

Java Modeling Language

OpenJML e KeY a confronto

Giacomo Intagliata,

Matricola: T16501

Rafael Ventura,

Matricola: 08286A

Cos'è Java Modeling Language?

Linguaggio per specifiche formali

Utilizzato per stabilire la correttezza di un programma rispetto a un modello matematico

Java Modeling Language

Logica di Hoare

JML non letto
dal compilatore
Java

Sintassi simile
a Java

Metodologia
"Progettazione
a Contratto"

Progettazione a contratto

Design By Contract

L'idea centrale del DBC è che le entità software hanno degli obblighi nei confronti di altre entità in base a regole formalizzate tra di esse.

"Contratto"

Ogni classe ed ogni classe che la utilizza definiscono tra loro un "contratto".
Garantire la validità del contratto.
Definito all'interno del codice

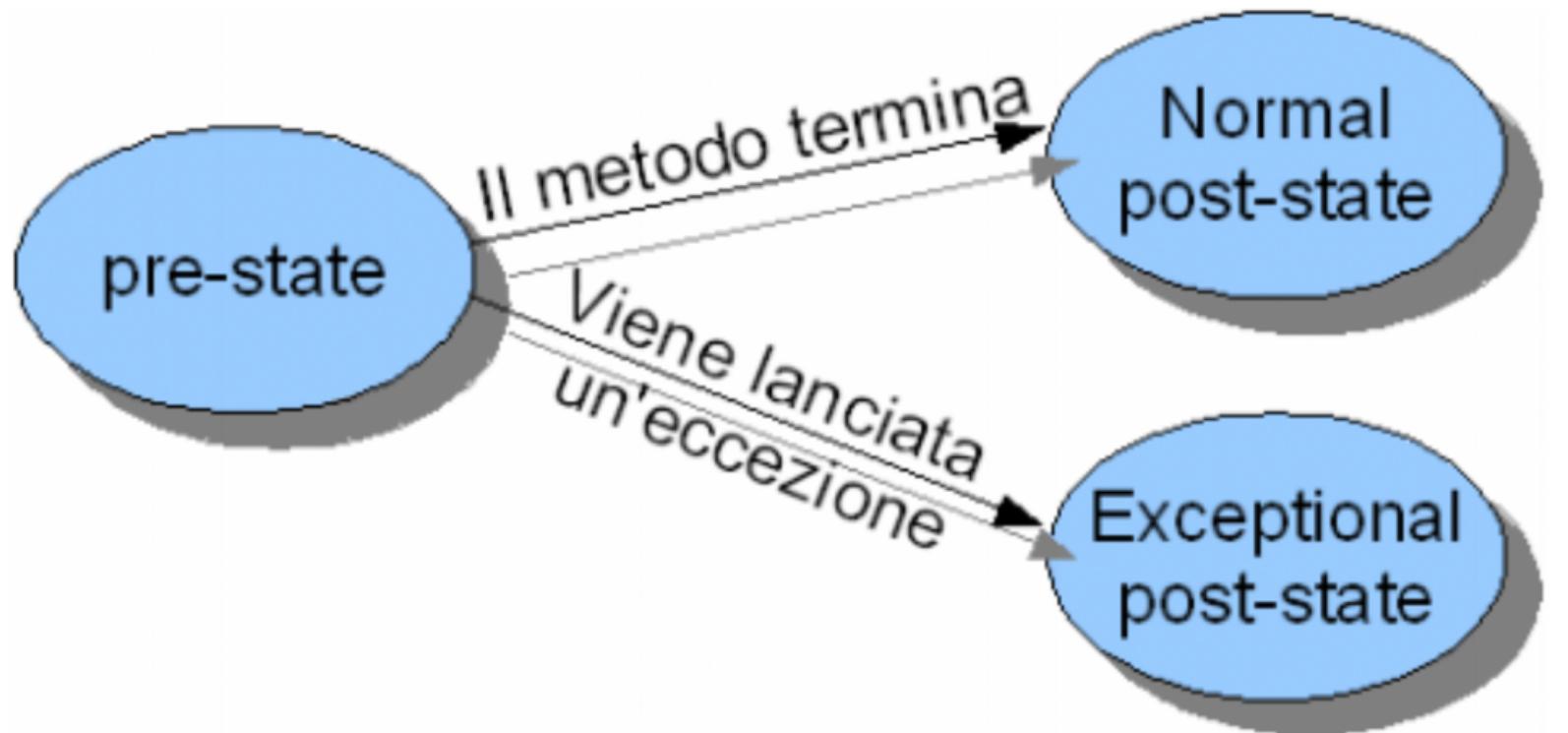
Precondizioni

e

Postcondizioni

Strutturano i contratti da verificare.

Stati JML



- **Normal post-state:** in questo caso il metodo termina senza lanciare eccezioni quindi le postcondizioni sono soddisfatte.
- **Exceptional post-state:** in questo caso termina invece con il lancio di un' eccezione, permessa dalla clausola "signals_only".

Direttive JML

REQUIRES

Definisce una pre-condizione sul metodo

SIGNALS

Definisce una condizione in base alla quale deve essere lanciata una eccezione dal metodo che segue

ASSERT

Definisce asserzione JML

INVARIANT

Definisce una proprietà invariante della classe

ENSURES

Definisce una post-condizione sul metodo

ALSO

Indica che un metodo deve ereditare le condizioni dalla sua superclasse

ASSIGNABLE

Indica oggetti locali al metodo e che possono essere modificati

Espressione JML

\RESULT

Permette di accedere al
valore di ritorno del
metodo

\OLD(<NAME>)

Riferimento al valore della
variabile <NAME>

A<==>B

A sse B

\FORALL<DOMINIO>;<RANGE_VALORI>;
<CONDIZIONE>

Quantificatore universale in un RANGE DI VALORI
in un certo DOMINIO che rispettano una
CONDIZIONE

JML Opensource ed altri modi di usarlo

- JML è Open Source
- È possibile fare assertion checking a runtime, static analysis, verifica formale tramite theorem prover, runtime debugging, unit testing, documentation ecc....

OpenJML

Cos'è?

Strumento di verifica del
programma per
programmi Java annotati
con JML

Funzionamento dietro le quinte

Traduce le specifiche scritte
in JML nel formato SMT-LIB e
passa il problema da provare
ad un SOLVER

Sviluppato da
David Cok

Successore di ESC/JAVA2

The KeY Project



- KeY è un tool che permette la verifica interattiva di programmi Java
- La forza di KeY è nel verificare algoritmi difficili (es: sorting) dove abbiamo operazioni intere con overflow e bisogna trovare quantificatori e lemmi ispezionando e interagendo con lo stato di verifica
- Key può anche verificare dei problemi ‘più semplici’, ma è accortezza del programmatore controllare che il programma non contiene feature Java non supportate, come multithreading o librerie non adatte

Confronto tra OpenJML & Key

Differenze
Strutturali

Differenze
Sintattiche

Differenze
Semantiche

Differenze Sintattiche

Keywords	JML	KeY	OpenJML
\sum \product \num_of	Yes	Yes	No
\strictly_nothing strictly_pure	No	Yes	No
\not_assigned	Yes	No	No
\bSum \bProduct	No	Yes	No
\locset \intersect \set_union	No	Yes	No
\distinct	No	No	Yes
\index	No	Yes	Yes
Certain Java arithmetic: % ^	Via Java	No	Yes

Differenze Semantiche

- Controllo compilazione
- Controllo visibilità
- Sicurezza della memoria e comportamento anomalo
- Controllo sull'inizializzazione

Link Utili

- Boerman, Jan, et al. “Reasoning about JML: Differences between KeY and OpenJML.” KeyOpenJML, University of Twente, The Netherlands, 2017,
<https://wwwhome.ewi.utwente.nl/~marieke/KeyOpenJML.pdf>.
- JML Tutorial. <https://www.openjml.org/tutorial/>
- Ahrendt, Wolfgang, et al. "Deductive Software Verification - the Key Book: From Theory to Practice." Springer Berlin Heidelberg, 2016.

Grazie per l'attenzione

Giacomo Intagliata,
Matricola: T16501

Rafael Ventura,
Matricola: 08286A