

OBJECT CALISTHENICS

LE REGOLE DELL'OBJECT ORIENTED PROGRAMMING

1

UN SOLO LIVELLO DI INDENTAZIONE PER METODO

Evitare di avere più livelli di indentazione ci guida verso la scrittura di un metodo che faccia una ed una sola cosa.

2

NON USARE L'ELSE

Evitando il comando "else", il metodo risulterà più leggibile, avendo una linea di esecuzione principale che arriva in fondo, con qualche caso particolare.

3

ASTRAI PRIMITIVE E STRINGHE

Nessun parametri di metodi pubblici dovrebbe essere di un tipo primitivo o una stringa: crea astrazioni che rappresentino il significato della variabile.

4

FIRST CLASS COLLECTION

Come per primitive e stringhe, non usiamo array o liste come tipi: crea un astrazione che ti aiuti a gestire la lista sia in lettura che scrittura.

5

UNA CHIAMATA PER RIGA

Evita assolutamente le catene di chiamate: ogni classe dovrebbe comunicare solo con i suoi collaboratori più stretti.

6

NON USARE ABBREVIAZIONI

Le abbreviazioni non hanno nessuna utilità da quando non scriviamo più SMS: scrivi nomi esaustivi ed espliciti, a costo di essere verboso e prolisso.

7

MANTIENI ELEMENTI PICCOLI

Come per l'indentazione, mantenere classi e metodi di piccole dimensioni ci aiuta a far sì che si occupino di una ed una sola cosa.

8

MASSIMO 2 VARIABILI D'ISTANZA

Più variabili d'istanza abbiamo, minore è la coesione della classe: Sforziamoci di avere meno variabili d'istanza possibili, idealmente non più di 2.

9

NO GETTERS/SETTERS/PUBLIC PROPERTIES

L'idea originale di OOP è quella di oggetti che collaborano scambiandosi messaggi, evitiamo quindi di esporre dati interni: TELL, DON'T ASK.

10

TUTTE LE CLASSI DEVONO AVERE UNO STATO

Non creare metodi statici o classi di utility; crea solo elementi con una chiara responsabilità ed uno stato da mantenere per perseguirla.