

OCL com um Conjunto de inteiros

Álison Paixão e Márcio Góes

October 21, 2019

- Enunciado
- UML
- OCL
- Código-fonte e JML
- Proposta de melhoria
- Git-Hub
- Referências

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, fazer uma modelagem OCL e fazer o código-fonte em Java e que respeite as invariantes, pré-condições e pós-condições do modelo OCL utilizando JML.

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].

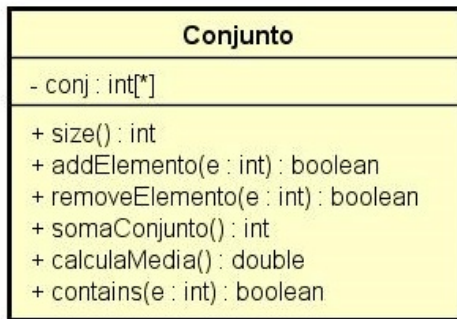


Figure: UML da classe conjunto [3]

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
```

```
public class Conjunto{
    public ArrayList<Integer> conj;
    public Conjunto(){
        conj = new ArrayList<Integer>();
    }

    //@invariant size()>=0;
    public int size(){
        return conj.size();
    }

    //@requires !conj.contains(e);
    //@ensures conj.contains(e);
    public boolean addElemento(int e){
        if(!contains(e)){
            conj.add(e);
            return true;
        }

        return false;
    }
}
```

```
//@requires conj.contains(e);
//@ensures !conj.contains(e);
public boolean removeElemento(int e){
    if(contains(e)){
        for(int i = 0; i<size(); i++){
            if(e == conj.get(i)){
                conj.remove(i);
                return true;
            }
        }
        return false;
    }
}

public int somaConjunto(){
    int res = 0;
    for(int i = 0; i<size(); i++){
        res += conj.get(i);
    }

    return res;
}
```

```
//@requires size()>0;
//@ensures \result == (\sum int i; 0 <= i && i < size();
//                               conj.get(i))/size();
public double calculaMedia(){
    return somaConjunto()*1.0/size();
}

public boolean contains(int e){
    for(int i = 0; i<size(); i++){
        if(e == conj.get(i)) return true;
    }
    return false;
}
}
```

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

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



Inc ChangeVision. *Astah Community*. Version 7.2.0. URL: <http://astah.net/>.



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>.



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



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.



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



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.



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



OCL – The Object Constraint Language in UML. URL: <http://home.agh.edu.pl/~regulski/psk/cw/wykladocl.pdf>.



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



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



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



The Java Modeling Language (JML). URL:
<http://www.eecs.ucf.edu/~leavens/JML/index.shtml>.

The End