Programmazione Orientata agli Oggetti in Linguaggio Java

Test e Correzione: Introduzione

versione 1.0

G. Mecca – Università della Basilicata – mecca@unibas.it

Sommario

- Panoramica
- Correttezza
- Manutenibilità
- In Queste Lezioni

Panoramica

- In questo modulo
  - ⇨continuiamo a parlare di tecniche e strumenti per ottenere codice di qualità
- In particolare
  - ⇨ci concentriamo su due qualità fondamentali
  - ⇨la correttezza
  - ⇨la manutenibilità

Correttezza

- Un obiettivo di qualità a breve termine
  - ⇨la correttezza
  - ⇨il software deve implementare interamente e correttamente le specifiche
  - ⇨ovvero realizzare correttamente i casi d'uso
- Le tecniche fondamentali
  - ⇨lo strumento principale per le verifiche di correttezza sono i test
  - ⇨e il debugging (correzione)

Correttezza

- Verificare il codice con i test
  - ⇨il gruppo di sviluppo deve mettere a punto un "piano dei test"
  - ⇨che deve essere eseguito con regolarità
  - ⇨spesso c'è un gruppo di lavoro appositamente destinato a effettuare test
- Attenzione
  - ⇨il piano dei test deve verificare la robustezza del codice

Correttezza

- Correggere il codice
  - ⇨quando viene rilevato un errore logico ("bug") attraverso i test o attraverso prove interattive
  - ⇨è necessario procedere al debugging
- Due tecniche fondamentali
  - ⇨utilizzo di un debugger
  - ⇨utilizzo di stampe di debug

Manutenibilità

- Ma esiste un'altra prospettiva
- Un obiettivo di qualità a lungo termine
  - ⇨la manutenibilità
- La regola n. 1 del programmatore
  - ⇨"le specifiche cambiano con il tempo"
  - ⇨e i sistemi software si devono adattare
  - ⇨la possibilità di adeguare facilmente e rapidamente il codice alle nuove specifiche è una caratteristica essenziale

>> alcuni esempi

Manutenibilità

- Di conseguenza
  - ⇨la qualità deve essere perseguita nel tempo
  - ⇨mantenendo alto il livello del codice anche in caso di modifica delle specifiche
- "Adattarsi ai cambiamenti"
  - ⇨una massima dell'eXtreme Programming
  - ⇨essere consapevoli che il software cambia
  - ⇨e adottare strumenti adeguati a gestire i cambiamenti mantenendo alta la qualità

Manutenibilità

- Per garantire la manutenibilità
  - ⇨alcune caratteristiche essenziali
- Tecniche per migliorare la manutenibilità
  - ⇨curare l'organizzazione del codice >> ruoli e responsabilità
  - ⇨curare la leggibilità del codice >> convenzioni di stile
  - ⇨automatizzare i test
  - ⇨semplificare il debugging

Manutenibilità

- Automatizzare i test
  - ⇨effettuare modifiche al codice potenzialmente introduce nuovi errori
  - ⇨è indispensabile eseguire il piano dei test dopo ciascuna modifica effettuata, soprattutto a distanza di tempo
  - ⇨questo obiettivo è raggiungibile solo se i test sono automatizzati
  - ⇨ovvero se sono "test di regressione"

Manutenibilità

- Test di regressione
  - ⇨test completamente automatizzato
  - ⇨eseguibile rapidamente, ripetutamente, senza intervento interattivo
- Obiettivo
  - ⇨ripetere frequentemente l'esecuzione dei test
  - ⇨per intercettare eventuali "regressioni" (passi indietro) introdotti nel codice dalle modifiche

Manutenibilità

- In caso di bug (errori)
  - ⇨è necessario procedere al debugging
  - ⇨se il software si evolve, è un'attività che può avvenire frequentemente
  - ⇨è indispensabile utilizzare strumenti semplici ed efficaci
  - ⇨che consentano di effettuare rapidamente il debugging

Correttezza

- Sistema di logging
  - ⇨sistema che permette di includere nel codice stampe di debug, che descrivono il funzionamento dell'applicazione
  - ⇨e di produrle solo su richiesta dell'utente (durante le sessioni di correzione)
  - ⇨oltre che in formati e su supporti diversi
  - ⇨escludendole nelle versioni in produzione

In Queste Lezioni

- In questo ciclo di lezioni
  - ⇨sviluppo di test di regressione e strumenti collegati (framework per il test)
  - ⇨utilizzo di tecniche di logging e debugging e strumenti collegati (debugger, framework per il logging)

In Queste Lezioni

- I progetti di riferimento
  - ⇨tutti quelli visti in precedenza
- Infatti
  - ⇨tutti i progetti dei moduli precedenti sono stati sviluppati con test di regressione
- In questo modulo
  - ⇨concludiamo la discussione del codice discutendo sintassi e semantica dei test

Riassumendo

- Panoramica
- Correttezza
- Manutenibilità
- In Queste Lezioni

Termini della Licenza