Capitulo 9 resumo:

Refatoração e Leis de Lehman

Refatoração consiste em modificar um programa para torná-lo mais compreensível e fácil de alterar, sem afetar seu comportamento. Segundo as Leis de Lehman, sistemas de software tendem a se degradar com o tempo devido ao aumento da complexidade interna, mas a refatoração pode mitigar esse efeito, preservando a qualidade interna. Exemplos incluem a extração de código para um novo método, melhorando modularidade, design e testabilidade. A técnica Extract Method é usada para reutilização de código, remoção de duplicações e melhor compreensão.

Técnicas de Refatoração: Inline Method, Move Method e Extract Class

Inline Method é o oposto de Extract Method, incorporando o corpo de um método diretamente nos pontos onde é chamado. Move Method transfere um método para uma classe mais adequada, melhorando coesão e reduzindo acoplamento, com variações como Pull Up Method e Push Down Method. Extract Class extrai uma nova classe de uma existente com múltiplas responsabilidades, organizando melhor o código e reduzindo sua complexidade.

Refatoração e Design de Código

Refatoração melhora o design e a estrutura do código por meio de técnicas como renomeação, extração de variáveis e remoção de flags, tornando-o mais legível e fácil de manter. Para ser eficaz, depende de bons testes, especialmente unitários, e pode ser feita de forma oportunista ou estratégica. A maioria das IDEs suporta refatorações automatizadas, aplicando-as após verificar pré-condições.

Code Smells e Soluções de Refatoração

Code Smells são sinais de código de baixa qualidade que indicam necessidade de refatoração, como código duplicado, métodos longos e classes grandes. Exemplos incluem Feature Envy, listas extensas de parâmetros e variáveis globais, com soluções como extração de classe e movimentação de métodos. Técnicas como Extract Method e Extract Class ajudam a eliminar esses problemas, tornando o código mais compreensível e fácil de modificar.

Objetos Imutáveis e Refatoração

Objetos imutáveis, como a classe String do Java, são recomendados sempre que possível, pois são seguros para compartilhamento e execução em múltiplas threads. Objetos simples, como datas e endereços, devem ser imutáveis para reduzir a presença de objetos mutáveis

no sistema. Code smells relacionados a objetos mutáveis e comentários desnecessários devem ser eliminados para melhorar a legibilidade e reduzir a dívida técnica. IDEs facilitam esse processo com ferramentas automatizadas de refatoração.