OCL com um Conjunto de inteiros

Álisson Paixão, Márcio Góes

October 21, 2019

1 Enunciado

O nosso problema é modelar um Conjunto de inteiros que seja possível adicionar elementos, remover elementos e calcular a média dos elementos desse conjunto. Para isso vamos modelar a classe em UML(na seção 3), fazer uma modelagem OCL (na seção 4) e fazer o código-fonte em Java e que respeite as invariantes, pré-condições e pós-condições do modelo OCL utilizando JML(na seção 5).

2 Ferramentas

AS ferramentas utilizadas para se fazer tudo foram o Astah [3] para a construção da classe uml do código feito, o código foi programado e o JML testado com o Eclipse[6], e o OCL foi testado com o software USE[2].

3 UML

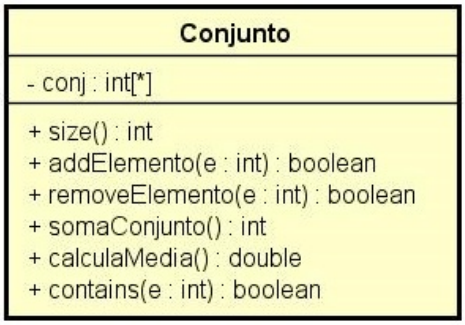

Figure 1: UML da classe conjunto [3]

4 OCL

```
model Conjuntos

class Conjunto
attributes
  conj : Collection(Integer);
operations
  calculaMedia() : Real
  addElemento(elem : Integer) : Boolean
  removeElemento(elem : Integer) : Boolean
end

constraints

-- Para adicionar um elemento no Conjunto, o elemento não pode
-- estar contido no conjunto e deve ser um inteiro
context Conjunto::addElemento(elem: Integer) : Boolean
pre: conj->count(elem) = 0
post: result = conj->includes(elem)

-- Para remover um elemento no Conjunto, o elemento tem que
-- estar contido no conjunto e deve ser um inteiro
context Conjunto::removeElemento(elem: Integer) : Boolean
pre: conj->includes(elem)
post: result = conj->count(elem) = 0

-- O tamanho do conjunto deve ser maior ou igual a 0
context Conjunto
inv tamanho: conj->size >= 0

-- Retorna a soma de todos os elementos dividido pelo número
-- de elementos, caso o número de elementos seja maior que 0
context Conjunto::calculaMedia() : Real
pre: conj->size > 0
post: result = conj->sum() / conj->size
```

5 Código-fonte e JML

```
1  public class Conjunto{
2          public ArrayList<Integer> conj;
3      public Conjunto(){
4          conj = new ArrayList<Integer>();
5      }
6      //@invariant size()>=0;
7      public int size(){
8          return conj.size();
9      }
10     //@requires !conj.contains(e);
11     //@ensures conj.contains(e);
12     public boolean addElemento(int e){
13         if(!contains(e)){
14             conj.add(e);
15             return true;
16         }
17             return false;
18     }
19     //@requires conj.contains(e);
20     //@ensures !conj.contains(e);
21     public boolean removeElemento(int e){
22         if(contains(e)){
23             for(int i = 0; i<size(); i++){
24                 if(e == conj.get(i)){
25                         conj.remove(i);
26                         return true;
27                 }
28             }
29         }
30         return false;
31     }
32     public int somaConjunto(){
33         int res = 0;
34         for(int i = 0; i<size(); i++){
35             res += conj.get(i);
36         }
37         return res;
38     }
39     //@requires size()>0;
40     //@ensures \result == (\sum int i; 0 <= i && i < size(); conj.get(i))/size();
41     public double calculaMedia(){
42         return somaConjunto()*1.0/size();
43     }
44     public  boolean contains(int e){
45         for(int i = 0; i<size(); i++){
46             if(e == conj.get(i)) return true;
47         }
48         return false;
49     }
50 }
```

6 Proposta de melhoria

- Utilizar uma estrutura de dados mais adequada para guardar o conjunto (ex: Set)
- Especificar uma restrição que um elemento não pode se repetir no conjunto

7 Repositório Git-Hub

- https://github.com/GoesMarcio/ModelagemPrecisa_ESOM

References

[1] University of Bremen Database Systems Group. *USE/OCL: Quick Tour*. URL: http://useocl.sourceforge.net/w/index.php/Quick_Tou.

[2] University of Bremen. *USE: UML-based Specification Environment*. Version 5.1.0. URL: http://useocl.sourceforge.net/w/index.php/Main_Page.

[3] Inc ChangeVision. *Astah Community*. Version 7.2.0. URL: http://astah.net/.

[4] Steve Cook et al. *The Amsterdam manifesto on OCL*. Springer, 2002, pp. 115–149.

[5] Joe Kiniry David Cok and Erik Poll. *Introduction to JML*. URL: https://www.cs.ru.nl/E.Poll/talks/jml_tutorial/1_intro_jml4up.pdf.

[6] Inc Eclipse Foundation. *Eclipse IDE*. Version 2018-09 (4.9.0). URL: https://www.eclipse.org/.

[7] *JML Reference Manual*. URL: http://www.eecs.ucf.edu/~leavens/JML/jmlrefman/jmlrefman_toc.html.

[8] *OCL – The Object Constraint Language in UML*. URL: http://home.agh.edu.pl/~regulski/psk/cw/wykladocl.pdf.

[9] Erik Poll. *The Java Modeling language JML*. URL: hhttps://www.cs.ru.nl/E.Poll/talks/Charter_JML.pdf.

[10] *The Java Modeling Language (JML)*. URL: http://www.eecs.ucf.edu/~leavens/JML/index.shtml.

[11] Database Systems Group Bremen University. *USE A UML based Specification Environment*. Preliminary Version 0.1. 2007. URL: http://www.db.informatik.uni-bremen.de/projects/use/use-documentation.pdf.

[12] Jos B Warmer and Anneke G Kleppe. *The object constraint language: getting your models ready for MDA*. Addison-Wesley Professional, 2003.

[13] Jos Warmer and Anneke Kleppe. *OCL: The constraint language of the UML*. SIGS PUBLICATIONS INC 71 WEST 23RD ST, 3RD FLOOR, NEW YORK, NY 10010 USA, 1999.