

## Trabalho Prático 2 - Refactoring

### Refatoração do parser de resultados de análises de confiabilidade.

#### 1) As operações de refatoração:

Esse trabalho consiste em refatorar o parser de resultados de análise de confiabilidade desenvolvido para o trabalho prático 1. Com base nos resultados das execuções da suíte de testes desenvolvida junto com o parser, as operações de refatoração listadas em seguida deverão ser aplicadas no projeto e, obviamente, os testes deverão continuar passando. Para a execução da suíte de testes os mesmos arquivos contendo resultados de análise de confiabilidade (**analysisTime.out** e **totalTime.out**) deverão ser utilizados.

As operações de refatoração a serem aplicadas no projeto são **Extrair Método**, **Substituir método por objeto-método** e **Extrair Classe**. A aplicação de cada uma dessas operações está explicada a seguir.

A operação **Extrair Método** deverá ser aplicada nos trechos de código em que (i) é realizada a alocação dos objetos que fazem a leitura dos arquivos com os resultados das análises e (ii) nos trechos de código em que o resultado da transformação dos dados (em sequências em linhas ou colunas) é escrito de volta em arquivo. Como resultado da refatoração, os métodos extraídos deverão conter apenas as instruções necessárias para abrir os arquivos de entrada e saída do parser.

A operação **Substituir método por objeto-método** deverá ser aplicada no método que transforma os valores lidos dos arquivos em sequências de dados (em linhas ou colunas). Todo o método que realiza a transformação deverá ser movido para um objeto-método, e o método de origem deverá se tornar apenas uma indireção para o novo método. Os elementos que não são parte do método (i.e. não foram declarados no escopo do método de origem) deverão permanecer na classe de origem.

A operação **Extrair Classe** deverá ser aplicada nos métodos de leitura dos arquivos de entrada e escrita de arquivo de saída, de modo a movê-los para uma nova classe chamada **Persistência**. Os elementos da classe de origem que são utilizados apenas por esses métodos de leitura/escrita de arquivos deverão ser movidos para a nova classe.

#### 2) A execução / apresentação das refatorações

As refatorações deverão ser realizadas isoladamente uma das outras, ou seja, não deverão ser feitas duas ou mais refatorações ao mesmo tempo. A cada refatoração realizada deverá ser realizado um *commit* no repositório cuja mensagem deverá ter o seguinte formato:

```
[Refact] Nome da operação de refatoração
Origem: Nome da classe em que foi iniciada a refatoração
Impactos: Nome das classes e métodos que foram afetados
(criados, apagados ou alterados) pela refatoração
```

Explicação: texto explicativo da operação de refatoração, apresentando claramente o que foi realizado na classe e os resultados obtidos.

#### Exemplo de mensagem:

```
[Refact] Substituir Type Code por subclasses
Origem: CalculoOcupacaoDasFaixas.java
Impactos:      Calculo1aFaixa.java,      Calculo2aFaixa.java,
Calculo3aFaixa.java, Calculo4aFaixa.java, Calculo5aFaixa.java
Explicação: a refatoração foi aplicada no método computar()
da classe CalculoOcupacaoDasFaixas. Os valores (1..5) do switch
case são type codes para as faixas de calculo do IRPF. Eles foram
substituídos pelos retornos do método polimórfico getFaixa(),
sobrescritos nas respectivas classes CalculoXaFaixa. Foi
necessária ainda a criação do método fábrica create na classe
CalculoOcupacaoDasFaixas.
```

### 3) Avaliação das refatorações

As refatorações serão avaliadas tanto pelos resultados alcançados ao final da refatoração quanto pela execução dos passos da refatoração. Como resultados das refatorações serão considerados (i) se o teste unitário correspondente continua passando (40% da nota) e (ii) os atributos e métodos resultantes da operação (30% da nota). Os passos da refatoração serão avaliados quanto às evidências de sua execução. Caso não seja encontrada evidência da execução do passo da refatoração serão descontados 10% da nota para cada passo não executado, limitados a 30% de penalidade. Para garantir a correta aplicação dos passos de refatoração recomenda-se fortemente que **atenham-se apenas às transformações indicadas em cada passo da refatoração**, ou seja, não execute nada além do que foi solicitado pelo passo de refatoração.

### 4) Desenvolvimento / entrega do trabalho

O trabalho deverá ser realizado pelos mesmos grupos formados para o trabalho prático 1 (até três componentes). Os nomes e matrículas dos componentes deverão ser apresentados no arquivo README.md do repositório. A avaliação da participação dos membros no trabalho será realizada pela análise de autoria dos commits.

Cada grupo deverá enviar email para o professor indicando o nome dos componentes do grupo e o endereço do repositório Github do trabalho.

Valor do trabalho: 60% da nota final (10% extrair método, 30% substituir método por objeto método, 20% extrair classe).

Data de entrega: 01 de novembro de 2021.

Forma de entrega: via repositório Github.