



TÉCNICO LISBOA

Tracer

Grupo 4

Alberto Carvalho, 66933

David Duarte, 68505

José Cavalheiro, 64804

Organização do código

- **Trace:** responsável por guardar toda a informação originada pelo TraceVM ou TraceVMExtended. Implementa método print que imprime a informação relativa ao objecto recebido como argumento
- **TraceInfo:** guarda a informação relativa a cada ocorrência em que o objecto foi usado.
- **Role:** enumerado que mapeia papéis no prefixo de saída adequado

Organização do código

- **TraceTranslator:** responsável por instrumentar chamadas a métodos, construtores
- **TraceTranslatorExtended:** subclasse de TraceTranslator. Instrumenta também uso de handlers, fields, casts e arrays

Organização do código

- **TraceVM**: responsável por associar o TraceTranslator ao *class loader* e delegar o carregamento de determinadas classes
- **TraceVMExtended**: subclasse de TraceVM. Responsável por associar o TraceTranslator ao *class loader*

TraceInfo

TraceInfo

- Guarda uma String que contém o *long name*, o nome do ficheiro e a linha relativos à invocação em que o objecto foi usado.
- Guarda o papel (role) associado à TraceInfo usado para identificar se a TraceInfo é relativa ao argumento, ao retorno, *cast*, *handler*, à escrita de um *field* ou à leitura de um *field*
- Responsável por gerar uma String, contendo a informação, que irá ser imprimida

Trace

Trace

- Guarda a informação num:

`IdentityHashMap<Object, LinkedList<TraceInfo>>`

- Quando é adicionada nova informação (TraceInfo) relativa a um dado Object, são detectados ciclos, evitando duplicar informação.
- O método print obtém a lista de TraceInfo(s) associada ao objecto recebido como argumento. Para cada elemento imprime a informação usando o método da própria classe TraceInfo

Trace/TraceInfo

Vantagens:

- A TraceInfo ao guardar uma String evita redundância de informação, pois usamos a propriedade de re-utilização de Strings do Java
- Ao usarmos um IdentityHashMap podemos ter acesso directo à informação relativa a um dado objecto usando a sua referência

Trace/TraceInfo

Abordagens alternativas:

- Usar ArrayList para guardar as TraceInfo
 - Vantagem:
 - Maior simplicidade de implementação
 - Desvantagem:
 - Acesso mais lento à informação relativa a um dado objecto

TraceTranslator

TraceTranslator

- Quando uma classe é carregada é chamado um método que irá instrumentar as chamadas a métodos e construtores
- Existem vários métodos auxiliares de forma a evitar duplicação de código
- Contém métodos cuja chamada vai ser injectada no código das classes instrumentadas

TraceTranslator

- Para instrumentar a chamada a métodos e as chamadas a construtores injecta-se a chamada a método antes e depois da execução do método de forma a garantir sequência temporal
- Para instrumentar a chamada a um método cujo retorno seja *void*, apenas se injecta chamada a método antes do método ser executado
- Chamada a método injectada determinará papel de cada TraceInfo criada

TraceTranslatorExtended

TraceTranslatorExtended

- Subclasse de TraceTranslator
- Re-utiliza maioria dos métodos criados no TraceTranslator de forma a evitar repetição de código
- Instrumenta chamadas a métodos, construtores, *arrays*, acesso para leitura ou escrita de *fields*, *casts* e *handlers*

TraceTranslatorExtended

- Formato de impressão da informação relativa a:
 - Leitura de *field*:
 - * Read Field: *behaviour* on *file:line*
 - Escrita em *field*:
 - * Write Field: *behaviour* on *file:line*
 - *Cast*:
 - * Cast: *behaviour* on *file.java:line*
 - *Handler*:
 - * Handler: *behaviour* on *file:line*

TraceVM

TraceVM

- Associa TraceTranslator ao *class loader*
- Delega o carregamento das classes: Trace, Translator e Role de forma a evitar que sejam carregadas por *class loaders* diferentes
- Executa a main da classe cujo nome se recebeu como argumento

TraceVMExtended

TraceVMExtended

- Subclasse de TraceVM
- Associa TraceTranslatorExtended ao *class loader*
- Delega o carregamento das classes: Trace, Translator, TranslatorExtended e Role
- Executa a main da classe cujo nome se recebeu como argumento