

# Simulatore di Gestione della Memoria di un Elaboratore SiGeM



<http://stylosoft.altervista.org>  
[stylosoft@gmail.com](mailto:stylosoft@gmail.com)

Risultati\_Analisi

17 Marzo 2008

Documento Esterno - Formale - v1.0

Risultati\_Analisi\_1.0.pdf

## Redazione:

Luca Rubin

## Revisione:

Daniele Bonaldo

## Approvazione:

Luca Rubin

## Lista di distribuzione:

Prof. Vardanega Tullio  
Prof. Palazzi Claudio  
Stylosoft

## Registro delle modifiche:

Versione	Data	Descrizione delle modifiche
1.0	17/03/08	Correzioni grammaticali, preparazione documento per la revisione RQ, avanzamento versione 1.0
0.1	02/03/08	Prima Stesura

Versione:  
1.0

Creazione documento:  
02/03/08

Ultima modifica:  
17/03/08

Pagina 1 di 5

# Simulatore di Gestione della Memoria di un Elaboratore SiGeM



<http://stylosoft.altervista.org>  
[stylosoft@gmail.com](mailto:stylosoft@gmail.com)

## Sommario

In questo documento vengono raccolti i risultati dell'analisi statica sul prodotto SiGeM.

## Indice

1 Introduzione .....	3
1.1 Scopo del documento .....	3
1.2 Strumenti utilizzati.....	3
2 Dettaglio dei difetti individuati.....	3

## 1 Introduzione

### 1.1 Scopo del documento

Tale documento ha lo scopo di chiarire e rendere note le modalità con cui si è svolta l'analisi statica del codice prodotto dalla fase di codifica. La creazione di tale documento è stata voluta per permettere, al piano di qualifica, di rimanere quanto più leggibile ed efficiente.

### 1.2 Strumenti utilizzati

Per ottenere quanto riportato in questo documento, sono stati utilizzati gli strumenti:

- FindBugs
- RefactorIT
- PMD

Tali strumenti sono prevalentemente integrati in NetBeans (il nostro ambiente di sviluppo). Per questo motivo i risultati ottenuti sono stati generati in modo automatico e veloce.

## 2 Dettaglio dei difetti individuati

Attraverso l'utilizzo di FindBugs sono emersi i seguenti errori (tra parentesi è riportato il numero di errori rilevati per ciascuna categoria):

- denominazione errata dei metodi rispetto a quanto specificato nelle norme di codifica (JCC) (40)
- assegnazioni errate a variabili (2)
- probabili vulnerabilità del codice, dovute al ritorno di oggetti modificabili da altre componenti a insaputa del programmatore (3)
- visibilità dei campi dati o dei metodi assegnate in modo da poter

causare possibili vulnerabilità del codice (2)

- inefficienza nella chiamata di alcuni costruttori (7)
- possibilità di marcare alcune classi (anche interne) come statiche (9)
- inefficienza nella chiamata di alcuni metodi (10)
- variabili che non vengono mai lette o scritte (3)
- variabili che non vengono mai lette (2)
- dereferenziazione di variabili nulle (1)
- utilizzo del metodo equal() senza la sua ridefinizione (3)
- cattura di eccezioni mai sollevate (1)

Ad ogni errore sopra elencato è stato posto rimedio. E' da sottolineare il fatto che non tutti problemi individuati da FindBugs sono stati classificati come errori in quanto in alcuni casi le indicazioni fornite dal programma di analisi, sono da considerarsi semplici linee guida, che non sempre possono essere rispettate.

Grazie ai software RefactorIT e PMD sono state ricavate le metriche e gli indicatori necessari per la valutazione del codice prodotto. Sulla base di queste misurazioni sono state apportate le dovute modifiche ai sorgenti, in modo da risolvere o attenuare eventuali problemi riscontrati.

Le metriche che sono state oggetto della valutazione da parte dei verificatori sono state:

- complessità ciclomatica: misura la complessità del codice; un valore troppo elevato comporta difficoltà di verifica del codice, oltre a possibile inefficienza del programma.

Complessità media	Complessità massima	Numero correzioni apportate
3	17	1

- linee di codice per file: tiene conto delle linee totali di cui si compone un file sorgente (codice più commenti). Un valore troppo elevato rende difficile la verifica; inoltre il codice risulta essere meno comprensibile. La dove è stato rilevato un valore troppo elevato, è stata valutata la possibilità di dividere la classe in più classi distinte.

Lunghezza media (in righe)	Lunghezza massima (in righe)	Numero correzioni apportate
254	1774	2

- densità dei commenti: viene misurata tramite il rapporto tra linee di codice e linee di commento. Un rapporto troppo basso può essere indice di un codice difficile da mantenere e da verificare in quanto difficile da comprendere.

Densità media	Densità minima	Numero correzioni apportate
0,57	0,11	36

Dato che gli strumenti elencati al punto 1.2 sono integrati nel nostro ambiente di sviluppo (NetBeans) non ci è possibile riportare parte delle anomalie riscontrate in quanto, tali segnalazioni avvengono contemporaneamente alla stesura del codice. Questo fatto comporta che il programmatore è in grado di correggere immediatamente parte di tali anomalie, alleggerendo le attività di verifica.