01 – Jogo da Velha

Existem alguns problemas no código que podem dificultar sua manutenção, como:

1. A classe Tabuleiro está realizando múltiplas responsabilidades, como armazenar o estado atual do tabuleiro, exibir o tabuleiro na tela e verificar o vencedor. Isso viola o princípio de responsabilidade única (SRP) do SOLID.
2. O código não está usando encapsulamento para proteger o estado do tabuleiro. Qualquer parte do código pode acessar e modificar a lista de spots, o que pode levar a bugs e comportamentos inesperados.

Como solução, podemos aplicar o princípio de responsabilidade única e separar as responsabilidades em classes diferentes. Podemos criar uma classe Tabuleiro para armazenar o estado do tabuleiro e outra classe TabuleiroView para exibir o tabuleiro na tela. Também podemos usar encapsulamento para proteger o estado do tabuleiro e apresentar métodos públicos para acessá-lo e modificá-lo.

Classe TabuleiroView

public class TabuleiroView {

    public static *void* display(*Tabuleiro* *tabuleiro*) {

*String* tabuleiroFormatado = tabuleiro.getSpots()[0] + " | " + tabuleiro.getSpots()[1] + " | " + tabuleiro.getSpots()[2] + "\n" +

                                    tabuleiro.getSpots()[3] + " | " + tabuleiro.getSpots()[4] + " | " + tabuleiro.getSpots()[5] + "\n" +

                                    tabuleiro.getSpots()[6] + " | " + tabuleiro.getSpots()[7] + " | " + tabuleiro.getSpots()[8];

        System.out.print(tabuleiroFormatado);

    }

}

Classe JogoDaVelha

public class JogoDaVelha {

    public static *void* main(*String*[] *args*) {

*Tabuleiro* tabuleiro = **new** Tabuleiro();

        TabuleiroView.display(tabuleiro);

        // lógica do jogo

    }

}

Classe Tabuleiro

public class Tabuleiro {

    private *String*[] spots;

    public Tabuleiro() {

*this*.spots = **new** *String*[9];

        for (*int* i = 0; i < 9; i++) {

*this*.spots[i] = " ";

        }

    }

    public *String*[] getSpots() {

        return spots;

    }

    public *void* setSpot(*int* *index*, *String* *value*) {

*this*.spots[index] = value;

    }

    public *boolean* verificaGanhador() {

        if ((*this*.spots[0].equals(*this*.spots[1]) && *this*.spots[0].equals(*this*.spots[2])) ||

            (*this*.spots[3].equals(*this*.spots[4]) && *this*.spots[3].equals(*this*.spots[5])) ||

            (*this*.spots[6].equals(*this*.spots[7]) && *this*.spots[6].equals(*this*.spots[8])) ||

            (*this*.spots[0].equals(*this*.spots[3]) && *this*.spots[0].equals(*this*.spots[6])) ||

            (*this*.spots[1].equals(*this*.spots[4]) && *this*.spots[1].equals(*this*.spots[7])) ||

            (*this*.spots[2].equals(*this*.spots[5]) && *this*.spots[2].equals(*this*.spots[8])) ||

            (*this*.spots[0].equals(*this*.spots[4]) && *this*.spots[0].equals(*this*.spots[8])) ||

            (*this*.spots[2].equals(*this*.spots[4]) && *this*.spots[2].equals(*this*.spots[6]))) {

            return true;

        } else {

            return false;

        }

    }

}