

Programmazione Orientata agli Oggetti in Linguaggio Java

Eccezioni: Introduzione

versione 3.0

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons
(vedi ultima pagina)

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



Eccezioni: Introduzione >> Sommario



Sommario

- Panoramica
- Robustezza
- I Progetti di Riferimento



Panoramica

- In questo modulo
 - ⇒proseguiamo l'analisi metodologica cominciata nel modulo precedente
 - ⇒parliamo di tecniche per garantire la qualità del codice
- Un argomento fondamentale
 - ⇒molti sviluppano codice
 - ⇒pochi sviluppano codice di qualità



Panoramica

- Le qualità principali del software
 - ⇒in ordine di priorità
 - ⇒correttezza
 - ⇒usabilità
 - ⇒manutenibilità
 - ⇒efficienza



Panoramica

○ Correttezza

- ⇒ rispondenza alle specifiche, ovvero implementazione corretta dei requisiti
- ⇒ robustezza, ovvero capacità di resistere a condizioni di utilizzo non completamente corrette
- ⇒ si tratta evidentemente del requisito fondamentale



Panoramica

○ Usabilità

- ⇒ rispondenza alle esigenze dell'utente
- ⇒ "fare correttamente le cose giuste in poco tempo"
- ⇒ semplicità di utilizzo e di apprendimento
- ⇒ efficacia nell'utilizzo degli utenti
- ⇒ soddisfazione nell'utilizzo degli utenti
- ⇒ negli ultimi anni ha acquisito un'importanza sempre maggiore



Panoramica

- Manutenibilità

- ⇒ capacità di essere adattato ai cambiamenti che intervengono nel tempo
- ⇒ è un requisito importantissimo: l'80% del lavoro sul software riguarda la manutenzione

- Molti diversi aspetti

- ⇒ semplicità, leggibilità, corretta organizzazione, portabilità



Panoramica

- Efficienza

- ⇒ utilizzo efficiente delle risorse di calcolo
- ⇒ tempo, memoria, rete

- Tradizionalmente

- ⇒ considerato essenziale (complessità computazionale)
- ⇒ al giorno d'oggi importante ma meno centrale di altri aspetti (aumento della potenza di calcolo)



Panoramica

- Nelle precedenti lezioni
 - ⇒ alcune tecniche essenziali per la qualità
 - ⇒ in particolare per la corretta organizzazione
- Ruoli e Responsabilità
 - ⇒ incapsulamento
 - ⇒ strati applicativi
 - ⇒ architettura di riferimento



Panoramica

- In questo modulo
 - ⇒ ci concentriamo sulla correttezza e sulla robustezza
- In particolare
 - ⇒ discutiamo la gestione delle eccezioni
- Successivamente
 - ⇒ tecniche di test e di correzione



Robustezza

- Robustezza

- ⇒ qualità strettamente collegata alla correttezza

- ⇒ il software deve comportarsi correttamente anche in condizioni eccezionali

- Eccezione

- ⇒ qualsiasi situazione in cui il programma non può svolgere normalmente il proprio lavoro



Robustezza

- Possibili cause di eccezioni

- ⇒ errori interni di programmazione

- ⇒ utilizzo "scorretto" da parte dell'utente

- ⇒ malfunzionamento di sistemi esterni

- Gestione delle eccezioni

- ⇒ la tecnica per garantire la robustezza è la gestione delle eccezioni

- ⇒ ovvero riconoscere le condizioni eccezionali e cercare di mettere riparo senza abortire



Robustezza

- Gestire condizioni eccezionali

- ⇒ i linguaggi OO forniscono istruzioni ad hoc

- ⇒ per sollevare eccezioni segnalando il verificarsi di condizioni eccezionali >> istruzione throw

- ⇒ e per gestire le eccezioni nel caso si verificano >> blocchi try/catch/finally



I Progetti di Riferimento

- Gestione delle eccezioni

- ⇒ una caratteristica pervasiva del linguaggio Java

- ⇒ compare in tutti i progetti del modulo precedente

- In particolare ci concentreremo su

- ⇒ Indovina il Numero

- ⇒ la classe Console



Riassumendo

- Panoramica
- Robustezza
- I Progetti di Riferimento



Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.