Ente Ospedaliero Cantonale

Naming conventions

Presentazione corso Alcor

Java Naming Convention

- Class: **nomi**, camelCase e cominciare con la maiuscola (String, Integer, Dog, Animal, …)
- Interface: **aggettivi** , camelCase e cominciare con la maiuscola (Runnable, Listener, …)
- Method: **verbi**, camelCase e cominciare con la minuscola (main, print, addTerms,…)
- Variable: camelCase e devono cominciare con la minuscola (firstName, counter, …)
- Constant: snake_case e devono essere tutte maiuscole (WIDTH, FIRST_DAY_OF_WEEK, …)

Java Naming Convention

È davvero quello che intendiamo?

```
JSONObject doAllTheStuff = doAllTheStuff(patientId, edsId, passageId, sermedId, sessionId, scopeId);
```

```
final SistemaGestPazUscita sistemaGestPazUscita = new SistemaGestPazUscita();
try {
    sistemaGestPazUscita.faitutto();
} catch (Exception e) {
```

Java Naming Convention

Perché i nomi sono importanti

- Capire cosa fa/a cosa serve senza dover leggere codice/documentazione
- Codice più comprensibile → più facile cercare
- Codice più comprensibile → più facile parlarne
- Codice più comprensibile → più facile da far capire

Java Naming Convention

Facile:

- I nomi devono essere parlanti
- I nomi devono rispettare quello che rappresentano
- I nomi non devono essere abbreviati
- I nomi non devono essere lunghi
- I nomi devono essere leggibili

Java Naming Convention

Perché è difficile e spesso non lo facciamo?

- Non c’è uno standard
- Siamo diversi
- Il nome va spesso dato prima del contenuto
- Deve essere mantenuto

Java Naming Convention

Quindi?

- Prendiamoci il tempo di pensare ai nomi
- Refactoring anche dei nomi
- Non perdersi in dettagli
- Ricordiamo di non essere gli unici a leggere il codice

Java Naming Convention

Grazie per l’attenzione