

```
// ===== Arquivo: ./1a_Lista de Exercícios_Introdução_Davi Ferreira de Souza/Ex1.java
=====
import java.util.Scanner;

public class Ex1 {
    public static void main(String[] args)
    {
        double a,b,c, delta;
        Scanner scan = new Scanner(System.in);

        System.out.println("Insira o coeficiente a: ");
        a = scan.nextDouble();

        System.out.println("Insira o coeficiente b: ");
        b = scan.nextDouble();

        System.out.println("Insira o coeficiente c: ");
        c = scan.nextDouble();

        delta = Math.pow(b,2) - 4 * a * c;

        if(delta > 0)
        {
            double x1 = (-b + Math.sqrt(delta)) / (2 * a);
            double x2 = (-b - Math.sqrt(delta)) / (2 * a);

            System.out.println("x1 = " + x1 + "\nx2 = " + x2 + "\n");
        }
        else if(delta == 0)
        {
            double x = (-b / (2*a));

            System.out.println("x = " + x + "\n");
        }
        else
        {
            System.out.println("Equação não possui raízes reais.\n");
        }
    }
}
```

```
// ===== Arquivo: ./1a_Lista de Exercícios_Introdução_Davi Ferreira de Souza/Ex2.java
=====
import java.util.Scanner;

public class Ex2
{
    public static int fibonnaci(int n)
    {
        if(n == 0) return 0;
        if(n == 1) return 1;

        return fibonnaci(n - 1) + fibonnaci(n - 2);
    }

    public static void main(String[] args)
    {
        int n;
        Scanner scan = new Scanner(System.in);

        System.out.println("Insira o número para obter o n-ésimo termo de fibonnaci: ")
    }
}
```

```

;
    n = scan.nextInt();

    System.out.println("Fibonnaci(n) = " + fibonnaci(n) + "\n");
}
}

```

// ===== Arquivo: ./1a_Lista de Exercícios_Introdução_Davi Ferreira de Souza/Ex3.java
=====

```

import java.util.Scanner;

public class Ex3
{
    public static void bubbleSort(int[] vetor)
    {
        boolean ordenado = false;
        int temp;

        while(!ordenado)
        {
            ordenado = true;

            for(int i = 0; i < vetor.length - 1; i++)
            {
                if(vetor[i] > vetor[i+1])
                {
                    temp = vetor[i+1];
                    vetor[i+1] = vetor[i];
                    vetor[i] = temp;
                    ordenado = false;
                }
            }
        }
    }

    public static void main(String[] args)
    {
        int n;

        Scanner scan = new Scanner(System.in);

        System.out.print("Insira o número de elementos do vetor: ");
        n = scan.nextInt();

        int vetor[] = new int[n];

        for(int i = 0; i < n; i++)
        {
            System.out.println("Vetor[" + (i+1) + "] =");
            vetor[i] = scan.nextInt();
        }

        bubbleSort(vetor);

        for(int i = 0; i < n; i++)
        {
            System.out.print("Vetor[" + (i + 1) + "] = " + vetor[i] + "\n");
        }
    }
}

```

// ===== Arquivo: ./1a_Lista de Exercícios_Introdução_Davi Ferreira de Souza/Ex4.java
=====

```

import java.util.Scanner;

public class Ex4
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);

        int n;

        System.out.println("Insira a dimensÃo das matrizes: ");
        n = scan.nextInt();

        int[][] m1 = new int[n][n];
        int[][] m2 = new int[n][n];

        System.out.println("Insira os elementos da matriz 1:");

        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {
                m1[i][j] = scan.nextInt();
            }
        }

        System.out.println("Insira os elementos da matriz 2:");

        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {
                m2[i][j] = scan.nextInt();
            }
        }

        System.out.println("Matriz 1 + Matriz 2 = \n");

        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {
                System.out.print((m1[i][j] + m2[i][j]) + " ");
            }
            System.out.println();
        }
    }
}

```

// ===== Arquivo: ../1a_Lista de ExercÃcios_IntroduÃÃo_Davi Ferreira de Souza/Ex5.java =====

```

import java.io.IOException;
import java.util.Calendar;
import java.util.Scanner;
import java.text.SimpleDateFormat;

class Aluno
{
    String nome;
    Calendar dataNascimento;
    String RG;
    String endereco;
    String telefone;
    int RA;
    double CR;

    public void setNome(String nome){this.nome = nome;}
    public void setDataNascimento(Calendar data){this.dataNascimento = data;}
}

```

```

public void setRG(String RG){this.RG = RG;}
public void setEndereco(String endereco){this.endereco = endereco;}
public void setTelefone(String telefone){this.telefone = telefone;}
public void setRA(int RA){this.RA = RA;}
public void setCR(double CR){this.CR = CR;}

public String getNome(){return this.nome;}
public Calendar getDataNascimento(){return this.dataNascimento;}
public String getRG(){return this.RG;}
public String getEndereco(){return this.endereco;}
public String getTelefone(){return this.telefone;}
public int getRA(){return this.RA;}
public double getCR(){return this.CR;}

public void imprimeAluno()
{
    SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
    System.out.println("-----");
    System.out.println("ALUNO: " + this.nome);
    System.out.println("Data de Nascimento: " + sdf.format(this.dataNascimento.getTime()));
    System.out.println("RG: " + this.getRG());
    System.out.println("Endereco: " + this.getEndereco());
    System.out.println("Telefone: " + this.getTelefone());
    System.out.println("RA: " + this.getRA());
    System.out.println("CR: " + this.getCR());
    System.out.println("-----");
}
}

```

```

public class Ex5
{
    public static void ordenaVetor(Aluno[] alunos,int numAlunos, int escolha)
    {
        boolean ordenado = false;
        Aluno temp;
        if(escolha == 1)
        {
            while(!ordenado)
            {
                ordenado = true;
                for(int i = 0; i < numAlunos - 1; i++)
                {
                    if(alunos[i].getRA() > alunos[i+1].getRA())
                    {
                        temp = alunos[i];
                        alunos[i] = alunos[i+1];
                        alunos[i+1] = temp;
                        ordenado = false;
                    }
                }
            }
        }
        else if(escolha == 2)
        {
            while(!ordenado)
            {
                ordenado = true;
                for(int i = 0; i < numAlunos - 1; i++)
                {
                    if(alunos[i].getNome().compareTo(alunos[i+1].getNome()) > 0)
                    {
                        temp = alunos[i];
                        alunos[i] = alunos[i+1];
                        alunos[i+1] = temp;
                        ordenado = false;
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}
else
{
    while(!ordenado)
    {
        ordenado = true;
        for(int i = 0; i < numAlunos - 1; i++)
        {
            if(alunos[i].getCR() < alunos[i+1].getCR())
            {
                temp = alunos[i];
                alunos[i] = alunos[i+1];
                alunos[i+1] = temp;
                ordenado = false;
            }
        }
    }
}
}

public static void main(String[] args)
{
    Scanner scanner = new Scanner(System.in);
    Aluno[] alunos = new Aluno[100];

    int numAlunos = 0;
    boolean rodando = true;
    int escolha;

    while(rodando)
    {
        System.out.println("-----");
        System.out.println("1 - Cadastrar novo aluno.");
        System.out.println("2 - Ler o valor de um RA e imprimir os dados do aluno com este RA.");
        System.out.println("3 - Imprimir todos os cadastros.");
        System.out.println("4 - Sair do programa.\n");
        System.out.println("-----");

        System.out.println("Insira sua escolha: ");
        escolha = scanner.nextInt();

        switch(escolha)
        {
            case 1:
                Aluno alunoNovo = new Aluno();

                scanner.nextLine();
                System.out.print("Digite o nome do aluno: ");
                String nome = scanner.nextLine();

                System.out.print("Digite o dia de nascimento: ");
                int dia = scanner.nextInt();
                System.out.print("Digite o mês de nascimento (1-12): ");
                int mes = scanner.nextInt();
                System.out.print("Digite o ano de nascimento: ");
                int ano = scanner.nextInt();
                scanner.nextLine();

                Calendar dataNascimento = Calendar.getInstance();
                dataNascimento.set(ano, mes - 1, dia);

                System.out.print("Insira o RG do aluno: ");
                String RG = scanner.nextLine();

                System.out.print("Digite o endereço do aluno: ");
                String endereco = scanner.nextLine();

```

```

System.out.print("Insira o telefone do aluno: ");
String telefone = scanner.nextLine();

System.out.print("Insira o RA do aluno: ");
int RA = scanner.nextInt();

System.out.print("Insira o CR do aluno: ");
double CR = scanner.nextDouble();

alunoNovo.setNome(nome);
alunoNovo.setDataNascimento(dataNascimento);
alunoNovo.setRG(RG);
alunoNovo.setEndereco(endereco);
alunoNovo.setTelefone(telefone);
alunoNovo.setRA(RA);
alunoNovo.setCR(CR);

alunos[numAlunos] = alunoNovo;
numAlunos++;

break;
case 2:
    int RAbusca;

    System.out.println("Insira o RA do aluno a ser buscado: ");

    RAbusca = scanner.nextInt();

    for(int i = 0; i < numAlunos; i++)
    {
        if(alunos[i].getRA() == RAbusca)
        {
            alunos[i].imprimeAluno();
        }

        System.out.println("Aperte enter para continuar:");
        scanner.nextLine();

    }

    break;
case 3:

    System.out.println("-----");
----");

    System.out.println("1 - Imprimir alunos ordenados por RA");
    System.out.println("2 - Imprimir alunos ordenados por Nome");
    System.out.println("3 - Imprimir alunos ordenados por CR");
    System.out.println("-----");
----");

    System.out.println("Insira sua escolha: ");
    escolha = scanner.nextInt();

    ordenaVetor(alunos, numAlunos, escolha);

    for(int i = 0; i < numAlunos; i++)
    {
        alunos[i].imprimeAluno();
    }

    break;
case 4:
    rodando = false;
    break;
default:
    System.out.println("Escolha inválida.\nTente novamente.");
}
}

```

```

    }
}

// ===== Arquivo: ./4a_Lista de Exercícios_GUI_Davi Ferreira de Souza/Ex1.java =====
import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;

public class Ex1
{
    static JFrame mainFrame;
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(() ->
        {
            mainFrame = new JFrame("Ex1");
            mainFrame.setSize(new Dimension(800,800));
            mainFrame.setLocationRelativeTo(null);
            mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            JPanel painelPrincipal = new JPanel(new BorderLayout());

            JPanel painelBotoes = new JPanel();
            painelBotoes.setLayout(new BoxLayout(painelBotoes, BoxLayout.Y_AXIS));

            JButton botaoAbrir = new JButton("Abrir");
            JButton botaoSalvar = new JButton("Salvar");
            JButton botaoSalvarComo = new JButton("Salvar como");
            JButton botaoSair = new JButton("Sair");

            painelBotoes.add(botaoAbrir);
            painelBotoes.add(botaoSalvar);
            painelBotoes.add(botaoSalvarComo);
            painelBotoes.add(botaoSair);

            JTextArea fieldTexto = new JTextArea(30,20);

            painelPrincipal.add(painelBotoes, BorderLayout.LINE_START);
            painelPrincipal.add(fieldTexto, BorderLayout.CENTER);

            mainFrame.add(painelPrincipal);
            mainFrame.setVisible(true);

        });
    }
}

// ===== Arquivo: ./4a_Lista de Exercícios_GUI_Davi Ferreira de Souza/Ex2.java =====
import javax.swing.*;
import java.awt.*;

public class Ex2
{
    static JFrame mainFrame;
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(() ->

```

```

{
    mainFrame = new JFrame("Ex2");
    mainFrame.setSize(new Dimension(600,150));
    mainFrame.setLocationRelativeTo(null);
    mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JTextField textField1 = new JTextField(15);
    JTextField textField2 = new JTextField(15);

    JButton botaoSoma = new JButton("Soma");
    JButton botaoSubtrai = new JButton("Subtrai");
    JButton botaoMultiplica = new JButton("Multiplica");
    JButton botaoDivide = new JButton("Divide");

    JLabel label = new JLabel(" = 0");

    JPanel painelPrincipal = new JPanel(new FlowLayout(FlowLayout.CENTER));

    JPanel painelBotoes = new JPanel();
    painelBotoes.setLayout(new BoxLayout(painelBotoes, BoxLayout.Y_AXIS));

    painelBotoes.add(botaoSoma);
    painelBotoes.add(botaoSubtrai);
    painelBotoes.add(botaoMultiplica);
    painelBotoes.add(botaoDivide);

    painelPrincipal.add(textField1);
    painelPrincipal.add(painelBotoes);
    painelPrincipal.add(textField2);
    painelPrincipal.add(label);

    mainFrame.add(painelPrincipal);

    mainFrame.setVisible(true);

    });
}

```

// ===== Arquivo: ./4a_Lista de Exercícios_GUI_Davi Ferreira de Souza/Ex3.java =====

```

import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;

public class Ex3
{
    static JFrame mainFrame;
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(()->
        {
            mainFrame = new JFrame("Ex3");
            mainFrame.setSize(new Dimension(600,100));
            mainFrame.setLocationRelativeTo(null);
            mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            JTextField textField1 = new JTextField(15);
            JTextField textField2 = new JTextField(15);

            String[] opcoes = {"Soma", "Subtrai", "Multiplica", "Divide"};
            JComboBox caixaSelecao = new JComboBox<>(opcoes);
            caixaSelecao.setSelectedIndex(0);

            JLabel label = new JLabel(" = 0");

```



```

        JButton botaoCalcular = new JButton("Calcular");

        JPanel painelPrincipal = new JPanel(new BorderLayout());
        JPanel painelEntrada = new JPanel(new FlowLayout(FlowLayout.CENTER));
        JPanel painelBotao = new JPanel(new BorderLayout());

        painelBotao.add(botaoCalcular);

        painelEntrada.add(textField1);
        painelEntrada.add(caixaSelecao);
        painelEntrada.add(textField2);
        painelEntrada.add(label);

        painelPrincipal.add(painelEntrada, BorderLayout.NORTH);
        painelPrincipal.add(painelBotao, BorderLayout.CENTER);

        mainFrame.add(painelPrincipal);

        mainFrame.setVisible(true);

    });
}

// ===== Arquivo: ./4a_Lista de Exercícios_GUI_Davi Ferreira de Souza/Ex4.java =====
import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;

public class Ex4
{
    static JFrame mainFrame;
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(()->
        {
            mainFrame = new JFrame("Ex4");
            mainFrame.setSize(new Dimension(600,100));
            mainFrame.setLocationRelativeTo(null);
            mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            JTextField textField1 = new JTextField(15);
            JTextField textField2 = new JTextField(15);

            String[] opcoes = {"Soma", "Subtrai", "Multiplica", "Divide"};
            JComboBox caixaSelecao = new JComboBox<>(opcoes);
            caixaSelecao.setSelectedIndex(0);

            JLabel label = new JLabel(" = 0");

            JButton botaoCalcular = new JButton("Calcular");

            JPanel painelPrincipal = new JPanel(new BorderLayout());
            JPanel painelEntrada = new JPanel(new FlowLayout(FlowLayout.CENTER));
            JPanel painelBotao = new JPanel(new FlowLayout(FlowLayout.CENTER));

            painelBotao.add(botaoCalcular, BorderLayout.CENTER);

            painelEntrada.add(textField1);
            painelEntrada.add(caixaSelecao);
            painelEntrada.add(textField2);
            painelEntrada.add(label);

            painelPrincipal.add(painelEntrada, BorderLayout.NORTH);

```

```

        painelPrincipal.add(painelBotao, BorderLayout.CENTER);

        mainFrame.add(painelPrincipal);

        mainFrame.setVisible(true);

    });
}

}

// ===== Arquivo: ./5a_Lista de Exercícios_Listeners_Davi Ferreira de Souza/Ex1.java =====
import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.*;
import java.io.*;

public class Ex1
{
    static JFrame mainFrame;
    static String nomeArquivo = "";
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(() ->
        {

            mainFrame = new JFrame("Ex1");
            mainFrame.setSize(new Dimension(800,800));
            mainFrame.setLocationRelativeTo(null);
            mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            JPanel painelPrincipal = new JPanel(new BorderLayout());

            JPanel painelBotoes = new JPanel();
            painelBotoes.setLayout(new BoxLayout(painelBotoes, BoxLayout.Y_AXIS));

            JButton botaoAbrir = new JButton("Abrir");
            JButton botaoSalvar = new JButton("Salvar");
            JButton botaoSalvarComo = new JButton("Salvar como");
            JButton botaoSair = new JButton("Sair");

            JTextArea fieldTexto = new JTextArea(30,20);

            botaoAbrir.setMaximumSize(new Dimension(200,50));
            botaoSalvar.setMaximumSize(new Dimension(200,50));
            botaoSalvarComo.setMaximumSize(new Dimension(200,50));
            botaoSair.setMaximumSize(new Dimension(200,50));

            painelBotoes.add(botaoAbrir);
            painelBotoes.add(botaoSalvar);
            painelBotoes.add(botaoSalvarComo);
            painelBotoes.add(botaoSair);

            //Função botões:
            botaoAbrir.addMouseListener(new MouseAdapter()
            {
                @Override
                public void mouseClicked(MouseEvent e)
                {
                    nomeArquivo = JOptionPane.showInputDialog("Insira o nome do arquivo:

");

                    StringBuffer txt = new StringBuffer();
                    try {
                        FileInputStream in = new FileInputStream(nomeArquivo);

```

```

        Scanner sin = new Scanner(in);
        while (sin.hasNextLine())
        {
            txt.append(sin.nextLine()).append("\n");
        }
        in.close();

        fieldTexto.setText(txt.toString());
    }
    catch (FileNotFoundException ex)
    {
        JOptionPane.showMessageDialog(mainFrame, "Erro: Arquivo não encontrado.");
    }
    catch (IOException ex)
    {
        JOptionPane.showMessageDialog(mainFrame, "Erro ao ler o arquivo: " + ex.getMessage());
    }
}

});

botaoSalvar.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        if(nomeArquivo == "")
        {
            nomeArquivo = JOptionPane.showInputDialog("Insira o nome do arquivo: ");
        }

        String texto = fieldTexto.getText();

        try
        {
            FileOutputStream out = new FileOutputStream(nomeArquivo);
            out.write(texto.getBytes("UTF-8"));
            out.close();
        }
        catch (FileNotFoundException ex)
        {
            JOptionPane.showMessageDialog(mainFrame, "Erro: Arquivo não encontrado.");
        }
        catch (IOException ex)
        {
            JOptionPane.showMessageDialog(mainFrame, "Erro ao ler o arquivo: " + ex.getMessage());
        }
    }
});

botaoSalvarComo.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e)
    {
        nomeArquivo = JOptionPane.showInputDialog("Insira o nome do arquivo: ");

        String texto = fieldTexto.getText();

        try

```

```

        {
            FileOutputStream out = new FileOutputStream(nomeArquivo);
            out.write(texto.getBytes("UTF-8"));
            out.close();
        }
        catch (FileNotFoundException ex)
        {
            JOptionPane.showMessageDialog(mainFrame, "Erro: Arquivo não encontrado.");
        }
        catch (IOException ex)
        {
            JOptionPane.showMessageDialog(mainFrame, "Erro ao ler o arquivo: " + ex.getMessage());
        }
    }
});

botaoSair.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        System.exit(0);
    }
});

painelPrincipal.add(painelBotoes, BorderLayout.LINE_START);
painelPrincipal.add(fieldTexto, BorderLayout.CENTER);

mainFrame.add(painelPrincipal);
mainFrame.setVisible(true);

});

}
}

```

// ===== Arquivo: ./5a_Lista de Exercícios_Listeners_Davi Ferreira de Souza/Ex2.java =====

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.KeyAdapter;

public class Ex2
{
    static JFrame mainFrame;
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(()->
        {
            mainFrame = new JFrame("Ex2");
            mainFrame.setSize(new Dimension(600,150));
            mainFrame.setLocationRelativeTo(null);
            mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            JTextField textField1 = new JTextField(15);
            JTextField textField2 = new JTextField(15);

            JButton botaoSoma = new JButton("Soma");

```

```

JButton botaoSubtrai = new JButton("Subtrai");
JButton botaoMultiplica = new JButton("Multiplica");
JButton botaoDivide = new JButton("Divide");

botaoSoma.setMaximumSize(new Dimension(100,50));
botaoSubtrai.setMaximumSize(new Dimension(100,50));
botaoMultiplica.setMaximumSize(new Dimension(100,50));
botaoDivide.setMaximumSize(new Dimension(100,50));

JLabel label = new JLabel(" = 0");

label.setMaximumSize(new Dimension(100,50));

JPanel painelPrincipal = new JPanel(new FlowLayout(FlowLayout.CENTER));

JPanel painelBotoes = new JPanel();
painelBotoes.setLayout(new BoxLayout(painelBotoes, BoxLayout.Y_AXIS));

textField1.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyTyped(KeyEvent e)
    {
        if(Character.isLetter(e.getKeyChar()))
        {
            e.consume();
        }
    }
});

textField2.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyTyped(KeyEvent e)
    {
        if(Character.isLetter(e.getKeyChar()))
        {
            e.consume();
        }
    }
});

botaoSoma.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        if(textField1.getText().isEmpty() || textField2.getText().isEmpty())
        {
            if(textField1.getText().isEmpty())
            {
                JOptionPane.showMessageDialog(mainFrame, "Insira um n mero v
 ilido no input 1.", null, 0);
            }

            if(textField2.getText().isEmpty())
            {
                JOptionPane.showMessageDialog(mainFrame, "Insira um n mero v
 ilido no input 1.", null, 0);
            }
        }
        else
        {
            label.setText(" = " + String.valueOf(Double.parseDouble(textField
1.getText()) + Double.parseDouble(textField2.getText())));
            label.revalidate();
            label.repaint();
        }
    }
});

```

```

    });

    botaoSubtrai.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent e)
        {
            if(textField1.getText().isEmpty() || textField2.getText().isEmpty())
            {
                if(textField1.getText().isEmpty())
                {
                    JOptionPane.showMessageDialog(mainFrame, "Insira um número v
                    ã;lido no input 1.", null, 0);
                }

                if(textField2.getText().isEmpty())
                {
                    JOptionPane.showMessageDialog(mainFrame, "Insira um número v
                    ã;lido no input 1.", null, 0);
                }

            }
            else
            {
                label.setText(" = " + String.valueOf(Double.parseDouble(textField
                1.getText()) - Double.parseDouble(textField2.getText())));
                label.revalidate();
                label.repaint();
            }
        }
    });

    botaoMultiplica.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent e)
        {
            if(textField1.getText().isEmpty() || textField2.getText().isEmpty())
            {
                if(textField1.getText().isEmpty())
                {
                    JOptionPane.showMessageDialog(mainFrame, "Insira um número v
                    ã;lido no input 1.", null, 0);
                }

                if(textField2.getText().isEmpty())
                {
                    JOptionPane.showMessageDialog(mainFrame, "Insira um número v
                    ã;lido no input 1.", null, 0);
                }

            }
            else
            {
                label.setText(" = " + String.valueOf(Double.parseDouble(textField
                1.getText()) * Double.parseDouble(textField2.getText())));
                label.revalidate();
                label.repaint();
            }
        }
    });

    botaoDivide.addMouseListener(new MouseAdapter()
    {
        @Override
        public void mouseClicked(MouseEvent e)
        {
            if(textField1.getText().isEmpty() || textField2.getText().isEmpty())
            {
                if(textField1.getText().isEmpty())
                {

```

```

        JOptionPane.showMessageDialog(mainFrame, "Insira um número válido no input 1.", null, 0);
    }

    if(textField2.getText().isEmpty())
    {
        JOptionPane.showMessageDialog(mainFrame, "Insira um número válido no input 1.", null, 0);
    }

    else if(Double.parseDouble(textField2.getText()) == 0)
    {
        JOptionPane.showMessageDialog(mainFrame, "Não é possível realizar divisões por zero.", null, 0);
    }
    else
    {
        label.setText(" = " + String.format("%.2f", Double.parseDouble(textField1.getText()) / Double.parseDouble(textField2.getText())));
        label.revalidate();
        label.repaint();
    }
});

painelBotoes.add(botaoSoma);
painelBotoes.add(botaoSubtrai);
painelBotoes.add(botaoMultiplica);
painelBotoes.add(botaoDivide);

painelPrincipal.add(textField1);
painelPrincipal.add(painelBotoes);
painelPrincipal.add(textField2);
painelPrincipal.add(label);

mainFrame.add(painelPrincipal);

mainFrame.setVisible(true);

});
}
}

```

// ===== Arquivo: ./5a_Lista de Exercícios_Listeners_Davi Ferreira de Souza/Ex3/Exemplo1.java =====

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.KeyAdapter;

```

// Utiliza um JComboBox para seleção de operações e um JButton para exibir a operação selecionada.

// O ActionListener do JComboBox atualiza a variável com a operação escolhida pelo usuário,

// enquanto o ActionListener do JButton exibe essa informação em uma caixa de diálogo.

```

public class Exemplo1
{

```

```

static JFrame mainFrame;
static String opcaoSelecionada;
public static void main(String[] args)
{
    SwingUtilities.invokeLater(()->
    {
        mainFrame = new JFrame("Exemplo 1 - Ex3");
        mainFrame.setSize(new Dimension(800,800));
        mainFrame.setLocationRelativeTo(null);
        mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        String opcoes[] = {"Opcao 1", "Opcao 2", "Opcao 3", "Opcao 4"};

        opcaoSelecionada = opcoes[0];

        JComboBox<String> caixaSelecao = new JComboBox<>(opcoes);
        JButton botao = new JButton("Mostra seleÃ§Ã£o");
        JPanel painelPrincipal = new JPanel(new FlowLayout());

        caixaSelecao.setSelectedIndex(0);

        caixaSelecao.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                opcaoSelecionada = String.valueOf(caixaSelecao.getSelectedItem());
            }
        });

        botao.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e)
            {
                JOptionPane.showMessageDialog(mainFrame, "OpÃ§Ã£o selecionada: " + opcaoS
elecionada, "FunÃ§Ã£o botÃ£o", 0);
            }
        });

        painelPrincipal.add(botao);
        painelPrincipal.add(caixaSelecao);

        mainFrame.add(painelPrincipal);
        mainFrame.setVisible(true);

    });
}

```

// ===== Arquivo: ./5a_Lista de ExercÃ-cios_Listeners_Davi Ferreira de Souza/Ex3/Exemplo2.java =====

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.KeyAdapter;

```

// Este exemplo demonstra o uso de KeyListeners para restringir a entrada do usuÃrio em campos de texto (JTextField).
// SÃo apresentados trÃs campos: um que aceita apenas letras, outro que aceita apenas nÃmeros e um terceiro que aceita apenas sÃmbolos (excluindo letras e nÃmeros).
// Esse tipo de validaÃÃo Ã muito Ãtil em formulÃrios de cadastro ou entradas contro

ladas, garantindo que o usuário digite apenas o tipo de dado esperado.

```
public class Exemplo2
{
    static JFrame mainFrame;

    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(() ->
        {
            mainFrame = new JFrame("Exemplo 2 - Ex3");
            mainFrame.setSize(new Dimension(800,400));
            mainFrame.setLocationRelativeTo(null);
            mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            JPanel painelPrincipal = new JPanel();
            painelPrincipal.setLayout(new BoxLayout(painelPrincipal, BoxLayout.Y_AXIS));

            JLabel labelLetra = new JLabel("Essa TextField só aceita letras: ");
            JLabel labelNumero = new JLabel("Essa TextField só aceita números: ");
            JLabel labelSimbolos = new JLabel("Essa TextField só aceita caracteres que n
            ão são letras ou números");

            JTextField textFieldLetra = new JTextField(10);
            JTextField textFieldNumero = new JTextField(10);
            JTextField textFieldSimbolos = new JTextField(10);

            textFieldLetra.addKeyListener(new KeyAdapter() {
                @Override
                public void keyTyped(KeyEvent e)
                {
                    if(!Character.isLetter(e.getKeyChar()))
                    {
                        e.consume();
                    }
                }
            });

            textFieldNumero.addKeyListener(new KeyAdapter() {
                @Override
                public void keyTyped(KeyEvent e)
                {
                    if(!Character.isDigit(e.getKeyChar()))
                    {
                        e.consume();
                    }
                }
            });

            textFieldSimbolos.addKeyListener(new KeyAdapter() {
                @Override
                public void keyTyped(KeyEvent e)
                {
                    if(Character.isDigit(e.getKeyChar()) || Character.isLetter(e.getKeyCh
ar()))
                    {
                        e.consume();
                    }
                }
            });

            painelPrincipal.add(labelLetra);
            painelPrincipal.add(textFieldLetra);
            painelPrincipal.add(labelNumero);
            painelPrincipal.add(textFieldNumero);
            painelPrincipal.add(labelSimbolos);
            painelPrincipal.add(textFieldSimbolos);
        }
    }
}
```

```

        mainFrame.add(painelPrincipal);

        mainFrame.setVisible(true);

    });
}
}

// ===== Arquivo: ./5a_Lista de Exercícios_Listeners_Davi Ferreira de Souza/Ex3/Exemplo3
.java =====
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.event.KeyAdapter;

// Este exemplo demonstra o uso de MouseListener combinado com uma Thread para implementa
r a funcionalidade de arrastar (drag and drop) uma JLabel.
// Quando o usuário pressiona o botão do mouse sobre a label, uma thread é iniciada pa
ra atualizar continuamente sua posição conforme o mouse se move.
// A movimentação é interrompida quando o botão do mouse é liberado.
// Este exemplo mostra como capturar eventos de clique e movimento do mouse, além de int
egrar manipulação de interface com controle concorrente (thread).

public class Exemplo3
{
    static JFrame mainFrame;
    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(() ->
        {
            mainFrame = new JFrame("Exemplo 3 - Ex3");
            mainFrame.setSize(new Dimension(800,800));
            mainFrame.setLocationRelativeTo(null);
            mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            JPanel painelPrincipal = new JPanel(null);

            JLabel label = new JLabel("Me arraste");
            label.setBackground(new Color(0,0,0));
            label.setForeground(new Color(255,255,255));
            label.setBounds(200, 200, 100, 30);
            label.setOpaque(true);

            Point offset = new Point();

            label.addMouseListener(new MouseAdapter()
            {
                boolean arrastando = false;
                public void mousePressed(MouseEvent e)
                {
                    offset.setLocation(e.getPoint());
                    arrastando = true;
                    new Thread(() ->
                    {
                        while (arrastando)
                        {
                            try
                            {

```

```

        PointerInfo pi = MouseInfo.getPointerInfo();
        Point p = pi.getLocation(); // posição global do mouse
        SwingUtilities.convertPointFromScreen(p, painelPrincipal)

;

        int newX = p.x - offset.x;
        int newY = p.y - offset.y;

        SwingUtilities.invokeLater(() ->
        {
            label.setLocation(newX, newY);
        });

        Thread.sleep(1);
    }
    catch (InterruptedException ex)
    {
        ex.printStackTrace();
    }
    }).start();
}

public void mouseReleased(MouseEvent e)
{
    arrastando = false;
}
});

painelPrincipal.add(label);

mainFrame.add(painelPrincipal);
mainFrame.setVisible(true);

});
}
}

```

// ===== Arquivo: ./6a_Lista de Exercícios_Collections_Davi Ferreira de Souza/Ex1.java =====

```

import java.util.*;

public class Ex1
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        int n;

        System.out.print("Insira o número de alunos: ");
        n = scan.nextInt();

        scan.nextLine();

        Vector<String> nomeAlunos = new Vector<>();

        for(int i = 0; i < n; i++)
        {
            nomeAlunos.add(scan.nextLine());
        }

        nomeAlunos.sort(null);

        System.out.println("Nome dos alunos ordenados: ");
        System.out.print(nomeAlunos.toString());

    }
}

```

```
}
```

```
// ===== Arquivo: ./6a_Lista de Exercícios_Collections_Davi Ferreira de Souza/Ex2.java =====
```

```
import java.util.*;
```

```
public class Ex2
```

```
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);

        TreeSet<String> alunos = new TreeSet<>();

        System.out.print("Insira o número de alunos: ");
        int n = scan.nextInt();
        scan.nextLine();

        for(int i = 0; i < n; i++)
        {
            alunos.add(scan.nextLine());
        }

        System.out.println("Alunos ordenados: ");
        System.out.print(alunos.toString());

    }
}
```

```
// ===== Arquivo: ./6a_Lista de Exercícios_Collections_Davi Ferreira de Souza/Ex3.java =====
```

```
import java.util.*;
```

```
public class Ex3
```

```
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);

        int n = scan.nextInt();
        int[] alunos = new int[n];

        for (int i = 0; i < n; i++)
        {
            alunos[i] = scan.nextInt();
        }

        Arrays.sort(alunos);

        int soma = 0;
        for (int i = 0; i < n - 1; i += 2)
        {
            if (alunos[i] != alunos[i+1])
            {
                soma += (alunos[i+1] - alunos[i]);
            }
        }

        System.out.println(soma);
    }
}
```

```
// ===== Arquivo: ./6a_Lista de Exercícios_Collections_Davi Ferreira de Souza/Ex4.java =
=====
import java.util.*;

public class Ex4
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);

        int n;

        n = scan.nextInt();
        scan.nextLine();

        for(int i = 0; i < n; i++)
        {
            Stack<Character> diamantes = new Stack<>();
            int soma = 0;
            String entrada;

            entrada = scan.nextLine();

            for(int j = 0; j < entrada.length(); j++)
            {
                if(entrada.charAt(j) == '<'){
                    diamantes.push('<');
                }
                if(entrada.charAt(j) == '>')
                {
                    if(!diamantes.isEmpty())
                    {
                        if(diamantes.peek() == '<')
                        {
                            diamantes.pop();
                            soma = soma + 1;
                        }
                    }
                }
            }

            System.out.println(soma);
        }
    }
}
```

```
// ===== Arquivo: ./6a_Lista de Exercícios_Collections_Davi Ferreira de Souza/Ex5.java =
=====
import java.util.*;

public class Ex5
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        int p, a, r;

        while (true)
        {
            p = scan.nextInt();
            a = scan.nextInt();
            r = scan.nextInt();
            if(scan.hasNextLine()) scan.nextLine();

            if (p == 0 && a == 0 && r == 0)
            {
                break;
            }
        }
    }
}
```

```

    }

    TreeSet<String> perolas = new TreeSet<>();
    HashMap<String, Integer> alunos_perolas = new HashMap<>();

    for (int i = 0; i < p; i++)
    {
        perolas.add(scan.nextLine());
    }

    for (int i = 0; i < a; i++)
    {
        String nomeAluno = scan.nextLine();
        int soma_perolas = 0;

        for (int j = 0; j < r; j++)
        {
            String perola = scan.nextLine();
            if (perolas.contains(perola))
            {
                soma_perolas++;
            }
        }

        alunos_perolas.put(nomeAluno, soma_perolas);
    }

    int maxPerolas = Collections.max(alunos_perolas.values());
    List<String> melhoresAlunos = new ArrayList<>();

    for (Map.Entry<String, Integer> entry : alunos_perolas.entrySet())
    {
        if (entry.getValue() == maxPerolas)
        {
            melhoresAlunos.add(entry.getKey());
        }
    }

    Collections.sort(melhoresAlunos);
    System.out.println(String.join(", ", melhoresAlunos));
}

scan.close();
}
}

```

// ===== Arquivo: ../7a_Lista de Exercícios-Threads_Davi Ferreira de Souza/Ex2.java =====
import java.util.*;

```

class OperacaoMatriz extends Thread
{
    private int [][] A,B,matriz_soma;
    private int inicioLinha, inicioColuna, finalLinha, finalColuna;

    public OperacaoMatriz(int[][] A, int[][] B, int[][] matriz_soma, int inicioLinha, int
    inicioColuna, int finalLinha, int finalColuna)
    {
        this.A = A;
        this.B = B;
        this.matriz_soma = matriz_soma;
        this.inicioLinha = inicioLinha;
        this.inicioColuna = inicioColuna;
        this.finalLinha = finalLinha;
        this.finalColuna = finalColuna;
    }

    @Override
    public void run()

```

```

    {
        for(int i = inicioLinha; i < finalLinha; i++)
        {
            for(int j = inicioColuna; j < finalColuna; j++)
            {
                this.matriz_soma[i][j] = this.A[i][j] + this.B[i][j];
            }
        }
    }
}

public class Ex2
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);

        System.out.print("Insira o tamanho da matriz: ");
        int n = scan.nextInt();

        int[][] A = new int[n][n];
        int[][] B = new int[n][n];
        int[][] matriz_soma = new int[n][n];

        System.out.println("Insira os elementos da matriz A:");

        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {
                A[i][j] = scan.nextInt();
            }
        }

        System.out.println("Insira os elementos da matriz B:");

        for(int i = 0; i < n; i++)
        {
            for(int j = 0; j < n; j++)
            {
                B[i][j] = scan.nextInt();
            }
        }

        int meio = n/2;

        Thread t1 = new OperacaoMatriz(A, B, matriz_soma, 0, 0, meio, meio);
        Thread t2 = new OperacaoMatriz(A, B, matriz_soma, 0, meio, meio, n);
        Thread t3 = new OperacaoMatriz(A, B, matriz_soma, meio, 0, n, meio);
        Thread t4 = new OperacaoMatriz(A, B, matriz_soma, meio, meio, n, n);

        t1.start();
        t2.start();
        t3.start();
        t4.start();

        try
        {
            t1.join();
            t2.join();
            t3.join();
            t4.join();
        }
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }

        System.out.println("Matriz soma: ");
        for(int i = 0; i < n; i++)

```

```

    {
        for(int j = 0; j < n; j++)
        {
            System.out.print(matriz_soma[i][j] + " ");
        }
        System.out.println("");
    }

```

```

        scan.close();
    }
}

```

// ===== Arquivo: ../7a_Lista de Exercícios-Threads_Davi Ferreira de Souza/Ex7.java =====

```

import javax.swing.*;
import javax.swing.border.Border;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.util.*;
import java.io.*;

```

```

class RodaCacaNiquel implements Runnable

```

```

{
    private JTextField textField;

    public RodaCacaNiquel(JTextField textField)
    {
        this.textField = textField;
    }

    @Override
    public void run()
    {

        int numeroDeGiros = (int) (Math.random() * 151) + 1;

        for(int i = 0; i < numeroDeGiros; i++)
        {
            SwingUtilities.invokeLater(()->
            {
                textField.setEnabled(true);
                textField.setText(String.valueOf((int) (Math.random() * 8)));
                textField.setEnabled(false);
            });

            try
            {
                Thread.sleep(50);
            }
            catch (InterruptedException ex)
            {
                Thread.currentThread().interrupt();
            }
        }
    }
}

```

```

public class Ex7 {

    static JFrame mainFrame;
    public static void main(String[] args)
    {

```



```

mainFrame = new JFrame("Caça Niquel");
mainFrame.setSize(new Dimension(450, 450));
mainFrame.setLocationRelativeTo(null);
mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

mainFrame.setLayout(new BoxLayout(mainFrame.getContentPane(), BoxLayout.Y_AXIS));

//Declaração do painel
JPanel painelNumeros = new JPanel();
painelNumeros.setLayout(new BoxLayout(painelNumeros, BoxLayout.X_AXIS));
painelNumeros.setAlignmentX(Component.CENTER_ALIGNMENT);

//Declaração dos JTextFields
Dimension fieldSize = new Dimension(100, 100);
Font fieldFont = new Font("SansSerif", Font.BOLD, 36);
JTextField num1 = new JTextField(2);
JTextField num2 = new JTextField(2);
JTextField num3 = new JTextField(2);

//Declaração do botão
JButton botao = new JButton("Jogar");

//Estilos
num1.setEnabled(false);
num2.setEnabled(false);
num3.setEnabled(false);

for (JTextField num : new JTextField[]{num1, num2, num3})
{
    num.setPreferredSize(fieldSize);
    num.setMaximumSize(fieldSize);
    num.setHorizontalAlignment(JTextField.CENTER);
    num.setFont(fieldFont);
    num.setEnabled(false);
}

botao.setAlignmentX(Component.CENTER_ALIGNMENT);
botao.setPreferredSize(new Dimension(200, 60));
botao.setMaximumSize(new Dimension(200, 60));
botao.setFont(new Font("SansSerif", Font.BOLD, 18));

botao.setMaximumSize(new Dimension(300, 100));

botao.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e)
    {
        Thread t1 = new Thread(new RodaCacaNiquel(num1));
        Thread t2 = new Thread(new RodaCacaNiquel(num2));
        Thread t3 = new Thread(new RodaCacaNiquel(num3));

        t1.start();
        t2.start();
        t3.start();

        new Thread(() ->
        {
            try
            {
                t1.join();
                t2.join();
            }
        }
    }
});

```

```

        t3.join();
    }
    catch (InterruptedException ex)
    {
        Thread.currentThread().interrupt();
    }
    // agendar o diÃ¡logo na EDT
    SwingUtilities.invokeLater(() ->
    {
        if (num1.getText().equals(num2.getText())
            && num1.getText().equals(num3.getText()))
        {
            JOptionPane.showMessageDialog(mainFrame, "VocÃª venceu!", "Ve
nceu!", 1);
        }
        else
        {
            JOptionPane.showMessageDialog(mainFrame, "VocÃª perdeu.", "Pe
rdeu.", 1);
        }
    });
    }).start();

    }
});

painelNumeros.add(num1);
painelNumeros.add(Box.createRigidArea(new Dimension(20, 0)));
painelNumeros.add(num2);
painelNumeros.add(Box.createRigidArea(new Dimension(20, 0)));
painelNumeros.add(num3);

mainFrame.add(painelNumeros);
mainFrame.add(Box.createRigidArea(new Dimension(0, 100)));
mainFrame.add(botao);

mainFrame.setLocationRelativeTo(null);
mainFrame.setVisible(true);

    }
}

```

```

// ===== Arquivo: ./8a_Lista_de_ExercÃcios_Jogo da Velha em Rede_Davi Ferreira de Souza/
Cliente.java =====
import java.net.*;
import java.util.*;
import java.io.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Cliente
{
    private static JFrame mainFrame;

    //Cores
    private static Color backgroundColor = new Color(249, 249, 249); // #F9F9F9
    private static Color foregroundColor = new Color(51, 51, 102); // #333366
    private static Color accentColor = new Color(255, 111, 97); // #FF6F61
    private static Color accentHover = new Color(255, 211, 182); // #FFD3B6
    private static Color inputBorderColor = new Color(204, 204, 204); // #CCCCCC
    private static Color secondaryButtonColor = new Color(108, 91, 123); // #6C5B7B
    private static Color inputBackground = new Color(255, 255, 255); // #FFFFFF
}

```

```

private static Color formBackground = new Color(252, 238, 245); // #FCEEF5
private static Color highlightColor = new Color(255, 215, 0); // #FFD700

//Fonte
private static Font fonte = new Font("Helvetica", Font.PLAIN, 16);

//Rede
private static Socket cliente;
private static PrintWriter out;
private static BufferedReader in;

//Jogo
private static int numJogador = 0;
private static JButton[] botoes = new JButton[9];
private static boolean minhaVez = false;

public static void main(String[] args)
{
    mainFrame = new JFrame("Jogo da velha");
    mainFrame.setSize(new Dimension(800,800));
    mainFrame.setLocationRelativeTo(null);
    mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    menuConexao();

    mainFrame.setVisible(true);
}

private static void menuConexao()
{
    mainFrame.getContentPane().removeAll();

    JPanel painelPrincipal = new JPanel();
    JPanel painelTextFields = new JPanel(new GridBagLayout());

    painelPrincipal.setLayout(new BoxLayout(painelPrincipal, BoxLayout.Y_AXIS));

    //DeclaraÃ§Ã£o JLabel
    JLabel labelIp = new JLabel("Insira o IP do servidor: ");
    JLabel labelPort = new JLabel("Insira o port do servidor: ");

    //DeclaraÃ§Ã£o TextField
    JTextField fieldIp = new JTextField(15);
    JTextField fieldPort = new JTextField(15);

    //DeclaraÃ§Ã£o botÃ£o
    JButton botao = new JButton("Continuar");

    //Estilos

    JLabel[] labels = {labelIp, labelPort};
    JTextField[] fields = {fieldIp, fieldPort};

    for(JLabel j:labels)
    {
        j.setForeground(formBackground);
        j.setFont(fonte.deriveFont(Font.BOLD, 16));
        j.setPreferredSize(new Dimension(200, 80));
    }

    for(JTextField f:fields)
    {
        f.setForeground(formBackground);
        f.setBackground(inputBackground);
        f.setFont(fonte.deriveFont(Font.PLAIN, 24));
        f.setPreferredSize(new Dimension(200, 50));
        f.setBorder(BorderFactory.createLineBorder(inputBorderColor));
    }
}

```

```

f.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyTyped(KeyEvent e)
    {
        if(!Character.isDigit(e.getKeyChar()) && !(e.getKeyChar() == '.'))
        {
            e.consume();
        }
    }
});
}

botao.setPreferredSize(new Dimension(200, 60));
botao.setBackground(accentColor);
botao.setForeground(Color.WHITE);
botao.setFont(fonte.deriveFont(Font.BOLD, 18));
botao.setFocusPainted(false);
botao.setBorder(BorderFactory.createLineBorder(highlightColor, 2));
botao.setOpaque(true);
botao.setContentAreaFilled(true);

painelPrincipal.setBackground(background-color);
mainFrame.setBackground(background-color);
painelTextFields.setBackground(background-color);

//Funcionalidades
botao.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        botao.setBackground(accentHover);
    }

    @Override
    public void mouseExited(MouseEvent e)
    {
        botao.setBackground(accentColor);
    }

    @Override
    public void mouseClicked(MouseEvent e)
    {
        if(!fieldIp.getText().isEmpty() && !fieldPort.getText().isEmpty())
        {
            try
            {
                cliente = new Socket(fieldIp.getText(), Integer.parseInt(fieldPort.getText()));

                out = new PrintWriter(cliente.getOutputStream(), true);
                in = new BufferedReader(new InputStreamReader(cliente.getInputStream()));

                catch(Exception excep){JOptionPane.showMessageDialog(mainFrame, excep.getMessage(), 0);}
            }
            finally
            {
                String recebido;
                try
                {
                    recebido = in.readLine();
                    if(("START1").equals(recebido))
                    {
                        numJogador = 1;
                    }
                    else if(("START2").equals(recebido))
                    {
                        numJogador = 2;
                    }
                }
            }
        }
    }
});

```

```

        menuJogo();
        listenerServidor();
    } catch (Exception excep) { JOptionPane.showMessageDialog(mainFrame,
excep.getStackTrace(), excep.getMessage(), 0); }

    }
}
else
{
    JOptionPane.showMessageDialog(mainFrame, "Entrada inválida em um dos
campos", "Entrada inválida", 0);
}
});

GridBagConstraints gbc = new GridBagConstraints();

gbc.gridx = 0;
gbc.gridy = 0;

painelTextFields.add(labelIp, gbc);

gbc.gridx++;

painelTextFields.add(labelPort, gbc);

gbc.gridy = 1;
gbc.gridx = 0;

painelTextFields.add(fieldIp, gbc);

gbc.gridx++;

painelTextFields.add(fieldPort, gbc);

gbc.gridx = 0;
gbc.gridy = 2;
gbc.gridwidth = 2;
gbc.anchor = GridBagConstraints.CENTER;
gbc.insets = new Insets(20, 0, 0, 0);

painelTextFields.add(botao, gbc);

painelPrincipal.add(painelTextFields);

mainFrame.add(painelPrincipal);
}

private static void menuJogo()
{
    mainFrame.getContentPane().removeAll();
    mainFrame.setTitle("Jogador " + numJogador);

    JPanel painelBotoes = new JPanel(new GridLayout(3, 3, 5, 5));

    for(int i = 0; i < 9; i++)
    {
        JButton botao = new JButton();
        final int index = i;

        botao.setFont(fonte);
        botao.setPreferredSize(new Dimension(80, 80));

        botao.setBackground(accentColor);
        botao.setForeground(Color.WHITE);
        botao.setFont(fonte.deriveFont(Font.BOLD, 24));
        botao.setFocusPainted(false);
    }
}

```

```

        botao.setBorder(BorderFactory.createLineBorder(highlightColor, 2));
        botao.setOpaque(true);
        botao.setContentAreaFilled(true);

        botao.getModel().addChangeListener(e -> {
            ButtonModel model = botao.getModel();
            if (model.isRollover() && minhaVez && botao.getText().isEmpty()) {
                botao.setBackground(accentHover);
            } else {
                String text = botao.getText();
                if ("X".equals(text)) {
                    botao.setBackground(new Color(100, 149, 237));
                } else if ("O".equals(text)) {
                    botao.setBackground(new Color(152, 251, 152));
                } else {
                    botao.setBackground(accentColor);
                }
            }
        });

        botao.addMouseListener(new MouseAdapter()
        {
            @Override
            public void mouseClicked(MouseEvent e)
            {
                if (minhaVez && botao.getText().isEmpty())
                {
                    out.println("JOGADA:" + index);
                }
            }
        });

        botoes[i] = botao;
        painelBotoes.add(botao);
    }

    mainFrame.add(painelBotoes);

    mainFrame.revalidate();
    mainFrame.repaint();
}

private static void listenerServidor()
{
    new Thread(() ->
    {
        try
        {
            String linha;
            while ((linha = in.readLine()) != null)
            {
                String[] parts = linha.split(":");
                switch (parts[0])
                {
                    case "ESTADO":
                        String grid = parts[1];
                        minhaVez = Integer.parseInt(parts[2]) == 1;
                        SwingUtilities.invokeLater(() -> updateBoard(grid, minhaVez))
                        ;

                        break;
                    case "GAMEOVER":
                        System.out.println("DEBUG: Recebido GAMEOVER - " + linha);
                        String finalGrid = parts[1];
                        String result = parts[2];
                        SwingUtilities.invokeLater(() -> {
                            updateBoard(finalGrid, false);
                            showGameOver(result);
                        });
                        break;
                }
            }
        }
    }
}

```

```

        }
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
})).start();
}

```

```

private static void updateBoard(String grid, boolean minhaVezAtual)
{
    for (int i = 0; i < 9; i++)
    {
        char c = grid.charAt(i);
        botoes[i].setText(c == '-' ? "" : String.valueOf(c));
        botoes[i].setEnabled(minhaVezAtual && botoes[i].getText().isEmpty());

        if (botoes[i].getText().equals("X"))
        {
            botoes[i].setBackground(new Color(100, 149, 237));
        }
        else if (botoes[i].getText().equals("O"))
        {
            botoes[i].setBackground(new Color(152, 251, 152));
        }
        else
        {
            botoes[i].setBackground(accentColor);
        }
    }
}

```

```

}

private static void setButtonsEnabled(boolean enabled)
{
    for (JButton b : botoes)
    {
        b.setEnabled(enabled);
    }
}

```

```

private static void showGameOver(String result)
{
    String msg;
    if (result.equals("WIN" + numJogador)) msg = "VocÃa venceu!";
    else if (result.equals("WIN" + (3 - numJogador))) msg = "VocÃa perdeu!";
    else msg = "Empate!";
    JOptionPane.showMessageDialog(mainFrame, msg, msg, JOptionPane.PLAIN_MESSAGE);
    setButtonsEnabled(false);

    System.exit(0);
}
}

```

```

// ===== Arquivo: ./8a_Lista_de_ExercÃcios_Jogo da Velha em Rede_Davi Ferreira de Souza/
Server.java =====
import java.net.*;
import java.util.*;
import java.io.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

```

```

class JogoVelha
{
    private char[] jogo = new char[9];
}

```

```

private int jogadorAtual = 1;

public JogoVelha()
{
    for(int i = 0; i < 9; i++)
    {
        jogo[i] = '-';
    }
}

public synchronized boolean marcaPosicao(int index)
{
    if(index < 0 || index >= 9 || jogo[index] != '-') return false;

    jogo[index] = (jogadorAtual == 1) ? 'X' : 'O';
    return true;
}

public synchronized void updatePlayer()
{
    jogadorAtual = 3 - jogadorAtual;
}

public synchronized String getEstadoAtual()
{
    return new String(jogo);
}

public synchronized int getJogadorAtual()
{
    return jogadorAtual;
}

public synchronized String checaVitoria()
{
    int[][] vitorias =
    {
        {0,1,2},{3,4,5},{6,7,8},
        {0,3,6},{1,4,7},{2,5,8},
        {0,4,8},{2,4,6}
    };

    for(int[] v : vitorias)
    {
        if (jogo[v[0]] != '-' &&
            jogo[v[0]] == jogo[v[1]] &&
            jogo[v[1]] == jogo[v[2]])
        {
            return jogo[v[0]] == 'X' ? "WIN1" : "WIN2";
        }
    }

    for (char c : jogo)
    {
        if (c == '-') return null;
    }
    return "DRAW";
}

}

class EstadoJogo implements Runnable
{
    private Socket s1, s2;
    private PrintWriter o1, o2;
    private BufferedReader i1, i2;
    private JogoVelha jogo = new JogoVelha();

    public EstadoJogo(Socket c1, Socket c2) throws IOException

```



```

{
    this.s1 = c1;
    this.s2 = c2;
    o1 = new PrintWriter(s1.getOutputStream(), true);
    o2 = new PrintWriter(s2.getOutputStream(), true);
    i1 = new BufferedReader(new InputStreamReader(s1.getInputStream()));
    i2 = new BufferedReader(new InputStreamReader(s2.getInputStream()));
}

@Override
public void run()
{
    try {
        o1.println("START1");
        o2.println("START2");

        Thread updateThread = new Thread(() ->
        {
            try
            {
                while (jogo.checaVitoria() == null)
                {
                    String estado = jogo.getEstadoAtual();
                    int vez = jogo.getJogadorAtual();

                    o1.println("ESTADO:" + estado + ":" + (vez == 1 ? 1 : 0));
                    o2.println("ESTADO:" + estado + ":" + (vez == 2 ? 1 : 0));

                    Thread.sleep(1);
                }
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        });
        updateThread.start();

        while(true)
        {
            int vez = jogo.getJogadorAtual();
            BufferedReader in = (vez == 1) ? i1 : i2;

            String line = in.readLine();
            if (line == null || !line.startsWith("JOGADA:"))
            {
                break;
            }

            int posicaoJogada = Integer.parseInt(line.substring(7));

            if(jogo.marcaPosicao(posicaoJogada))
            {
                String resultado = jogo.checaVitoria();
                String estado = jogo.getEstadoAtual();

                if (resultado != null)
                {
                    if (resultado != null)
                    {
                        if (resultado != null) {

                            o1.println("GAMEOVER:" + estado + ":" + resultado);
                            o2.println("GAMEOVER:" + estado + ":" + resultado);

                            try { Thread.sleep(100); } catch (InterruptedException e)
                                {}

                            // Fecha as conexões
                            o1.close(); o2.close();
                            i1.close(); i2.close();
                        }
                    }
                }
            }
        }
    }
}

```

```

        s1.close(); s2.close();

        break;
    }
}

jogo.updatePlayer();
}
}
}
catch(Exception e)
{
    e.printStackTrace();
}
finally
{
    try
    {
        i1.close(); o1.close(); s1.close();
        i2.close(); o2.close(); s2.close();
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
}
}

public class Server
{
    private ServerSocket server;

    public void start(int port)
    {
        try {
            server = new ServerSocket(port);
            System.out.println("Servidor iniciado: " + InetAddress.getLocalHost().getHostAddress() + ":" + port);

            while (true) {
                Socket cliente1 = server.accept();
                System.out.println("Cliente 1 conectado: " + cliente1.getRemoteSocketAddress());

                Socket cliente2 = server.accept();
                System.out.println("Cliente 2 conectado: " + cliente2.getRemoteSocketAddress());

                new Thread(new EstadoJogo(cliente1, cliente2)).start();
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
        finally
        {
            stop();
        }
    }

    public void stop() {
        try
        {
            if (server != null && !server.isClosed())
            {
                server.close();
                System.out.println("Servidor encerrado.");
            }
        }
    }
}

```

```

        catch (Exception e)
        {
            e.printStackTrace();
        }
    }

    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        int port;

        System.out.print("Insira o port do servidor: ");
        port = scan.nextInt();

        Server a = new Server();
        a.start(port);

        scan.close();
    }
}

```

// ===== Arquivo: ./Botao_flutuante_Davi Ferreira de Souza/BotaoFlutuante.java =====

```

import javax.swing.*;
import java.awt.*;

class Main
{
    static private JFrame mainFrame;

    public static int direcao_x = 1;
    public static int direcao_y = 1;

    public static void main(String[] args)
    {
        SwingUtilities.invokeLater(() ->
        {
            int width = 80;
            int height = 40;

            mainFrame = new JFrame("Botao Flutuante");
            mainFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            mainFrame.setSize(800, 800);
            mainFrame.setLocationRelativeTo(null);
            mainFrame.setLayout(null);
            mainFrame.getContentPane().setBackground(new Color(0,0,0));

            JButton botao = new JButton("Ok");
            botao.setLocation(new Point(100,350));
            botao.setSize(new Dimension(width, height));
            botao.setVisible(true);
            botao.setOpaque(true);

            botao.addActionListener(e->{System.exit(0);});

            new Thread(() ->
            {
                try
                {
                    while (true)
                    {
                        Thread.sleep(10);

                        SwingUtilities.invokeLater(() ->
                        {
                            int x = botao.getX();
                            int y = botao.getY();

```

```

0)         if(x + width >= mainFrame.getContentPane().getWidth() || x <=
{
    direcao_x *= -1;
}
<= 0)     if(y + height >= mainFrame.getContentPane().getHeight() || y
{
    direcao_y *= -1;
}

    botao.setLocation(x + 2 * direcao_x, y + 2 * direcao_y);

    Color corAtual = botao.getBackground();
    int r = (corAtual.getRed() + 1) % 256;
    int g = (corAtual.getGreen() + 2) % 256;
    int b = (corAtual.getBlue() + 3) % 256;

    botao.setBackground(new Color(r, g, b));
    botao.setForeground(new Color(r,g,b).brighter());

        });
    }
    catch (InterruptedException e) {
        e.printStackTrace();
    }
    }).start();

    mainFrame.add(botao);
    mainFrame.setVisible(true);
});

}
}

```

// ===== Arquivo: ./Contest Collections/1861.java =====

```

import java.io.IOException;
import java.util.*;
public class Main {

    public static void main(String[] args) throws IOException
    {
        Scanner scan = new Scanner(System.in);
        HashMap<String, Integer> numero_assassinados = new HashMap<String, Integer>();
        ArrayList<String> lista_assassinados = new ArrayList<>();

        while(scan.hasNext())
        {
            String assassino;
            String assassinado;

            assassino = scan.next();
            assassinado = scan.next();

            lista_assassinados.add(assassinado);

            if(numero_assassinados.containsKey(assassino))
            {
                numero_assassinados.put(assassino, numero_assassinados.get(assassino) + 1
);
            }
            else
            {

```

```

        numero_assassinados.put (assassino, 1);
    }

}
System.out.println("HALL OF MURDERERS");
List<String> nomes = new ArrayList<>(numero_assassinados.keySet());
Collections.sort(nomes);

for(String a : nomes)
{
    if(!lista_assassinados.contains(a)) {
        System.out.println(a + " " + numero_assassinados.get(a));
    }
}

}

}

```

```

// ===== Arquivo: ./Contest Collections/Main.java =====
import java.io.IOException;
import java.util.Scanner;
import java.util.Stack;

public class Main {

    public static void main(String[] args) throws IOException {

        Scanner scanner = new Scanner(System.in);

        while(scanner.hasNext())
        {
            String str = scanner.next();
            Stack <Character> parenteses = new Stack<>();
            boolean ok = true;

            for(int i = 0; i < str.length(); i++)
            {
                if(str.charAt(i) == ')')
                {
                    if(parenteses.isEmpty())
                    {
                        ok = false;
                        break;
                    }

                    if(parenteses.peek() != '(')
                    {
                        ok = false;
                        break;
                    }
                    else
                    {
                        parenteses.pop();
                    }
                }

                if(str.charAt(i) == '(') {parenteses.add('(');}
            }

            if(ok && parenteses.isEmpty())
            {
                System.out.println("correct");
            }
        }
    }
}

```

```

        }
        else
        {
            System.out.println("incorrect");
        }
    }
}
}

```

// ===== Arquivo: ../Teste Socket/Cliente.java =====

```

import java.net.*;
import java.io.*;
import java.util.*;

public class Cliente
{
    private static Socket cliente;
    private static PrintWriter out;
    private static BufferedReader in;

    private void startConnection(String ip, int port)
    {
        try
        {
            cliente = new Socket(ip, port);
            out = new PrintWriter(cliente.getOutputStream(), true);
            in = new BufferedReader(new InputStreamReader(cliente.getInputStream()));
        } catch (Exception e) {e.printStackTrace();}
    }

    public String sendMessage(String msg)
    {
        out.println(msg);
        try
        {
            String resp = in.readLine();
            return resp;
        } catch (Exception e) {e.printStackTrace();}

        return "";
    }

    public void stopConnection()
    {
        try
        {
            in.close();
            out.close();
            cliente.close();
        } catch (Exception e) {e.printStackTrace();}
    }

    public static void main(String[] args)
    {
        Cliente c = new Cliente();
        Scanner scan = new Scanner(System.in);

        c.startConnection("localhost", 1234);
        while (scan.hasNext())
        {
            c.sendMessage(scan.nextLine());
        }
        c.stopConnection();
    }
}

```

```

    }

}

// ===== Arquivo: ./Teste Socket/ServerTeste.java =====
import java.net.*;
import java.io.*;

public class ServerTeste {
    private ServerSocket server;
    private Socket cliente;

    private PrintWriter out;
    private BufferedReader in;

    public void start(int port)
    {
        try
        {
            server = new ServerSocket(port);
            cliente = server.accept();
            out = new PrintWriter(cliente.getOutputStream(), true);
            in = new BufferedReader(new InputStreamReader(cliente.getInputStream()));
        } catch (Exception e) {e.printStackTrace();}

        String a;
        try
        {
            while ((a = in.readLine()) != null) {
                System.out.println("Cliente: " + a);
                out.println("Servidor recebeu: " + a);
            }
        } catch (Exception e) {e.printStackTrace();}

    }

    public void stop() {
        try
        {
            in.close();
            out.close();
            cliente.close();
            server.close();
        } catch (Exception e) {e.printStackTrace();}
    }

    public static void main(String[] args)
    {
        ServerTeste a = new ServerTeste();
        a.start(1234);
    }
}

```