

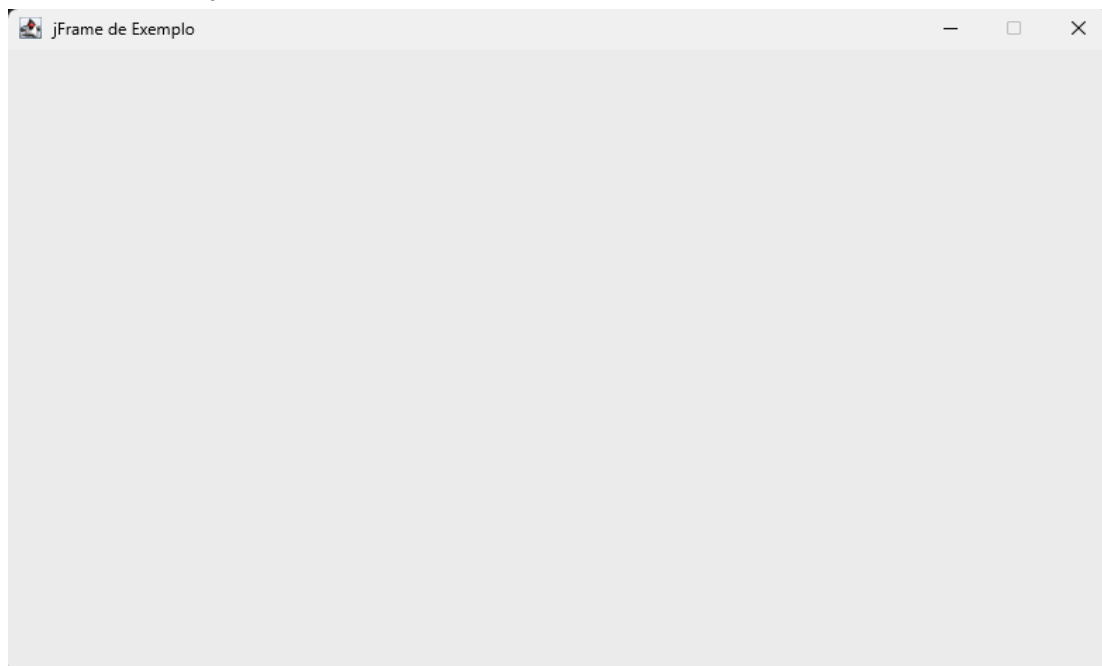
1. Escreva um trabalho explicando Java Swing ou JavaFX. Implemente exemplos que ilustrem os tópicos abordados no trabalho.

Java Swing é uma biblioteca própria para a criação de janelas e interfaces de usuário nativa da linguagem de programação Java. Ela contém métodos para a utilização de widgets pré-fabricados nas interfaces gráficas, como botões, labels, campos de texto e outros. É conhecida por ser leve, de fácil uso e por ser responsável por desenhar os componentes da interface por conta própria, sem atarefar o sistema operacional com essa responsabilidade. Todas as aplicações usando Java Swing acabam tendo uma aparência similar. Em Java Swing, as janelas são chamadas de JFrames. Exemplo de criação de um JFrame padrão:

```
import javax.swing.*;

public class JFrame_Exemplo_01 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("JFrame de Exemplo");
        frame.setSize(800, 500);
        frame.setResizable(false);
        frame.setLayout(null);
        frame.setLocationRelativeTo(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Note a biblioteca Java Swing sendo importada para o projeto na classe Main, bem como a utilização de seus métodos a partir da classe JFrame. É instanciado um objeto JFrame chamado frame, o qual é dado como parâmetro o seu título, “JFrame de Exemplo”, e em seguida são definidas as demais características da janela, como o tamanho, se é redimensionável ou não e entre outros.



Explicação Linha por Linha:

```
JFrame frame = new JFrame("JFrame de Exemplo");
```

É instanciado o JFrame frame e configurado o seu título (o título aparecerá no topo da janela).

```
frame.setSize(800, 500);
```

É configurado a altura e a largura de frame, no caso, 800 por 500 pixels.

```
frame.setResizable(false);
```

É configurado que frame não será redimensionável.

```
frame.setLayout(null);
```

É configurado que frame terá um layout nulo, o que permite maior liberdade na hora de distribuir seus elementos.

```
frame.setLocationRelativeTo(null);
```

É configurada a posição original de frame na tela. "null" garante que a janela aparecerá no centro absoluto da tela.

```
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

É configurado o comportamento que frame terá quando o usuário clicar no botão X de saída, No caso, ele irá encerrar todas as atividades do programa.

```
frame.setVisible(true);
```

É por fim, é configurado se frame será ou não visível para o usuário. No caso, "true" garante que ele será completamente visível para o usuário.

Note também que, como não foram instanciados outros objetos pertencentes à janela, como por exemplo, JLabels, JButtons ou outros, estes não estão presentes no JFrame.

Vamos instanciar um JButton.

```
import javax.swing.*;
```

```
public class JFrame_Exemplo_01 {
```

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

```
        JFrame frame = new JFrame("JFrame de Exemplo");
```

```
        JButton button = new JButton("Teste");
```

```
        button.setBounds(350, 220, 100, 40);
```

```
        frame.add(button);
```

```
        frame.setSize(800, 500);
```

```
        frame.setResizable(false);
```

```
        frame.setLayout(null);
```

```
        frame.setLocationRelativeTo(null);
```

```
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

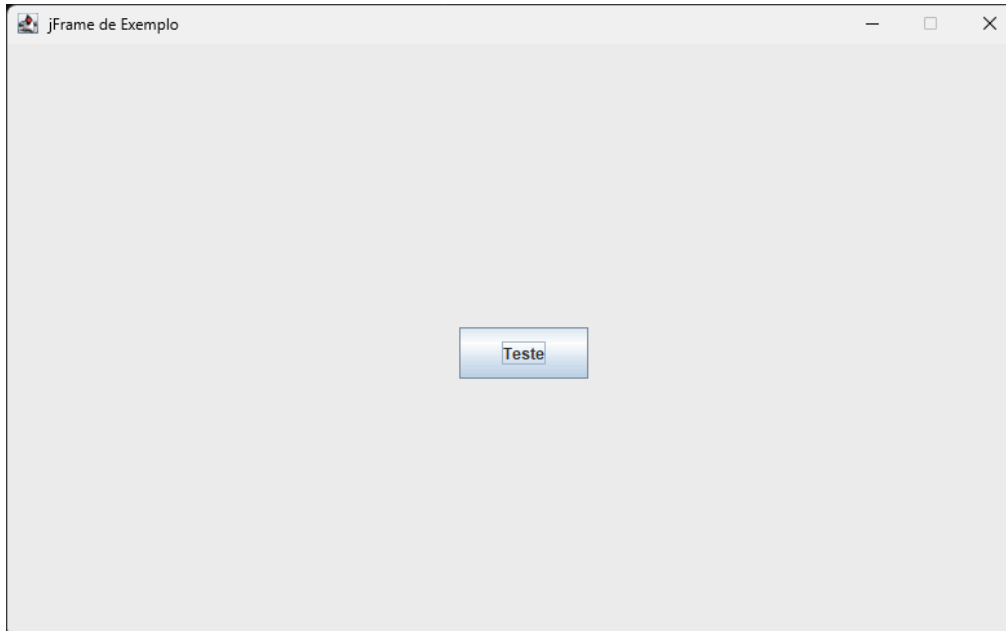
```
        frame.setVisible(true);
```

```
    }
```

```
}
```

Note que agora, há um JButton nomeado de “button” e com a String “Teste” sendo dada como parâmetro. A String define o texto que ficará no centro do botão que está sendo adicionado a Frame.

O código agora resulta na seguinte janela ao ser executado:



Explicação passo a passo da criação de **button**:

```
JButton button = new JButton("Teste");  
button.setBounds(350, 220, 100, 40);  
frame.add(button);
```

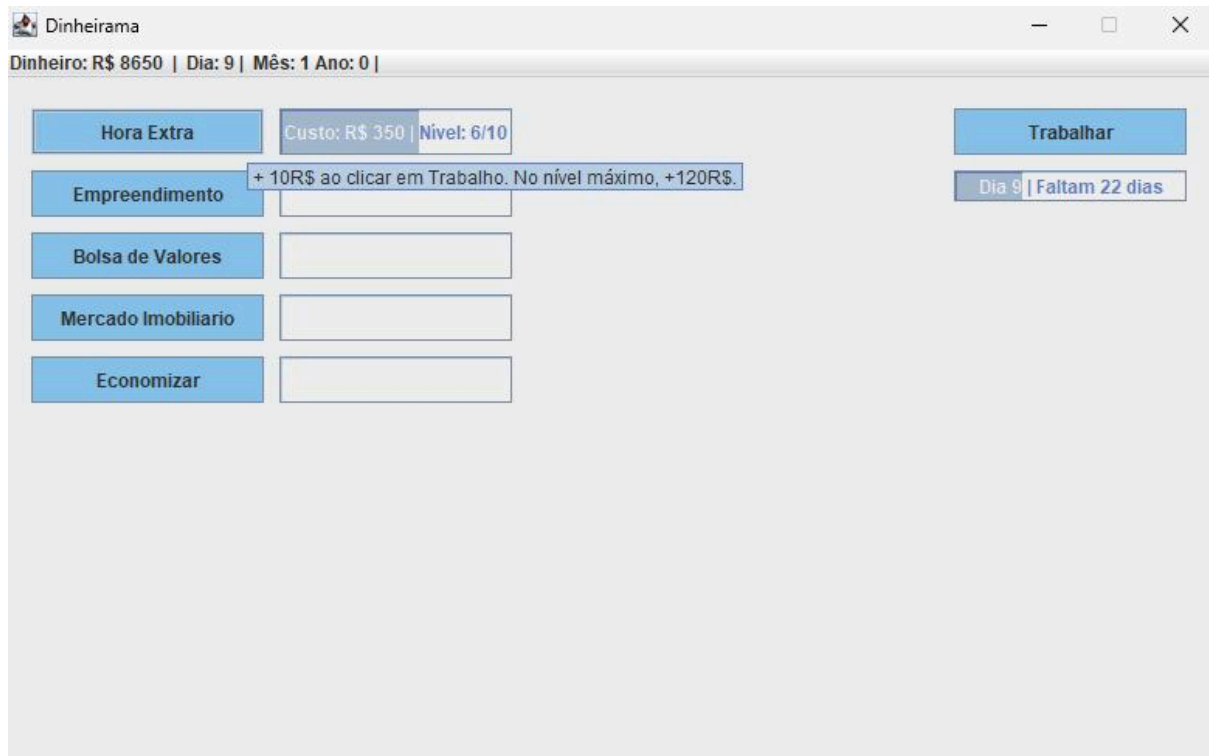
- 1) É instanciado um JButton chamado *button* e atribuído ao seu texto a String “Teste”. Texto aparecerá no centro do botão.
- 2) É configurado as posições *X* e *Y* do botão mais sua largura e altura em *setBounds*.
- 3) Por fim, *button* é adicionado a **frame** por meio de *frame.add(button)*.

É possível configurar muitas coisas mais em um JButton, como por exemplo, a cor de fundo e a cor do texto que é exibido. Caso nada seja configurado, o JButton padrão do Java Swing será utilizado. No caso do código usado como exemplo acima, *button* é um JButton padrão.

É possível também criar JLabels, como por exemplo,

```
private JLabel currentMoneyLabel;  
currentMoneyLabel = new JLabel(" Dinheiro: R$ " + currentMoney +  
" | ");  
menuBar.add(currentMoneyLabel);
```

Utilizando dos widgets disponíveis pelo Java Swing, é possível fazer janelas completas. Exemplo de janela com elementos aleatórios:



Neste JFrame intitulado “Dinheirama”, há seis botões JButton personalizados com a cor azul, seis barras JProgressBar e um JMenuBar com labels variadas. O Java Swing consegue ser bem integrado com as muitas classes do Java, o que permite, por exemplo, a criação do calendário genérico utilizado no programa por meio da classe Timer.

Também é possível criar painéis de pop up usando javax.swing.JOptionPane:

```
JOptionPane.showMessageDialog(null, "Olá, este é um exemplo de painel de popup ");
```

Estes painéis de pop up também podem servir como forma de obter dados do usuário, como por exemplo, um popup pedindo o nome ou alguma outra informação.

Os pop ups são descartados assim que é clicado no botão OK ou qualquer outro correspondente. O dado salvo na variável permanece.

O JMenuBar também é capaz de armazenar botões e barras de progresso, sendo um componente bastante diverso.