



Entrega 3 - Projeto de Código

Alunos(as):

Leonardo Fernandes Padre - 200067036

Henrique Sandoval Camargo Hida - 180113569

Lucas Soares Barros - 202017700

Caio Felipe Alves Braga - 211030694

1. Princípios de bom projeto de código e relação com os maus-cheiros de código

a) Simplicidade

- **Definição:** O código deve ser simples e direto, evitando complexidades desnecessárias. Isso facilita a leitura, a manutenção e a evolução do software.
- **Relação com maus-cheiros:**
 - **Método Longo (Long Method):** Difícil de compreender e manter.
Refatoração sugerida: **Extrair Método (Extract Method)**.
 - **Código Duplicado (Duplicated Code):** Aumenta a complexidade.
Refatoração sugerida: **Extrair Método (Extract Method)** ou **Substituir Algoritmo (Substitute Algorithm)**.

b) Elegância

- **Definição:** O código deve ser bem estruturado e esteticamente agradável.
- **Relação com maus-cheiros:**
 - **Nomes não descritivos (Poor Naming):** Torna o código menos claro.
Refatoração sugerida: **Renomear (Rename)**.

c) Modularidade

- **Definição:** O código deve ser dividido em módulos **coesos e reutilizáveis**.

- **Relação com maus-cheiros:**
 - **Classe Grande (Large Class):** Viola a modularidade. Refatoração sugerida: **Extrair Classe (Extract Class)**.
 - **Acoplamento excessivo (Feature Envy):** Métodos que acessam demasiadamente outras classes. Refatoração sugerida: **Mover Método (Move Method)**.

d) Boas Interfaces

- **Definição:** Interfaces devem ser claras, consistentes e fáceis de usar.
- **Relação com maus-cheiros:**
 - **Interface grande (Large Interface):** Dificulta o uso e a manutenção. Refatoração sugerida: **Extrair Interface (Extract Interface)**.

e) Extensibilidade

- **Definição:** Facilita a adição de novas funcionalidades sem grandes alterações.
- **Relação com maus-cheiros:**
 - **Código rígido (Rigidity):** Difícil de modificar. Refatoração sugerida: **Extrair Método (Extract Method)** ou **Extrair Classe (Extract Class)**.

f) Evitar Duplicidade

- **Definição:** Promove reutilização e consistência.
- **Relação com maus-cheiros:**
 - **Código Duplicado (Duplicated Code):** Refatoração sugerida: **Extrair Método (Extract Method)** ou **Substituir Algoritmo (Substitute Algorithm)**.

g) Portabilidade

- **Definição:** Deve funcionar em diferentes plataformas.
- **Relação com maus-cheiros:**
 - **Dependência de plataforma (Platform Dependency):** Refatoração sugerida: **Introduzir Camada de Abstração (Introduce Abstraction Layer)**.

h) Código Idiomático e Bem Documentado

- **Definição:** Segue padrões e boas práticas da linguagem.
- **Relação com maus-cheiros:**
 - **Código obscuro (Obscure Code):** Difícil de entender. Refatoração sugerida: **Renomear (Rename)** e adicionar documentação.

2. Análise do Trabalho Prático 2

a) Código Duplicado (Duplicated Code)

- Onde ocorre: Nos métodos `getDeducao()` e `getTotalOutrasDeducoes()`.
- Princípio violado: Evitar duplicidade.
- Refatoração: Extrair Método (Extract Method).

b) Método Longo (Long Method)

- Onde ocorre: No método `calcularImposto()`.
- Princípio violado: Simplicidade.
- Refatoração: Extrair Método (Extract Method).

c) Classe Grande (Large Class)

- Onde ocorre: Na classe `IRPF`.
- Princípio violado: Modularidade.
- Refatoração: Extrair Classe (Extract Class).

d) Nomes Não Descritivos (Poor Naming)

- Onde ocorre: Nos métodos `getOutrasDeducoes()` e `getDeducao()`.
- Princípio violado: Código idiomático.
- Refatoração: Renomear (Rename).

e) Acoplamento Excessivo (Feature Envy)

- Onde ocorre: No método `cadastrarPensaoAlimenticia()`.
- Princípio violado: Modularidade.
- Refatoração: Mover Método (Move Method).

f) Uso Inadequado de Streams

- Onde ocorre: Em `getDeducao()` e `getDependente()`.
- Princípio violado: Simplicidade.
- Refatoração: Substituir Stream por Loop (Replace Stream with Loop).

Referências Bibliográficas

- FOWLER, Martin. **Refactoring: Improving the Design of Existing Code**. Addison-Wesley Professional, 1999.
- GOODLIFFE, Pete. **Code Craft: The Practice of Writing Excellent Code**. No Starch Press, 2006.