Jogo de Batalha com Padrão Decorator

Projeto de Programação Orientada a Objetos

October 29, 2024

Introdução

Este projeto é uma implementação de um jogo de combate entre guerreiros utilizando o padrão de design **Decorator**. O objetivo é aplicar o conceito de composição de objetos para adicionar habilidades e melhorias aos personagens de maneira flexível e extensível, sem modificar a estrutura original das classes.

Estrutura do Projeto

O projeto está estruturado nas seguintes classes principais:

Classes Base e de Personagens

- **Personagem.cs**: Classe base abstrata que define atributos e métodos essenciais dos personagens.
- **Guerreiro.cs**: Classe concreta que representa o guerreiro padrão.
- **Arqueiro.cs e Ladino.cs**: Classes de personagens adicionais com características próprias.

Decoradores

- **PersonagemDecorator.cs**: Classe base para todos os decoradores.
- **EspadaFlamejante.cs**: Decorador que aumenta o ataque do guerreiro.
- **ArmaduraDeFerro.cs**: Decorador que aumenta a defesa do guerreiro.
- **AnelDecorator.cs**: Decorador que adiciona a habilidade de refletir dano ao guerreiro.
- **ArcoDivino.cs**: Decorador que incrementa atributos específicos do Arqueiro.
- **EsferaDePoder.cs**: Classe que representa esferas de poder coletadas pelo guerreiro para aumentar o poder reflexivo do anel.

Gerenciamento de Combate

- **Duelo.cs**: Classe que gerencia o combate em turnos entre dois personagens, aplicando as mecânicas de crítico, esquiva e dano reflexivo.
- **Program.cs**: Classe principal que configura os personagens e inicia o duelo.

Funcionalidades Implementadas

- **Combate em Turnos**: Os guerreiros alternam turnos de ataque com a possibilidade de crítico, esquiva e reflexo de dano.
- **Aplicação de Decoradores**: Permite modificar e adicionar habilidades a um personagem, como aumentar o ataque, defesa ou refletir dano.
- **Reflexo de Dano**: O anel permite refletir 10% do dano, incrementado em 10% para cada esfera de poder coletada até o limite de 100%.

Detalhes de Implementação

Cada classe foi implementada com base nos princípios de orientação a objetos e boas práticas de design, visando flexibilidade e extensibilidade. O padrão **Decorator** permite que novos atributos e funcionalidades sejam aplicados dinamicamente a instâncias de personagens sem modificar a estrutura original das classes.

Exemplo de Saída

A execução do programa exibe o combate entre dois personagens, incluindo detalhes como ataques, uso de crítico, esquivas e reflexo de dano com base nas habilidades adquiridas pelos personagens durante a batalha.

Conclusão

O projeto demonstra a eficácia do padrão **Decorator** ao permitir que novos comportamentos sejam aplicados aos personagens de forma modular e extensível. Este trabalho cumpre os requisitos de flexibilidade e uso avançado de Programação Orientada a Objetos, sendo uma excelente aplicação prática dos conceitos aprendidos.