Raul Alexandre Gonzalez Augusto RA: 211023698

1. **Faça uma pesquisa na Internet e explique o que a classe Frame e JFrame faz.**

Frame é uma janela com título e borda, é um container awt capaz de ter componentes disponíveis no awt, como botões, campos de textos..., como é gerada pelo sistema operacional os componentes devem ser nativos do SO e o estilo também será do SO.

Já o JFrame é uma versão estendida do Frame do awt, agora permitindo o uso de componentes da arquitetura Swing, mas é um pouco incompatível com o Frame, por isso não é comum utiliza-los juntos, além disso diferente do Frame o JFrame tem comportamentos padrões quando o usuário tenta fechar a janela, no Frame a gente precisa programar o funcionamento do do botão de sair.

2. Nas transparências de 3 até 9, execute cada um dos programas, descreva rapidamente (no máximo 7 linhas) o que eles fazem e comente linha a linha indicando a função de cada linha nos programas.

No Button cria uma janela com um botão escrito aperte-me e titulo Botao. No CheckBox cria uma janela com titulo Checkbox e com uma caixa de seleção escrito caixa de verificação. No outro CheckBox faz o mesmo do primeiro Checkbox mas não usa uma classe herdando Frame, Cria direto na classe principal. No Choice Cria uma janela com titulo Choice e com dois items de escolha, porém sobrepostos então so o item 1 aparece. No terceiro Checkbox adiciona um gridlayout permitindo que tenha duas caixas de seleção uma embaixo da outra. No GridLayout cria uma janela de titulo GridLayout com botões no layout de 3 linhas e 2 colunas.

Button

```
import java.awt.*; //importa awt
public class Botao extends Frame { // cria classe herdando Frame
       Button b = new Button("Aperte-me"); // cria instancia de Button b escrito
Aperte-me
 Botao( ) { //construtor
 super("Botao"); // titulo da janela
 add(b); // adiciona b na janela
 pack( ); // dimensiona a janela  para caber todos os componentes e layouts
com seus tamanhos ideais
 setVisible(true); //deixa janela visivel
 }

 static public void main(String[] args) { //cria método principal
 new Botao( ); // instancia classe Botao
```

```
 }
}
```

CheckBox

```
import java.awt.*; //importa awt
class CaixaVerif extends Frame { // cria classe herdando Frame
 Checkbox cb = new Checkbox("Caixa de Verificação"); // cria instancia de
Checkbox cb escrito Caixa de Verificação
 CaixaVerif( ) { //construtor da classe
 super("Checkbox"); // titulo da janela
 add(cb); // adiciona cb na janela
 pack( ); // dimensiona a janela  para caber todos os componentes e layouts
com seus tamanhos ideais
 setVisible(true); //deixa janela visivel
 }
       static public void main(String[] args) { //cria método principal
 new CaixaVerif( ); // instancia classe CaixaVerif
 }
}
```

CheckBox 2

```
import java.awt.*; //importa awt
class CaixaVerif1 { // cria classe
 static public void main(String[] args) { //cria metodo principal
 Checkbox cb = new Checkbox("Caixa de Verificação"); // cria instancia de
Checkbox cb escrito Caixa de Verificação
 Frame f = new Frame("CheckBox"); //cria instancia de frame f com titulo
CheckBox
 f.add(cb); //adiciona cb no f
 f.pack( );  //dimensiona a janela  para caber todos os componentes e layouts
com seus tamanhos ideais
 f.setVisible(true); //deixa janela visivel
 }
}
```

Choice

```
import java.awt.*; //importa awt
class Escolha extends Frame { // cria classe herdando Frame
 Escolha( ) { //construtor da classe
 super("Choice"); // titulo da janela
 Choice choice = new Choice(); //Cria instancia de Choice choice
 choice.addItem("Item 1"); //adiciona item 1 em choice
 choice.addItem("Item 2"); //adiciona item 2 em choice
 add(choice); // //adiciona choice na janela
```

```
 pack( ); // dimensiona a janela  para caber todos os componentes e layouts
com seus tamanhos ideais
 setVisible(true); //deixa janela visivel
 }
 static public void main(String[] args) { //cria metodo principal
 new Escolha( ); // instancia classe Escolha
 }
}
```

CheckBox3

```
import java.awt.*; //importa awt
class CaixaVerif2 extends Frame { // cria classe herdando Frame
 Checkbox cb1 = new Checkbox("Verificação UM"); //cria instancia de Checkbox
cb1 escrito Verificação UM
 Checkbox cb2 = new Checkbox("Verificação DOIS"); //cria instancia de Checkbox
cb2 escrito Verificação DOIS
 CaixaVerif2( ) { //construtor da classe
 super("Checkbox"); // titulo da janela
 setLayout(new GridLayout(2, 1)); // faz o layout ser do tipo grade de 2
linhas e 1 coluna
 add(cb1); //adiciona cb1 na janela
 add(cb2); //adiciona cb2 na janela
 pack( ); //dimensiona a janela  para caber todos os componentes e layouts com
seus tamanhos ideais
 setVisible(true); //deixa janela visivel
 }
 static public void main(String[] args) { //cria metodo principal
 new CaixaVerif2( ); // instancia classe CaixaVerif2
 }
}
```

GridLayout

```
import java.awt.*; //importa awt
class LayoutGrade extends Frame { //cria classe herdando Frame
 LayoutGrade( ) { //construtor da classe
 super("GridLayout"); // titulo da janela
 setLayout(new GridLayout(3, 2)); // faz o layout ser do tipo grade de 3
linhas e 2 coluna
 add(new Button("Botao 1")); //adiciona instancia de Button escrito Botao 1 na
janela
 add(new Button("2")); //adiciona instancia de Button escrito 2 na janela
 add(new Button("Botao 3")); //adiciona instancia de Button escrito Botao 3 na
janela
 add(new Button("4")); //adiciona instancia de Button escrito 4 na janela
 add(new Button("Botao 5")); //adiciona Button escrito Botao 5 na janela
 pack( ); //dimensiona a janela  para caber todos os componentes e layouts com
seus tamanhos ideais
```

```
 setVisible(true); //deixa janela visivel
 }
 static public void main(String[] args) { //cria metodo principal
 new LayoutGrade( );// instancia classe LayoutGrade
 }
}
```

3. Compare os programas das transparências 12, 13, 14, 15, 16, 18, respectivamente, com os das transparências 3, 4, 5, 6. 7, 8. Quais as diferenças? Escreva no máximo 3 linhas.

Todo componente awt tem um componente swing compatível, a maioria apenas adicionou um j no inicio exemplo: jFrame, JCheckbox, mas alguns mudaram o nome exemplo: Choice virou JComboBox, no swing não tem o método setLayout precisando importar o awt.

4. No programa da transparência 17 explique para que serve a classe ButtonGroup.

Serve para criar um grupo de botões de radio, um JRadioButton tem a característica de so poder selecionar um, então agrupamos os radio button para so podermos selecionar um. Se criamos dois radio button e não agrupamos eles é possível selecionar ambos pois eles não tem dependência, agora agrupados só será possível selecionar um.

5. Observe as transparências 18, 19 e 22 e descreva o que o objeto no parâmetro do método setLayout() faz na disposição dos componentes na janela. Para que servem as classes BorderLayout, GridLayout, FlowLayout.

O objeto no parâmetro do setLayout determina que tipo de layout terá na janela.
BorderLayout serve para a janela ter layout por localização (centro, norte, sul, leste, oeste).
GridLayout serve para a janela ter layout por grade com linhas e colunas.
FlowLayout é o layout padrão, ele coloca os componentes em linha, por padrão centralizado horizontalmente, com seus tamanhos ideais, se não tem mais espaço horizontal o FlowLayout utiliza mais linhas.

6. Observe o programa da transparência 21, para que a classe JPanel é utilizada?

Serve para criar painéis (containers dentro da janela), na transparência 21 foi criado um painel com 5 botões, esse painel foi colocado no leste da janela e no oeste foi colocado um outro botão.

7. Execute e leia o código do programa Psico.java. Explique, na sua visão, por que é usado Runnable. e não Thread na classe BotaoDoidao.

Pois BotaoDoidao já herda JButton e como so é possível herdar uma classe, então usa-se a interface Runnable invés da Classe Threads.

8. Para obter o máximo da execução do programa Swing.java, quando a janela dele estiver visível, pressione o botão "Botao 1" e continue respondendo os diálogos, note que as escolhas nos diálogos são apresentadas na saída padrão. Não se preocupe agora com o funcionamento do Botão, para o momento basta entender que quando ele for pressionado, o método "actionPerformed()" será executado. Analisando o código:

a) escreva uma linha que apresente um diálogo usando JOptionPane com a mensagem "Digite o seu nome" onde o usuário possa digitar um texto;

```
String inputName = JOptionPane.showInputDialog("Digite o seu nome");
```

b) retire o comentário de cada uma das linhas entre 89 até 92, apenas uma dessas quatro linhas deve estar sem comentário e execute o programa. O que você nota de diferença quando muda a linha sem comentário?

Muda o tema da janela. Cada linha tem um tema para o programa.