

Trabalhos Práticos - TP3



Arquitetura de Software - 2018/2

Alberane Lúcio Thiago da Cunha - alberane.cunha@posgrad.ufla.br
Willian Gaudêncio Rezende - willian.rezende@posgrad.ufla.br



Problema

- Analisar um código e propor uma possível refatoração
- Análise a ser implementada em um projeto Java
- Usando AST/Eclipse

A proposta de Solução..

The background is a solid teal color. It features several faint, semi-transparent geometric shapes. In the upper right, there is a large pie chart with three segments. To its right and below are several smaller pie charts. In the bottom right corner, there is a bar chart with four vertical bars of increasing height.

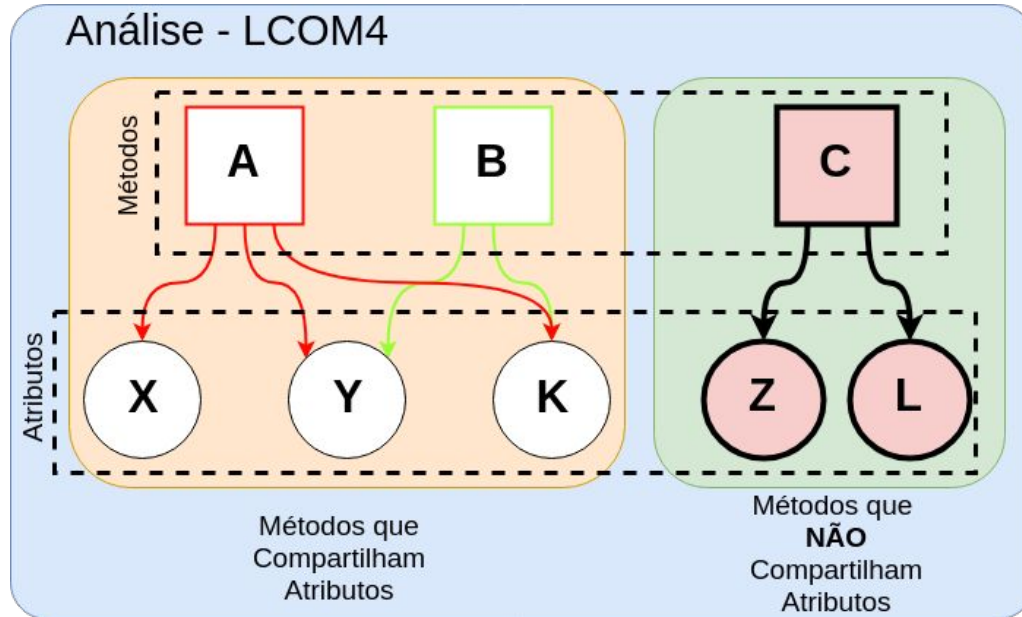


LCOM - Lacks Of Cohesion

- Análise de Refatoração baseada na **LCOM4** (Chidamber & Kemerer - [1993](#))
- **LCOM4** - Mede o **número** de componentes conectados
- (LCOM4 = 1)
 - indica uma classe coesa.
- LCOM4 ≥ 2
 - indica um problema. A classe deve ser dividida em classes menores.
- LCOM4 = 0
 - **Zero** acontece quando não há métodos em uma classe.

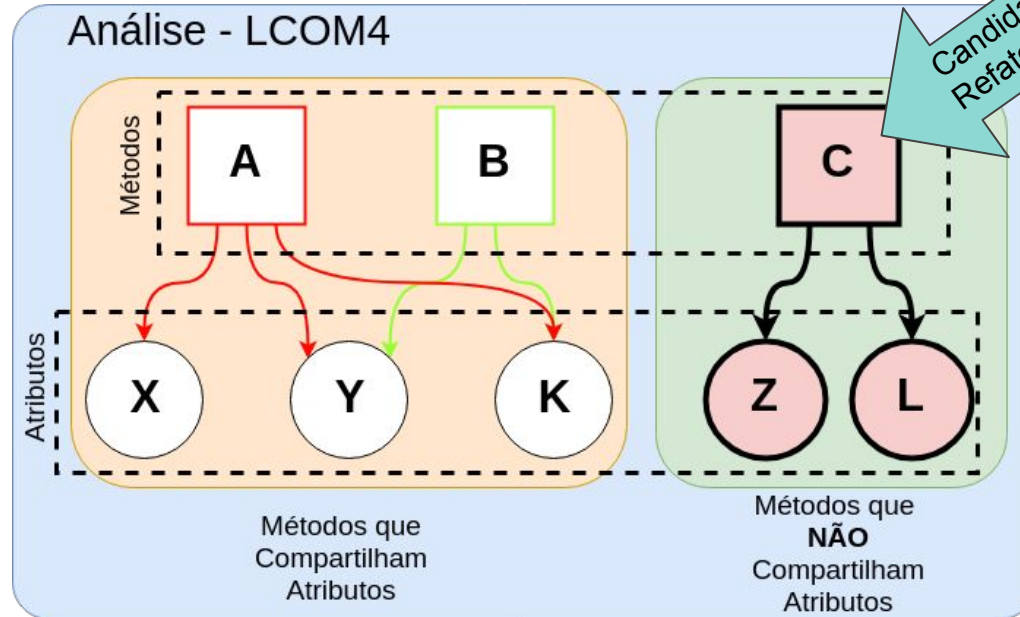
Identificando os “Desconectados”

- Métodos que **não** compartilham atributos podem ser **candidatos** à refatoração fazendo um **ExtracClass**.



Identificando os “Desconectados”

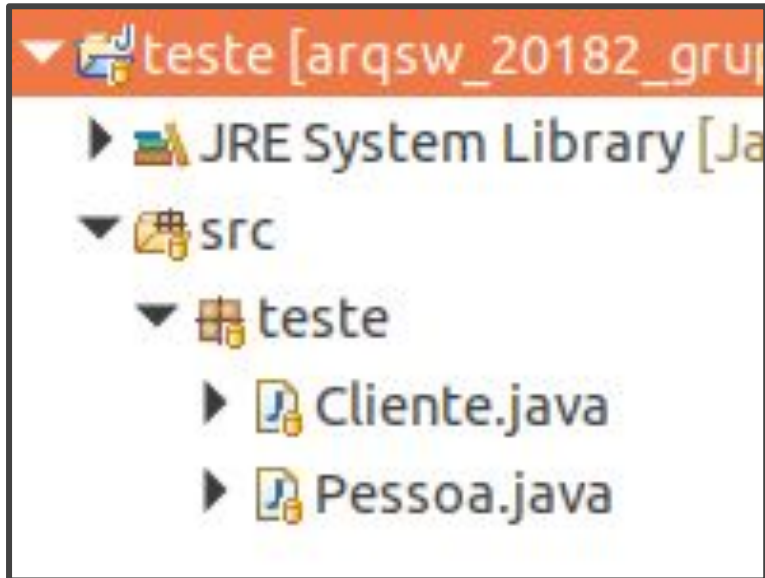
- Métodos que **não** compartilham atributos podem ser **candidatos** à refatoração fazendo um **ExtracClass**.



#ShowMeTheCode



O Projeto em Análise



Classes a serem Possivelmente Refatoradas

```
1 package teste;
2
3 public class Pessoa {
4
5     private String id;
6     private String cpf;
7     private String nome;
8     private String rua;
9     private String numero;
10    private String cidade;
11    private String bairro;
12    private String estado;
13    private String pais;
14    private String estado2;
15    private String pais2;
16
17    public void addPessoa() {
18        System.out.println(this.id + this.nome + this.cpf);
19        System.out.println(this.rua + this.numero + this.bairro + this.cidade);
20    }
21
22    public void addEndereco() {
23        System.out.println(this.rua + this.numero + this.bairro + this.cidade);
24    }
25
26    public void addEstadoPais() {
27        System.out.println(this.estado + this.pais);
28    }
29
30    public void addEstadoPais2() {
31        System.out.println(this.estado2 + this.pais2);
32    }
33 }
34
```

```
1 package teste;
2
3 public class Cliente {
4
5     private String id;
6     private String cpf;
7     private String nome;
8
9     private String rua;
10    private String numero;
11    private String cidade;
12    private String bairro;
13
14    private String infoCli;
15    private String tipoCli;
16
17    public void addCliente() {
18        System.out.println(this.id + this.nome + this.cpf + this.cidade);
19    }
20
21    public void addEnderecoCliente() {
22        System.out.println(this.rua + this.numero + this.bairro + this.cidade);
23    }
24
25    public void addContaCliente() {
26        System.out.println(this.tipoCli + this.infoCli + this.cidade);
27    }
28 }
29
```



```

1 package teste;
2
3 public class Pessoa {
4
5     private String id;
6     private String cpf;
7     private String nome;
8     private String rua;
9     private String numero;
10    private String cidade;
11    private String bairro;
12    private String estado;
13    private String pais;
14    private String estado2;
15    private String pais2;
16
17    public void addPessoa() {
18        System.out.println(this.id + this.nome + this.cpf);
19        System.out.println(this.rua + this.numero + this.bairro + this.cidade);
20    }
21
22    public void addEndereco() {
23        System.out.println(this.rua + this.numero + this.bairro + this.cidade);
24    }
25
26    public void addEstadoPais() {
27        System.out.println(this.estado + this.pais);
28    }
29
30    public void addEstadoPais2() {
31        System.out.println(this.estado2 + this.pais2);
32    }
33 }
34

```

Pessoa.java

Os Métodos

-- addEstadoPais
-- addEstadoPais2

Estão isolados e poderiam ir para uma classe fora de **Pessoa**

Código da Classe em:
<https://pastebin.com/A1760Med>

Cliente.java

A classe está coesa.

Os todos métodos compartilham ao menos um atributo

```
1 package teste;
2
3 public class Cliente {
4
5     private String id;
6     private String cpf;
7     private String nome;
8
9     private String rua;
10    private String numero;
11    private String cidade;
12    private String bairro;
13
14    private String infoCli;
15    private String tipoCli;
16
17    public void addCliente() {
18        System.out.println(this.id + this.nome + this.cpf + this.cidade);
19    }
20
21    public void addEnderecoCliente() {
22        System.out.println(this.rua + this.numero + this.bairro + this.cidade);
23    }
24
25    public void addContaCliente() {
26        System.out.println(this.tipoCli + this.infoCli + this.cidade);
27    }
28 }
29
```

Código da Classe em:
<https://pastebin.com/WVCB7e1H>



Solução

- Solução usando o Plugin do Eclipse - AST
- Executa as análise em todas as classes do projeto e fornece as seguinte saída

```
=====
Sugestão para análise de refatoração do Projeto
-----
```

```
Analisando Pacote/Classe: teste.Pessoa
```

```
Os Seguintes métodos possuem atributos exclusivos:
```

```
-> [addEstadoPais2, addEstadoPais]
```

```
Avalie fazer um ExtracClasss nestes métodos!
```

```
=====
Sugestão para análise de refatoração do Projeto
-----
```

```
Analisando Pacote/Classe: teste.Cliente
```

```
A classe está coesa.
-----
```



Obrigado....