Sudoku com POO – Eduardo S. De Oliveira

O desenvolvimento do projeto Sudoku em Python envolveu a transformação de um código inicialmente procedural em um programa totalmente orientado a objetos, aplicando os princípios fundamentais da Programação Orientada a Objetos (POO). O principal objetivo foi organizar a lógica do jogo de forma modular, tornando o código mais legível, reutilizável e fácil de manter, ao mesmo tempo em que permitia a expansão do jogo para novas funcionalidades no futuro.

O primeiro passo consistiu em identificar as entidades principais do jogo, separando responsabilidades em classes distintas. Foi definida uma classe Board, responsável por toda a lógica do tabuleiro, incluindo a criação do tabuleiro, a ocultação de números, a validação das jogadas e a apresentação visual do tabuleiro ao usuário. Esta separação permitiu aplicar o encapsulamento, protegendo atributos internos e garantindo que o acesso aos dados críticos do tabuleiro ocorresse apenas por meio de métodos específicos.

Em seguida, foi criada a classe abstrata JogoBase, que estabeleceu os métodos obrigatórios para qualquer implementação concreta do jogo, funcionando como um contrato de abstração. Essa abordagem assegura que qualquer classe derivada implemente os métodos essenciais, como iniciar o jogo e gerenciar a interação com o jogador. A classe Jogo foi implementada como uma extensão de JogoBase, fornecendo a lógica concreta para iniciar o jogo, exibir o menu de dificuldade, capturar entradas do usuário e atualizar o tabuleiro conforme as jogadas.

Durante a transformação, foram aplicados conceitos de polimorfismo, permitindo que métodos como a validação de jogadas pudessem ter comportamentos distintos dependendo das regras do tabuleiro ou da dificuldade selecionada. Isso proporciona flexibilidade para criar futuras versões do jogo com diferentes modos ou restrições sem alterar a interface principal. A herança foi utilizada para estruturar a relação entre JogoBase e Jogo, garantindo reutilização de código e consistência na implementação das funcionalidades obrigatórias.

Além disso, o código foi organizado para fornecer feedback visual ao usuário, utilizando cores para indicar jogadas corretas e incorretas, enquanto o fluxo do jogo permanece claro e linear, desde a seleção de dificuldade até a conclusão do Sudoku ou a opção de sair do jogo. A separação de responsabilidades em classes distintas facilitou a manutenção, teste e expansão do sistema.

Em conclusão, a transformação do código para uma estrutura orientada a objetos resultou em um programa modular, seguro e flexível, demonstrando claramente a aplicação de abstração, encapsulamento, herança e polimorfismo. Esse processo não apenas melhorou a organização e a legibilidade do código, mas também estabeleceu uma base sólida para futuras melhorias, como a criação de novos tipos de tabuleiros ou modos de jogo adicionais.