

UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
LINGUAGENS PARA APLICAÇÃO COMERCIAL

MARCUS VINÍCIUS LACERDA FAGUNDES

Mini-Seminário - Java Reflection API

Salvador-BA

Março de 2009

MARCUS VINÍCIUS LACERDA FAGUNDES

Mini-Seminário - Java Reflection API

Atividade apresentada como requisito para avaliação no curso de Ciência da Computação da Universidade Federal da Bahia na disciplina Linguagens Para Aplicação Comercial orientada pelo professor Adonai Estrela Medrado.

Salvador-BA

Março de 2009

SUMÁRIO

INTRODUÇÃO	p. 3
JAVA REFLECTION API	p. 3
2.1 Motivação para Desenvolvimento	p. 3
2.2 Principais características e restrições	p. 3
2.3 Licença	p. 4
CONCLUSÃO	p. 5

INTRODUÇÃO

Reflexão, ou do inglês, Reflection, é o processo no qual um software modifica e observa a estrutura e comportamento dele próprio, em tempo de execução. Em Java, é possível obter informações sobre vários recursos, como campos, métodos, construtores de classe, entre outros. Pode ser considerado um tipo particular de meta-programação.

Em Java, tais funcionalidades são obtidas utilizando-se *Java Reflection API*, conjunto de classes nos pacotes, *java.lang* e *java.lang.reflect*. Reflexão está disponível em Java desde a sua versão 1.1.

JAVA REFLECTION API

2.1 Motivação para Desenvolvimento

Reflexão pode ser utilizada por um programa que tem uma arquitetura orientada a componentes, com qual se é possível, remover e adicionar componentes, em tempo de execução, de acordo com o objetivo desejado. Além disso, é utilizado em depuradores de código, e até mesmo em máquinas virtuais de algumas linguagens de programação.

Segundo [1], um simples exemplo tangível, em Java, é a utilização em *Java Beans* ou *Value Objects*, em outras linguagens. Que são classes com apenas campos privados, e métodos de acesso para estes, e um construtor padrão. Diversas ferramentas utilizam Reflexão para obter propriedades destes componentes do Java, para serem carregados dinamicamente.

2.2 Principais características e restrições

- Os construtores da classe: nome e parâmetros.
- Campos da classe: nome, modificadores e tipo.
- Métodos da classe: nome, modificadores, tipo de retorno, parâmetros, e *Exceptions* que o método pode retornar.
- Instanciar objetos, invocar métodos e modificar propriedades.

- Manipular vetores e simular o operador *instanceof*.
- Overhead: Por conta dos tipos que são dinamicamente resolvidos, algumas otimizações não são realizadas. Consequentemente, operações utilizando Reflexão são mais lentas.
- Restrições de Segurança: Reflexão requer permissão em tempo de execução. Isso é relevante quando executada em um contexto de segurança restrita como em Aplets.
- Exposição aos campos e métodos, internos: Reflexão permite operações como, acesso a campo e métodos privados, o que pode gerar efeitos inesperados, e pode destruir a portabilidade.

2.3 Licença

O código da Java Reflection API, desenvolvido pela Sun, está escrito sob licença GNU GPL, versão 2.

2.3.1 Restrições e exigências

- O software que utilizam está licença, pode ser utilizado para qualquer propósito;
- Necessário disponibilizar o código fonte, para que o software seja estudado e adaptado: As modificações são regidas pela licença.

2.3.2 Obrigações do desenvolvedor

- Qualquer modificação, por terceiros, sob o código-fonte também deve ser disponibilizada.

2.4 Plataformas e sistemas operacionais suportados

- Plataformas: X86, x64, Intel Intanium, SPARC, Java Platform Standard Edition.
- Sistemas Operacionais: Linux, Solaris, Windows.

CONCLUSÃO

Reflexão é uma ferramenta poderosa, mas que deve ser utilizada apenas quando realmente necessária, e que segundo os próprios desenvolvedores do recurso na linguagem Java [3], deve ser usada por desenvolvedores que dominam os fundamentos da linguagem.

É relevante saber que Java Reflection API, trata de Reflexão para o caso específico para a linguagem de programação Java. Mas que este recurso se encontra em diversas linguagens comerciais tais como C#, PHP, Python, Ruby, entre outras.

Por fim, é necessário deixar claro que a maioria dos sistemas de informações desenvolvidos jamais precisarão do uso de Reflexão, no entanto, este recurso será uma solução elegante para alguns problemas, e imprescindível para outros, daí a importância desta.

REFERÊNCIAS

- [1] Using Java Reflection. Disponível em: <<http://java.sun.com/developer/technicalArticles/ALT/Reflection/index.html>>. Acessado em: 15 de Março de 2009.
- [2] Java Reflection API. Disponível em: <<http://www.docjar.com/docs/api/sun/reflect/package-index.html>> package_description. Acessado em: 15 de Março de 2009.
- [3] Trail: The Reflection API. Disponível em: <<http://java.sun.com/docs/books/tutorial/reflect/index.htm>>. Acessado em: 15 de Março de 2009.
- [4] Reflection (computer science). Disponível em: <[http://en.wikipedia.org/wiki/Reflection_\(computer_science\)](http://en.wikipedia.org/wiki/Reflection_(computer_science))>. Acessado em: 15 de Março de 2009.
- [5] GNU General Public License. Disponível em: <http://pt.wikipedia.org/wiki/GNU_General_Public_License>. Acessado em: 15 de Março de 2009.