

Java



lezioni di stile

alberto ferrari

naming

- × usare nomi **significativi** per tutti gli identificatori
 - × (nomi delle classi, dei campi, dei metodi, dei parametri, delle variabili ...)
- × evitare abbreviazioni
 - × per i metodi **accessor** il nome deve essere getAttributo()
 - × se restituiscono un valore **boolean** il nome deve essere isAttributo()
 - × (es. isVuoto())
- × i nomi delle **classi** iniziano con la lettera **maiuscola**
- × i nomi delle classi sono **sostantivi singolari**
- × i nomi dei **metodi** iniziano con la lettera **minuscola**
- × tutti i nomi utilizzano al loro interno una lettera maiuscola per rappresentare nomi composti (**camelCase**)
 - × (es. numeroStudenti)
- × costanti
 - × i nomi delle costanti sono formati da lettere **maiuscole** (es. PI)
 - × utilizzare **underscore** per separare nomi composti (es. NUMERO_STUDENTI)



layout

- * un livello di indentazione è formato da 4 spazi
- * tutte le istruzioni di un blocco sono indentate di un livello
- * le parentesi { } per le classi e i metodi sono sole su una riga

```
public int esempioMetodo()  
{  
    statements  
}
```

- * in tutti gli altri blocchi la parentesi aperta segue l'istruzione che stabilisce l'inizio del blocco
- * la parentesi chiusa è sola su una riga allineata all'istruzione che definisce l'inizio del blocco

```
while(condizione) {  
    statements  
}
```

```
if(condizione) {  
    statements  
}
```

```
else {  
    statements  
}
```

- * usare sempre le parentesi anche in caso di blocchi formati da una sola istruzione
- * inserire uno spazio prima e dopo ogni operando di una espressione
- * usare una riga vuota per separare logicamente parti di codice

documentazione

- × ogni classe inizia con un **commento** che contiene almeno:
 - × una sintetica **descrizione** della classe
 - × il nome dell'**autore**
 - × il numero di **versione** (o data)
 - × date due versioni di un codice deve essere possibile stabilire qual è la più recente
- × ogni **metodo** deve essere preceduto da un **commento**
- × i commenti devono essere in formato **Javadoc** (iniziare con /**)
- × inserire commenti al codice per chiarirne il significato
 - × non inserire commenti ovvi (*i programmatori conoscono Java!*)

- × **l'ordine di scrittura** di una classe deve essere:
 - × package statement
 - × import statements
 - × commento alla classe
 - × intestazione della classe
 - × attributi
 - × costruttori
 - × metodi

specifiche

- x gli attributi **non** devono essere pubblici (tranne gli attributi final)
- x utilizzare **sempre** un modificatore di accesso
 - x (non usare il modificatore di default)
- x importare **solo** le classi che si utilizzano, non interi package
 - x es. `import java.util.ArrayList;`
 - x e **non** `import java.util.*;`
- x inserire **sempre** un costruttore senza parametri
 - x (eventualmente con corpo vuoto)
- x chiamare **sempre** esplicitamente il costruttore della superclasse
- x nei costruttori inizializzare sempre **tutti** gli attributi

collezioni

- × utilizzare gli **iteratori** con le collezioni
- × per iterare una collezione utilizzare il costrutto **for-each**
- × se la collezione può essere **modificata** all'interno dell'iterazione
 - × utilizzare gli iteratori
 - × **non** un indice intero