

Relatório – Adaptação do Projeto de Sudoku para Programação Orientada a Objetos

Instituto Federal Catarinense – Campus Concórdia

Disciplina: Programação I

Prof.: Alisson Borges Zanetti

Tema: Sudoku – Projeto Orientado a Objetos

Dupla: Gabriela Nietiedt e Daniel Augusto Masson

1. Introdução

Este relatório apresenta o processo de adaptação do projeto do Sudoku, originalmente desenvolvido no segundo trimestre de maneira procedural, para o paradigma da **Programação Orientada a Objetos (POO)**.

O objetivo foi reorganizar, modularizar e encapsular a lógica do jogo utilizando classes, métodos, atributos e princípios fundamentais como **abstração, encapsulamento, modularização, herança e polimorfismo**.

2. Estrutura Geral do Projeto

O projeto final foi dividido em dois módulos principais:

a) Classe Tabuleiro (Tabuleiro.py)

Responsável por todas as funcionalidades técnicas do Sudoku:

- Geração de tabuleiro vazio;
- Preenchimento automático usando backtracking;
- Validação de jogadas;
- Geração de jogos com diferentes níveis de dificuldade.

Essa classe representa o *modelo lógico* do Sudoku, isolando a mecânica do jogo e evitando que o código do menu tenha que manipular diretamente estruturas internas.

b) Classe Menu (Menu.py)

Responsável pela interação com o usuário:

- Exibir o menu;
- Escolher dificuldade;
- Imprimir o tabuleiro;
- Receber entradas do jogador;
- Verificar acertos e erros.

O Menu utiliza um objeto Tabuleiro como dependência, sendo uma **composição**, algo muito importante de POO.

Utilizamo a ajuda do google Gemini para complementar a verificação dos dígitos que o usuário passa, onde foi utilizado o `.strip()` e o `.isdigit()`, cujos são métodos para remover espaços em branco e se são dígitos (de 0 a 9), respectivamente. A ajuda foi necessária para que o código ficasse mais compreensível e simples de escrever.

3. Encapsulamento e Organização Interna

A classe Tabuleiro utiliza métodos internos com **underscore** (ex: `_preencher`)

Isso indica que esses métodos são destinados apenas ao uso interno da própria classe, representando o princípio de **encapsulamento**.

Além disso:

- Somente métodos públicos são usados pelo menu;
- O estado interno do tabuleiro não é alterado diretamente pelo jogador;
- Toda verificação passa pelos métodos apropriados.

4. Abstração Utilizada

A classe Tabuleiro abstrai o “conceito” de um Sudoku, fornecendo operações de alto nível como:

- gerar_tabuleiro_completo()
- gerar_jogo(dificuldade)

O usuário (e a classe Menu) não precisa saber como o backtracking funciona; apenas usa os métodos prontos.

5. Polimorfismo e Herança

O projeto atual **não exige herança**, pois Sudoku não se divide naturalmente em subclasses. Entretanto, os métodos são organizados de forma que o uso de polimorfismo seria possível caso o projeto evoluísse (ex.: diferentes variações de Sudoku, modos de jogo etc.).

6. Conclusão

A adaptação para Orientação a Objetos tornou o projeto mais organizado, modular e fácil de manter.

A separação entre lógica interna do jogo (Tabuleiro) e interface textual (Menu) tornou o código mais limpo e alinhado às boas práticas estudadas durante o trimestre.

A reestruturação também permitiu entender melhor os conceitos de encapsulamento, composição e abstração, aplicando-os em um projeto real.