Refatoramento de Código

Exemplo extraído do livro *Refactoring: Improving the Design of Existing Code* (Fowler, 1999)

Parte 2: Refatorando o método extrato()

I. A Refatoração do Sistema de Locadora

A refatoração é um processo disciplinado de modificar um sistema (reestruturar o código existente) para melhorar a estrutura interna do código sem alterar seu comportamento externo. Uma refatoração não adiciona nem remove funcionalidades.

Refatoramento sempre deve ser feito apoiando-se em **Testes de Unidade** para assegurar-se de que as transformações no código não quebrem o código que funciona. Entretanto, não serão discutidas questões sobre os testes de unidade para não estender ainda mais a dinâmica destra prática.

Vamos considerar que os testes automáticos existam. Então, atacaremos o primeiro problema: **o método extrato() é muito grande e "faz tudo sozinho".** Sendo assim, vamos decompor este método em pedaços menores.

II. Tarefa 1 *(refactoring 1.1)*

Identifique o(s) bloco(s) de código com **alguma coesão** no método **extrato()** e mova-o(s) para novo(s) método(s) na classe **`Cliente`**. Lembrando: uma classe/método com **baixa coesão** assume responsabilidades que pertencem a outras classes/métodos, logo deveriam ser delegadas a outras. Depois de efetuar as mudanças, compile e rode o código novamente. A saída deve ser a mesma e a classe **`Locadora`** não deve ser afetada pelas mudanças.

III. Tarefa 2 *(refactoring 1.2)*

Você deve ter percebido que o cálculo do valor de um aluguel é coeso o bastante para virar um novo método. Com o intuito de trabalhar em cada pedaço/método individualmente, vamos fazer uma pequena mudança no método **`valorDeUmAluguel()`**: revisar o nome das variáveis/argumentos e alterar para um nome mais sugestivo, pois o código **deve comunicar bem o seu propósito para outros programadores.**

> *"Qualquer pessoa pode escrever código que um computador entende. Bons programadores escrevem código que um **ser humano pode entender**."*