Raul Alexandre Gonzalez Augusto RA: 211023698

1-Encontre o método “SwingUtilities.invokeLater”, por que ele é usado?

Esta sendo utilizado como classe anônima, é usado para receber um evento de interface e enfileira-lo para ser despachado pela thread de eventos de interface quando possível.

2-A chamada do método “pBar.setValue(cont);” na linha 22, altera a representação gráfica da JProgressBar. Explique por que o esta chamada não precisa do método “SwingUtilities.invokeLater”.

Pois a atualização já esta sendo feita no actionPerformed e como não é algo demorado que va travar a thread nele, então não há a necessidade de usar o invokeLater.

3-Poderíamos executar “pBar.setValue(cont++);” da linha 31 sem usar o “SwingUtilities.invokeLater”? Explique.

Poderiamos, como não é uma operação exigente não vai travar a thread principal em seu processamento, porém para evitar que a janela trave é sempre bom utilizar invokeLater ou invokeAndWait quando necessário.

4-É possível substituir o “SwingUtilities.invokeLater” pelo método “SwingUtilities.invokeAndWait”? Apresente as alterações necessárias no programa. Qual a diferença entre estes métodos?

Sim, a diferença é que invokeLater é assíncrono e não bloqueia a chamada enquanto realiza o evento já o invokeAndWait é síncrono e bloqueia a chamada enquanto realiza o evento.

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.lang.reflect.InvocationTargetException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JProgressBar;

import javax.swing.SwingUtilities;

public class TestaThreadWait extends JFrame implements Runnable, ActionListener {

  JProgressBar pBar = new JProgressBar(JProgressBar.HORIZONTAL, 0, 10);

  JButton b = new JButton("Inicia");

  int cont = 0;

  TestaThreadWait( ) {

    super("Thread com Swing");

    setLayout(new FlowLayout());

    add(b);

    add(pBar);

    b.addActionListener(this);

    pack( );

    setVisible(true);

  }

  public void actionPerformed(ActionEvent e) {

    cont = 0;

    pBar.setValue(cont);

    new Thread(this).start();

  }

  public void run() {

    try {

      while (cont <= 10) {

        SwingUtilities.invokeAndWait(new Runnable() {

          public void run() {

            pBar.setValue(cont++);

          }

        });

        Thread.sleep(1000);

      }

    } catch (InterruptedException e) {}

      catch (InvocationTargetException er) {}

  }

  static public void main(String[] args) {

    new TestaThreadWait( );

  }

}

5-Após a compilação quais arquivos serão gerados?

TestaThread.class e TestaThread$1.class