

Modelagem precisa para Baskara

Conrado Boeira, Enrique Dutra

23/10/2019

Sumário

- 1 Introdução
- 2 Modelos
- 3 Invariantes
- 4 Condições
- 5 Conclusão
- 6 Referências

Introdução

- Fórmula de Báskara : Fórmula utilizada para resolver equações de 2º grau.

$$-b \pm \frac{\sqrt{b^2 - 4ac}}{2a}$$

- Modelagem OCL : É uma linguagem para especificar restrições aplicadas a modelos UML.
É precisa, textual e formal.

Diagrama de classes UML

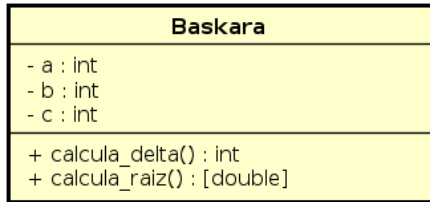


FIGURE – Diagrama de classes UML.

Invariantes

- $a \neq 0$
- $\Delta \geq 0$

Invariantes em código

```
class Baskara:  
    def __init__(self, a,b,c):  
        self.a = a  
        self.b = b  
        self.c = c  
  
        assert not a ==0, '0 valor de a deve ser diferente de 0'  
        assert self.calcula_delta() >= 0
```

FIGURE – Invariantes na linguagem Python.

```
context Baskara  
inv aDiferenteDeZero: not (self.a = 0)  
  
context Baskara  
inv deltaNaoNegativo: self.b*self.b - 4*self.a*self.c >= 0
```

FIGURE – Invariantes na linguagem OCL.

Pós-condições

- O tamanho da lista retornada pela função `calcula_raiz` é igual a 2.
- Substituindo as raízes encontradas na função `calcula_raiz` na equação $ax^2 + bx + c$ o resultado é 0

Invariantes em código

```
def calcula_raiz(self):  
    a = self.a  
    b = self.b  
    c = self.c  
    delta = self.calcula_delta()  
  
    resp1 = (-b + math.sqrt(delta)) / (2*a)  
    resp2 = (-b - math.sqrt(delta)) / (2*a)  
  
    resp = [resp1, resp2]  
  
    assert len(resp) == 2  
    assert (a*(resp[0]**2) + b*(resp[0]) + c) == 0  
    assert (a*(resp[1]**2) + b*(resp[1]) + c) == 0  
  
    return resp
```

FIGURE – Invariantes na linguagem Python.

Invariantes em código

```
context Baskara::calcula_raiz()
post raizesNaoNulas: (not (r1 = null)) and (not (r2 = null))

context Baskara::calcula_raiz()
post primeiraRaizZeraPolinomio: (self.a*(r1*r1) + self.b*r1 + self.c) = 0

context Baskara::calcula_raiz()
post segundaRaizZeraPolinomio: (self.a*(r2*r2) + self.b*r2 + self.c) = 0
```

FIGURE – Invariantes na linguagem Python.

Conclusão

Neste trabalho mostramos uma especificação da fórmula de Baskara usando OCL. Também testamos essa especificação com a ferramenta Use e desenvolvendo um código Python que aplica as asserções com o uso do comando assert.

Repositório Github

- https://github.com/conradoboeira/OCI_Baskara

Referências

- Use OCL (2015). Quick Tour [Online]. Disponível em :
http://useocl.sourceforge.net/w/index.php/Quick_Tour
(Acessado : 17 de Outubro 2019).
- Tutorialspoint (2019). Assertions in Python [Online].
Disponível em :
https://www.tutorialspoint.com/python/assertions_in_python.htm
(Acessado : 17 de Outubro 2019).