

//Classe JogodaVelha

package meuJogo;

```
public class JogoDaVelha {  
    private static Jogo jogo;  
  
    public static void main(String[] args) {  
  
        setjogo1(new Jogo());  
    }  
  
    public static Jogo getjogo1() {  
        return jogo;  
    }  
  
    public static void setjogo1(Jogo jogo1) {  
        JogoDaVelha.jogo = jogo1;  
    }  
}
```

//Classe Jogo

package meuJogo;

import java.util.Scanner;

```
public class Jogo {  
    private Tabuleiro tabuleiro;  
    private int rodada;  
    private int vez;  
    private Jogador jogador1;
```

```
private Jogador jogador2;  
public Scanner entrada = new Scanner(System.in);  
private int opcao;
```

```
public Jogo() { // inicia o jogo  
    while (true) {  
        rodada = 1;  
        vez = 1;  
        tabuleiro = new Tabuleiro(); // Inicia o tabuleiro.  
        iniciarJogadores(); // Inicia os jogadores.  
        while (Jogar())  
            ;  
    }
```

```
        System.out.println("Se deseja jogar novamente aperte uma tecla para sim ou (N) para  
        não."); // Exibe na tela a pergunta de reiniciar o jogo.
```

```
        String resposta = entrada.next(); // Recebe a resposta da pergunta
```

```
        if (resposta.equalsIgnoreCase("n")) {  
            System.out.println("Jogo encerrado.");  
            break;  
        }
```

```
    }
```

```
}
```

```
public void iniciarJogadores() { // mensagem de inicio para escolher o jogador  
    jogador1 = new Humano(1);  
    int opcao = escolherJogador();  
    if (opcao == 1) {  
        jogador2 = new Humano(2);
```

```

    }

    else if (opcao == 2) {
        System.out.println("Jogador VS Computador");

        int computador = escolherComputador();
        switch (computador) {
            case 1:
                jogador2 = new ComputadorA(2);
                break;
            case 2:
                jogador2 = new ComputadorB(2);
                break;
            case 3:
                jogador2 = new ComputadorC(2);
                break;
        }
    }
}

public int escolherJogador() { // opção para escolher qual o tipo de jogador 2
    int opcao = 0;
    do {
        try {
            System.out.println("Escolha com quem quer jogar:");
            System.out.println("1. Humano");
            System.out.println("2. Computador\n");
            opcao = entrada.nextInt();
            System.out.println();
        }
    }
}

```

```

        if (opcao != 1 && opcao != 2) {
            System.out.println("Opção invalida, escolha denovo.");
            if (opcao == 2) {

            }
        }
    } catch (Exception e) {
        entrada.next();
        System.out.println("Opção invalida, escolha denovo.");
    }
} while (opcao != 1 && opcao != 2);
return opcao;
}

```

```

public int escolherComputador() { // opção para escolher o computador
    int opcao = 0;
    do {
        try {
            System.out.println("Escolha o Computador (1, 2 ou 3):");
            opcao = entrada.nextInt();
            System.out.println();
            if (opcao != 1 && opcao != 2 && opcao != 3) {
                System.out.println("Opção invalida, escolha denovo.");
            }
        } catch (Exception e) {
            entrada.next();
            System.out.println("Opção invalida, escolha denovo.");
        }
    } while (opcao != 1 && opcao != 2 && opcao != 3);
    return opcao;
}

```

```

}

public boolean Jogar() {
    if (ganhou() == 0) {
        if (tabuleiro.tabuleiroCompleto()) { // Exibe a mensagem se o tabuleiro estiver completo
            System.out.println("Tabuleiro Completo. Jogo empatado");
            return false;
        }
        Jogador jogadorAtual; // exibe na tela qual jogador esta jogando no momento e qual tem a
vez .

        String nomeDoJogador;
        if (vez() == 1) {
            jogadorAtual = jogador1;
            nomeDoJogador = "Jogador 1";
        } else {
            jogadorAtual = jogador2;
            if (jogadorAtual instanceof Computador) {
                nomeDoJogador = "Computador"; // exibe na tela com quem esta jogando
            } else {
                nomeDoJogador = "Jogador 2"; // exibe na tela com quem esta jogando
            }
        }
        System.out.println("");
        System.out.println("Rodada " + ((rodada + 1) / 2) + ", " + nomeDoJogador); // exibe na
tela com quem esta

        // jogando e em que rodada está
        jogadorAtual.jogar(tabuleiro);
        vez++;
        rodada++;
    }
}

```

```

        return true;
    } else {
        if (vez() == 2) {
            System.out.println("Jogador 1 ganhou!"); // exibe na tela o ganhador
        } else {
            if (jogador2 instanceof Computador) {
                System.out.println("Computador ganhou!");
            } else {
                System.out.println("Jogador 2 ganhou!");
            }
        }
        return false;
    }
}

public int vez() { // evita o jogo ficar em loop separando a vez de cada um
    if (vez % 2 == 1)
        return 1;
    else
        return 2;
}

public int ganhou() { // checa quem ganhou
    if (tabuleiro.checaLinhas() == 1)
        return 1;
    if (tabuleiro.checaColunas() == 1)
        return 1;
    if (tabuleiro.checaDiagonais() == 1)
        return 1;
    if (tabuleiro.checaLinhas() == -1)

```

```
        return -1;
    if (tabuleiro.checaColunas() == -1)
        return -1;
    if (tabuleiro.checaDiagonais() == -1)
        return -1;
    return 0;
}
```

```
public int getOpcao() {
    return opcao;
}
```

```
public void setOpcao(int opcao) {
    this.opcao = opcao;
}
```

```
}
//Classe Jogador
package meuJogo;
```

```
public abstract class Jogador { // classe jogador
    protected int[] tentativa = new int[2];
    protected int numeroDoJogador;
    protected int computador;

    public Jogador(int numeroDoJogador) {
        this.numeroDoJogador = numeroDoJogador;
    }

    public int getNumeroDoJogador() {
```

```

        return numeroDoJogador;
    }

    public abstract void jogar(Tabuleiro tabuleiro);

    public abstract void Tentativa(Tabuleiro tabuleiro);

    public boolean checaTentativa(int[] tentativa, Tabuleiro tabuleiro) {
        if (tabuleiro.getPosicao(tentativa) == 0)
            return true;
        else
            return false;
    }
}

```

//Classe Humano

```

package meuJogo;

import java.util.Scanner;

public class Humano extends Jogador { //Classe Humano que herda da classe Jogador
    public Scanner entrada = new Scanner(System.in);

    public Humano(int jogador) {
        super(jogador);
        this.numeroDoJogador = jogador;
    }
}

```



```

@Override
public void jogar(Tabuleiro tabuleiro) {

    Tentativa(tabuleiro);
    tabuleiro.setPosicao(tentativa, numeroDoJogador);
}

```

```

@Override
public void Tentativa(Tabuleiro tabuleiro) {

    do {
        do {
            try { //impede que exception aconteça
                System.out.print("Linha: ");
                tentativa[0] = entrada.nextInt();
                if (tentativa[0] > 3 || tentativa[0] < 1) //Checa se foi digitado uma opção
                    System.out.println("Linha inválida. É 1, 2 ou 3"); //exibe na tela quando
                    a opção digitada não é válida.
            } catch (Exception e) { //impede que exception aconteça
                try { //impede que exception aconteça
                    entrada.next();
                } catch (Exception e1) { //impede que exception aconteça
                }
                System.out.println("Linha inválida. É 1, 2 ou 3");
            }
        } while (tentativa[0] > 3 || tentativa[0] < 1);
    } do {
        try {

```

```

        System.out.print("Coluna: ");
        tentativa[1] = entrada.nextInt();
        if (tentativa[1] > 3 || tentativa[1] < 1) //Checa se foi digitado uma opção
        valida para coluna.
            System.out.println("Coluna inválida. É 1, 2 ou 3"); //exibe na tela
            quando a opção digitada não é válida.
        } catch (Exception e) {
            try {
                entrada.next();
            } catch (Exception e1) {
            }
            System.out.println("Coluna inválida. É 1, 2 ou 3");
        }
    } while (tentativa[1] > 3 || tentativa[1] < 1);
    tentativa[0]--;
    tentativa[1]--;
    if (!checaTentativa(tentativa, tabuleiro)) // para checar se o local onde será feita a
    jogada esta livre.
        System.out.println("Esse local já foi marcado. Tente outro.");
    } while (!checaTentativa(tentativa, tabuleiro));
}
}

```

//Classe Computador

```
package meujogo;
```

```
import java.util.Random;
```

public abstract class Computador **extends** Jogador { // Classe Computador que herda da classe Jogador e será
extendida as classes filhas ComputadorA,B e C.

```
public Computador(int jogador) { //Define o computador.  
    super(jogador);  
    this.numeroDoJogador = jogador;  
  
}
```

```
Random rand = new Random(); //Faz jogadas Randomicas.
```

```
@Override  
public void jogar(Tabuleiro tabuleiro) {  
    Tentativa(tabuleiro);  
    tabuleiro.setPosicao(tentativa, numeroDoJogador);  
}
```

```
@Override  
public abstract void Tentativa(Tabuleiro tabuleiro);  
}
```

//Classe ComputadorA

```
package meuJogo;
```

```
public class ComputadorA extends Computador {
```

```
    public ComputadorA(int jogador) { // ComputadorA definição de jogdas.
```

```
    super(jogador);  
    this.numeroDoJogador = jogador;
```

```
}
```

```
@Override
```

```
public void Tentativa(Tabuleiro tabuleiro) {//Implementação de como o ComputadorA joga  
    do {  
        tentativa[0] = rand.nextInt(3);  
        tentativa[1] = rand.nextInt(3);  
    } while (!checaTentativa(tentativa, tabuleiro));  
}
```

```
}
```

```
//Classe ComputadorB
```

```
package meuJogo;
```

```
public class ComputadorB extends Computador {
```

```
    public ComputadorB(int jogador) { //ComputadorB definição de jogdas.  
        super(jogador);  
        this.numeroDoJogador = jogador;
```

```
}
```

```
@Override
```

```
public void Tentativa(Tabuleiro tabuleiro) { //Implementação de como o ComputadorB joga  
    do {  
        tentativa[1] = rand.nextInt(3);
```

```
        tentativa[0] = rand.nextInt(3);  
    } while (!checaTentativa(tentativa, tabuleiro));  
}
```

```
}
```

```
//Classe ComputadorC
```

```
package meuJogo;
```

```
public class ComputadorC extends Computador {
```

```
    public ComputadorC(int jogador) { //ComputadorC definição de jogdas.  
        super(jogador);  
        this.numeroDoJogador = jogador;  
    }
```

```
}
```

```
@Override
```

```
public void Tentativa(Tabuleiro tabuleiro) { //Implementação de como o ComputadorC joga  
    for (int linha = 0; linha < 3; linha++) {  
        for (int coluna = 0; coluna < 3; coluna++) {  
            tentativa[0] = linha;  
            tentativa[1] = coluna;  
            if (checaTentativa(tentativa, tabuleiro)) {  
                return;  
            }  
        }  
    }  
}
```

```

}
//Classe Tabuleiro

package meuJogo;

public class Tabuleiro {
    private int[][] tabuleiro= new int[3][3]; //Define o tamanho do tabuleiro

    public Tabuleiro(){
        zerarTabuleiro();
    }

    public void zerarTabuleiro(){ //zera o tabuleiro
        for(int linha=0 ; linha<3 ; linha++)
            for(int coluna=0 ; coluna<3 ; coluna++)
                tabuleiro[linha][coluna]=0;
    }

    public void exibeTabuleiro(){ //Exibe o tabuleiro
        System.out.println();
        for(int linha=0 ; linha<3 ; linha++){

            for(int coluna=0 ; coluna<3 ; coluna++){

                if(tabuleiro[linha][coluna]==-1){
                    System.out.print(" X ");
                }
                if(tabuleiro[linha][coluna]==1){
                    System.out.print(" O ");
                }
            }
        }
    }
}

```

```

        if(tabuleiro[linha][coluna]==0){
            System.out.print("  ");
        }

        if(coluna==0 || coluna==1)
            System.out.print("|");
    }
    System.out.println();
}

}

public int getPosicao(int[] tentativa){ //Pega a posição do tabuleiro
    return tabuleiro[tentativa[0]][tentativa[1]];
}

public void setPosicao(int[] tentativa, int jogador){
    if(jogador == 1)
        tabuleiro[tentativa[0]][tentativa[1]] = -1;
    else
        tabuleiro[tentativa[0]][tentativa[1]] = 1;

    exibeTabuleiro();
}

public int checaLinhas(){ //Checa as linhas
    for(int linha=0 ; linha<3 ; linha++){

        if( (tabuleiro[linha][0] + tabuleiro[linha][1] + tabuleiro[linha][2]) == -3)
            return -1;
    }
}

```

```

        if( (tabuleiro[linha][0] + tabuleiro[linha][1] + tabuleiro[linha][2]) == 3)
            return 1;
    }

    return 0;
}

public int checaColunas(){ //Checa as colunas
    for(int coluna=0 ; coluna<3 ; coluna++){

        if( (tabuleiro[0][coluna] + tabuleiro[1][coluna] + tabuleiro[2][coluna]) == -3)
            return -1;
        if ((tabuleiro[0][coluna] + tabuleiro[1][coluna] + tabuleiro[2][coluna]) == 3)
            return 1;
    }

    return 0;
}

public int checaDiagonais(){//Checa as diagonais
    if( (tabuleiro[0][0] + tabuleiro[1][1] + tabuleiro[2][2]) == -3)
        return -1;
    if( (tabuleiro[0][0] + tabuleiro[1][1] + tabuleiro[2][2]) == 3)
        return 1;
    if( (tabuleiro[0][2] + tabuleiro[1][1] + tabuleiro[2][0]) == -3)
        return -1;
    if( (tabuleiro[0][2] + tabuleiro[1][1] + tabuleiro[2][0]) == 3)
        return 1;
}

```



```
        return 0;
    }

    public boolean tabuleiroCompleto(){ //
        for(int linha=0 ; linha<3 ; linha++)
            for(int coluna=0 ; coluna<3 ; coluna++)
                if( tabuleiro[linha][coluna]==0 )
                    return false;
        return true;
    }
}
```