

Universidade Tecnológica Federal do Paraná

Engenharia de Software - Curso de Arquitetura de Software (AS27S)

INSTRUTOR: Prof. Dr. Gustavo Santos

Aluno: Alex Domingues da Silva, RA: 1496727

CCH - Padrões de Projeto Comportamentais - Visitor

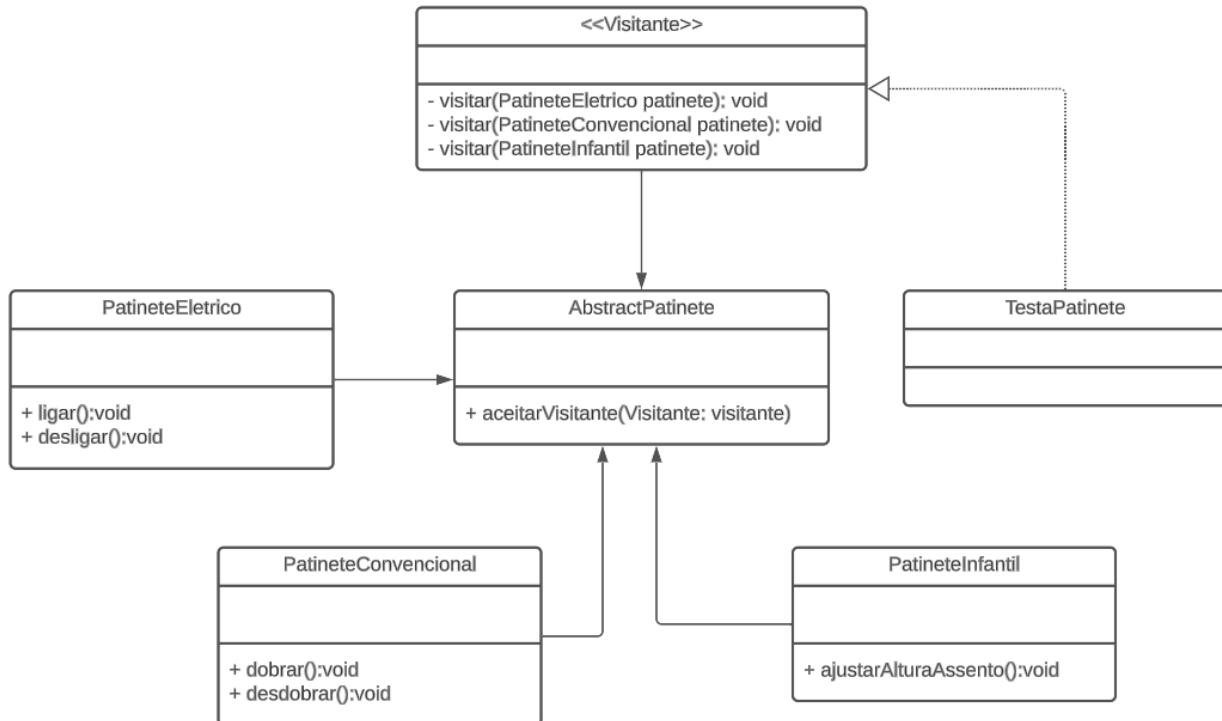
Problema

O código em questão resolve o problema de adicionar comportamentos específicos de verificação aos diferentes tipos de patinetes sem modificar suas classes. Caso contrário seria necessário adicionar diretamente esses comportamentos de verificação nas classes de patinete.

Descrição da Solução

A classe abstrata "AbstractPatinete" estabelece uma estrutura fundamental para a criação de patinetes, fornecendo o método "aceitarVisitante()", que recebe um objeto visitante como parâmetro. Isso permite a invocação do método "visitar()" nos patinetes. Na solução apresentada, a classe "TestaPatinete" implementa a interface `Visitante` e é recebida por cada um dos patinetes, permitindo que eles sejam testados por essa implementação.

Visão Geral



Exemplo de Código Java

Exemplo do código: [CCHS AS](#)

Consequências

- Vantagens
 - Permite estender a funcionalidade da estrutura sem alterar seu código existente, seguindo o princípio de aberto/fechado (Open/Closed Principle).
 - O Visitor separa as operações específicas dos elementos da estrutura de objeto em classes visitantes
 - Agrupamento de operações relacionadas: Permite agrupar operações relacionadas em uma única classe visitante

-
- Facilidade na adição de novas operações
 - Desvantagens
 - Potencial aumento da complexidade do código, especialmente em estruturas de objeto complexas.
 - Introdução de acoplamento adicional entre os elementos e o visitante.
 - Requer uma análise cuidadosa para determinar se a complexidade adicionada é justificada pelos benefícios obtidos.