variabili statiche e finali (ovvero, le costanti) devono essere private, se nessun'altra classe ne fa uso. Non bisogna definire due variabili sulla stessa riga. Non bisogna usare costanti numeriche all'interno del codice, ma, piuttosto, definire delle costanti (fanno eccezione a questa regola, i valori 0, 1 e -1 che sono spesso

usate per inizializzare, incrementare o decrementare una variabile). Quando si dichiarano variabili di tipo array, le parentesi quadre vanno raggruppate con il tipo, non con la variabile. Inoltre, bisogna specificare esplicitamente i valori iniziali nella creazione di un array oppure inizializzarlo tramite un ciclo `for`.

### Metodi

Ogni metodo pubblico, ad eccezione del metodo `main` e dei metodi che sovrascrivono metodi di una libreria, deve avere un commento che spieghi lo scopo del metodo e il suo funzionamento. I metodi possono avere al massimo dieci linee di codice (esclusi la dichiarazione, i commenti, le linee bianche e le linee contenenti solo parentesi graffe).

### Controllo del flusso

La scrittura delle strutture di controllo è regolata da quanto è stato detto nel terzo capitolo del libro: le regole esposte in quel contesto non sono dunque ripetute in queste linee guida, ma devono esserne considerate parte integrante. Non bisogna usare il ciclo `for` per costrutti complessi, ma solo quando una variabile scorre da un valore ad un altro valore mediante un incremento o decremento costante: in caso contrario, va usato un ciclo `while`. Bisogna evitare l'istruzione `switch` e usare, al suo posto, la struttura `if-else`. Inoltre, bisogna evitare le istruzioni `break` e `continue`, facendo invece uso di una variabile di tipo `boolean` per controllare il flusso di esecuzione.

### Regole lessicali

Le seguenti regole lessicali, sebbene possano sembrare non molto importanti, risultano al contrario molto utili a rendere il proprio codice più leggibile.

## Assegnazione dei nomi

I nomi di identificatori sono regolati da quanto è stato detto nel libro: tali regole non sono dunque ripetute in queste linee guida, ma devono esserne considerate parte integrante. Tutte le costanti sono in lettere maiuscole, con eventualmente il simbolo `_` al loro interno. Bisogna usare nomi singolari per gli array: il motivo di ciò è che, quando viene usato, il nome dell'array si riferisce, generalmente, a un solo elemento dell'array stesso.

## Spazi

Uno spazio bianco va inserito intorno ad ogni operatore di assegnazione. Inoltre uno spazio bianco va inserito dopo un punto e virgola, dopo una parentesi tonda aperta e prima di una parentesi tonda chiusa (ad eccezione, delle parentesi incluse in un'espressione

aritmetica o Booleana) e prima di una parentesi tonda aperta e dopo una parentesi tonda chiusa (ad eccezione del caso in cui si stia usando l'operatore di conversione esplicita). Non bisogna inserire uno spazio bianco dopo una virgola.

## Parentesi graffe

La parentesi graffe aperta deve essere scritta sulla stessa riga della parola chiave, mentre quella chiusa deve essere allineata con la parola chiave.

## Commenti

Non bisogna usare commenti di riga per commenti che richiedono più di una riga: in tal caso, bisogna usare commenti di paragrafo.