

Sudoku em POO

Luiz Eduardo e Kaiky Vieira

O desenvolvimento deste trabalho teve como objetivo implementar um jogo de Sudoku aplicando os princípios da Programação Orientada a Objetos (POO). Para isso, o primeiro passo foi analisar as responsabilidades envolvidas no funcionamento do jogo e separá-las em componentes independentes. Essa separação permitiu definir classes específicas, cada uma encarregada de uma parte fundamental do sistema, promovendo organização, encapsulamento e facilidade de manutenção.

Inicialmente, foi criada a classe abstrata Jogo, que serve como modelo para jogos interativos. Ela estabelece que toda forma de jogo deve possuir um método responsável pela execução da partida. Essa definição garante uniformidade e possibilita que o jogo de Sudoku siga um padrão claro de funcionamento, além de facilitar possíveis expansões futuras.

Em seguida, estruturou-se a classe GeradorSudoku, responsável exclusivamente pela lógica de geração dos tabuleiros. Essa classe contém os métodos que criam um Sudoku completo utilizando um algoritmo de backtracking e outro método dedicado a remover números conforme a dificuldade selecionada. Ao isolar essa funcionalidade, garantiu-se que a lógica de geração não se misture com o funcionamento do jogo em si.

A classe Tabuleiro foi criada para representar e manipular o estado do jogo. Ela é responsável por armazenar a matriz do Sudoku, validar movimentos, exibir o tabuleiro ao jogador e verificar se a solução está completa e correta. A presença dessa classe permite que o tabuleiro seja tratado como um objeto independente, centralizando toda a lógica relacionada à jogabilidade.

Com esses elementos prontos, desenvolveu-se a classe SudokuGame, que herda de Jogo e funciona como o núcleo da interação com o jogador. Nela são tratados a escolha da dificuldade, o fluxo do jogo, a leitura das jogadas e a comunicação entre o jogador e o tabuleiro. Essa classe utiliza os serviços oferecidos por GeradorSudoku e Tabuleiro, demonstrando claramente o uso de composição, um princípio fundamental da POO.

Por fim, o arquivo principal (main) foi deixado responsável apenas por iniciar o jogo, seguindo o princípio da responsabilidade única. Assim, o

projeto foi organizado de forma modular, com cada classe realizando uma função clara e bem definida. O resultado final é um sistema estruturado, fácil de entender, ampliar e manter, e que utiliza de maneira adequada os fundamentos da programação orientada a objetos.