06 - Objects and Data Structure

Importante distinguir o que deve ser alterado e o que não, dentro de uma classe

ABSTRAÇÃO DE DADOS

- Distinguir o que e quando deve ser atribuido.
- Quem tem acesso a qual atributo.
- Podemos forçar que atributos sejam definidos simultaneamente com um setter.
- Concreto X Abstrato
    - Ideal é esconder implementação.
    - Requer cuidado e atenção para fazer isso. Não é só ter getters e setters.

ESTRUTURA DE DADOS X OBJETOS

- Estrutura de Dados e Objeto são opostos:
- Objeto esconde dados atrás de abstrações para exprimir uma funcionalidade
- Estrutura de Dados expõe os dados sem ter nenhuma funcionalidade explícita.
- Dicotomia:
    - Código procedural (código que usa estrutura de dados):
        - Facilita adicionar novas funções sem alterar estrutura de dados
        - Dificulta adicionar novas estrutura de dados pois requer alteração nas funções
    - Orientação a objeto:
        - Facilita adicionar novas classes/estrutura de dados sem alterar funções.
        - Dificulta adicionar novas funções porque as classes devem ser alteradas.
    - Escolha entre os dois depende da situação. Trade-off.

LEI DE DEMETER

- Módulo deve ser ignorante sobre os detalhes do objeto que ele manipula.
- "Não conversar com estranhos".
- Um método de uma classe C só deve chamar os métodos de:
    - A classe C em si
    - Objeto criado pelo método
    - Objeto argumento do método
    - Objeto em um atributo da classe C
- Exemplo de violação da regra:

```
final String outputDir = ctxt.getOptions().getScratchDir().getAbsolutePath();

//Esse encadeamento de funções é ruim
//Melhor separar em várias linhas, como em:

Options opts = ctxt.getOptions();
File scratchDir = opts.getScratchDir();
final String outputDir = scratchDir.getAbsolutePath();
```

- Mesmo assim pode ser uma violação, dependendo da estrutura.
    - Ctxt, Options e ScratchDir são objetos ou estruturas de dados?
    - Se forem Estruturas de dados tudo bem expor sua estrutura interna. Se objeto, não
    - Dot notation seria melhor do que accessors aqui para sinalizar que é esturutra de dado.

HÍBRIDOS

- Objetos que são tanto objeto quanto estrutura de dados.
- Têm funcionalidade significativa e expõe os dados
- Pior dos dois mundos, devem ser evitados.

ESCONDER ESTRUTURA

- Pode requerer novos métodos para esconder dados internos, ser uma função.

DATA TRANSFER OBJECT (DTO)

- Estrutura de dados mais pura
- Apenas variáveis públicas, zero funcionalidade.
- Bem útil e frequente.

ACTIVE RECORD

- Tipo de DTO
- Tem dados públicos mas alguns métodos de navegação como save e find.
- Não deve conter regras de negócio.
    - Estas devem existir em outro objeto.