

# Refatoração

A refatoração é uma prática que pode parecer simples, mas, para ser feita de forma adequada, ela pode seguir alguns pilares:

1. **Abstração:** A abstração pode ser composta por dois métodos: o **pull-up** e o **push down**.
  - **Pull-up:** consiste em extrair o código de uma subclasse para uma superclasse. Pode ser realizado quando há uma sobrescrita desnecessária por parte das subclasses ou quando deseja-se realizar a abstração de uma determinada classe.
  - **Push-down:** consiste em levar o código da superclasse para a subclasse. Geralmente é um método utilizado por meio da sobrescrita de métodos, mas não somente por isso. É, também, possível trazer um método ou atributo para a especialização, quando necessário.
2. **Composição:** Pode ser feita com a abordagem de **extração** ou com a abordagem **inline**. Ambos apontam para a modularização do código, a qual visa dividir grandes partes de código em blocos menores.
  - **Extração:** Consiste em retirar partes do código existente (um método, por exemplo) e as transfere para um novo método.
  - **Inline:** Basicamente, é o comentário da extração. Consiste em pegar a chamada de um método, apagá-la, apagar o próprio método que está presente em outra classe e reescrevê-lo onde ele será utilizado. É uma técnica utilizada quando certos métodos são utilizados em lugares muito específicos ou em apenas uma classe.
3. **Recursos de movimentação:** Consiste em movimentar atributos e/ou métodos entre classes para aumentar a coesão do código, bem como reduzir dependências. É utilizado para facilitar a visualização do código.
4. **Refatoração vermelha-verde:** Também conhecido como ciclo red-green-refactor, é um ciclo de refatoração que tem, como base, o desenvolvimento orientado a testes.
  - **Fase vermelha:** Nesta fase, os desenvolvedores desenvolvem testes para as funcionalidades. Esses primeiros testes são feitos para falharem e, com o passar do ciclo, terem êxito com a implementação das funcionalidades.
  - **Fase verde:** Aqui, os desenvolvedores programam as funcionalidades que vão atender aos testes escritos na fase vermelha. Nessa fase, a velocidade é a prioridade, não a qualidade.

- **Fase de refatoração:** Aqui, o código é aprimorado. Torna-se mais limpo, eficiente e claro. A qualidade fica acima da velocidade, mas sem alterar o funcionamento do programa, é claro.
5. **Simplificação:** Aqui, como o próprio pilar já diz, o código é simplificado tendo nomes de métodos e classes simplificados, parâmetros de métodos reduzidos, entre outras otimizações de simplificação de código.