



PIBIC/CNPq/UFCG-2012

UM ESTUDO FORMAL E PRÁTICO BASEADO EM ADAPTADORES ENTRE COLEÇÕES JAVA PARA IMPLEMENTAR REFATORAMENTOS

Fábio Fernando de Oliveira Silva¹, Adalberto Cajueiro de Farias²

RESUMO

Refinamento de dados é uma teoria para justificar transformações específicas aplicadas a software. Dependendo da abordagem utilizada, modificar a estrutura de dados de um software pode requerer desde simples mudanças sintáticas até complexas provas de teoremas. De fato, mudar uma estrutura de dados em um programa nada mais é do que fazer um refinamento de dados no mesmo. Numa visão mais prática, essas mudanças podem ser vistas como refatoramentos e trazem diversos benefícios para adaptabilidade de software, bem como ocasionam uma melhoria geral da estrutura do software de forma a reduzir o impacto e custo das mudanças futuras. Em particular, o refatoramento de programas via modificação/substituição de coleções Java permite adaptar o software a mudanças de versão de bibliotecas, usar uma estrutura de dados mais adequada ao contexto, aumentar eficiência do software através do uso de estruturas de dados mais eficientes ou até mesmo prover flexibilidade de mudanças das estruturas de dados subjacentes. A presente proposta tem por objetivo principal a investigação do uso de adaptadores entre coleções Java, desempenhando o papel de uma camada que evita o máximo possível modificações no programa original permitindo mudar sua estrutura de dados subjacente.

Palavras-chave: engenharia de software, Java, estruturas de dados, refatoramento, collections, refinamento de dados.

A FORMAL AND PRATICAL STUDY BASED ON ADAPTERS BETWEEN JAVA COLLECTIONS TO IMPLEMENT REFACTORINGS

ABSTRACT

Data refinement is a theory to justify specific transformations applied to software. Depending on the approach used, changing the data structure of a software may require simple modifications to complex syntactic proofs of theorems. In fact, changing a data structure in a program is nothing more than a refinement to the same data. On a more practical view, these changes can be seen as refactorings and bring many benefits to the adaptability of the software, as well as cause a general improvement of the software structure to reduce the cost and impact of future changes. In particular, refactoring programs via modification/replace Java collections can be adapted to changes of software version libraries, using a data structure most suitable to the context, to increase efficiency through the use of software data structures or even more efficient provide the same flexibility to change data structures overlying. This proposal is primarily engaged in investigation of the use of adapters being Java collections, performing the role of a layer that avoids as much as possible modifications in the original program allows changing your underlying data structure.

Keywords: software engineering, Java, data structures, refactoring, collections, data refinement.

¹Aluno do Curso de Ciência da Computação, Departamento de Sistemas e Computação, UFCG, Campina Grande, PB, e-mail: fabio.silva@ccc.ufcg.edu.br

²Ciência da Computação, Professor Doutor, Departamento de Sistemas e Computação, UFCG, Campina Grande, PB, e-mail: adalberto@computacao.ufcg.edu.br