Exemplo Threads com Swing:

Fazer uma aplicação Java com 2 JLabel, com um ponto final, em formatação Negrito, tamanho 70 e um JButton. A aplicação deve conter 2 Threads e um método. O método deve fazer o ponto dar uma volta completa em 13 passos, onde os 3 primeiros devem fazer o ponto descer 10 px por vez, os 4 seguintes devem fazer o ponto andar 20 px à direita, por vez, os 3 passos seguintes devem fazer o ponto subir 10 px por vez e, por fim, os 4 seguintes devem fazer o ponto andar 20 px à esquerda, retornando ao ponto de partida. Cada Thread deve chamar o método para movimentar um JLabel diferente. O JButton inicia a aplicação.

```
package view;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JButton;

import controller.BolinhaController;

public class Principal extends JFrame {
	private static final long serialVersionUID = 1L;
	private JPanel contentPane;

	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					Principal frame = new Principal();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	public Principal() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 450, 300);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		contentPane.setLayout(null);

		JLabel lblBolinha1 = new JLabel(".");
		lblBolinha1.setFont(new Font("Tahoma", Font.BOLD, 70));
		lblBolinha1.setBounds(58, 64, 68, 74);
		contentPane.add(lblBolinha1);

		JLabel lblBolinha2 = new JLabel(".");
		lblBolinha2.setFont(new Font("Tahoma", Font.BOLD, 70));
		lblBolinha2.setBounds(313, 64, 68, 74);
		contentPane.add(lblBolinha2);

		JButton btnIniciar = new JButton("Iniciar");
		btnIniciar.setBounds(0, 0, 89, 23);
		contentPane.add(btnIniciar);

		BolinhaController bolinhaController =
				new BolinhaController(lblBolinha1, lblBolinha2, btnIniciar);
		btnIniciar.addActionListener(bolinhaController);
	}
}
```

```java
package controller;

import java.awt.Rectangle;

import javax.swing.JButton;
import javax.swing.JLabel;

public class ThreadBolinha extends Thread {

	private JLabel lblBolinha;
	private JButton btnIniciar;

	public ThreadBolinha(JLabel lblBolinha, JButton btnIniciar) {
		this.lblBolinha = lblBolinha;
		this.btnIniciar = btnIniciar;
	}

	private void mexeBolinha() {
		btnIniciar.setEnabled(false);

		Rectangle posicao;
		posicao = lblBolinha.getBounds();
		lblBolinha.setBounds(posicao);
		int contadorDeMov = 0;
		while (contadorDeMov <= 12) {
			if (contadorDeMov <= 2) {
				posicao.y = posicao.y + 10;
			} else {
				if (contadorDeMov > 2 && contadorDeMov <= 6) {
					posicao.x = posicao.x + 20;
				} else {
					if (contadorDeMov > 6 && contadorDeMov <= 9) {
						posicao.y = posicao.y - 10;
					} else {
						if (contadorDeMov > 9 && contadorDeMov <= 12) {
							posicao.x = posicao.x - 20;
						}
					}
				}
			}
			lblBolinha.setBounds(posicao);
			try {
				Thread.sleep(500);
			} catch (InterruptedException ex) {
				ex.printStackTrace();
			}
			contadorDeMov++;
		}

		btnIniciar.setEnabled(true);
	}

	@Override
	public void run() {
		mexeBolinha();
	}



}
```

```java
package controller;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JLabel;

public class BolinhaController implements ActionListener{

	private JLabel lblBolinha1;
	private JLabel lblBolinha2;
	private JButton btnIniciar;

	public BolinhaController(JLabel lblBolinha1,
			JLabel lblBolinha2, JButton btnIniciar) {
		this.lblBolinha1 = lblBolinha1;
		this.lblBolinha2 = lblBolinha2;
		this.btnIniciar = btnIniciar;
	}

	private void botaoBolinha(){
		Thread t1 = new ThreadBolinha(lblBolinha1, btnIniciar);
		Thread t2 = new ThreadBolinha(lblBolinha2, btnIniciar);
		t1.start();
		t2.start();
	}

	@Override
	public void actionPerformed(ActionEvent e) {
		botaoBolinha();
	}

}
```

Exemplo com Threads Aninhadas:

Fazer uma aplicação, em Java, que tenha, na tela, uma JProgressBar, um JLabel e um JButton. A aplicação deve ter duas Threads. A primeira, fará com que a JProgressBar incremente, num intervalo de 20 mS. A segunda, enquanto a primeira estiver viva, irá alternando 3 palavras em um tempo de exibição constante, de 100 mS. O Botão deve iniciar a chamada das Threads e fazer com que ele próprio desapareça.

```
package view;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JProgressBar;
import javax.swing.JButton;

import controller.ProgressBarController;

public class Tela extends JFrame {
	private static final long serialVersionUID = 1L;
	private JPanel contentPane;

	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					Tela frame = new Tela();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

	/**
	 * Create the frame.
	 */

	public Tela() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(100, 100, 450, 300);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);
		contentPane.setLayout(null);

		final JLabel lblNewLabel = new JLabel(" ");
		lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 18));
		lblNewLabel.setBounds(10, 132, 98, 49);
		contentPane.add(lblNewLabel);

		final JProgressBar progressBar = new JProgressBar();
		progressBar.setBounds(10, 35, 188, 42);
		contentPane.add(progressBar);

		JButton btnIniciar = new JButton("In\u00EDcio");
		btnIniciar.setBounds(10, 228, 89, 23);
		contentPane.add(btnIniciar);

		ProgressBarController pbController =
				new ProgressBarController(lblNewLabel, progressBar, btnIniciar);
		btnIniciar.addActionListener(pbController);

	}
}
```

```java
package controller;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JProgressBar;

public class ProgressBarController implements ActionListener{

	private JLabel lblNewLabel;
	private JProgressBar progressBar;
	private JButton btnIniciar;

	public ProgressBarController(JLabel lblNewLabel, JProgressBar progressBar,
			JButton btnIniciar){
		this.lblNewLabel = lblNewLabel;
		this.progressBar = progressBar;
		this.btnIniciar = btnIniciar;
	}

	private void acaoBarra(){
		btnIniciar.setEnabled(false);
		Thread tBanner = new ThreadBanner(lblNewLabel, progressBar, btnIniciar);
		tBanner.start();
	}

	@Override
	public void actionPerformed(ActionEvent arg0) {
		acaoBarra();
	}

}
```

```java
package controller;

import javax.swing.JProgressBar;

public class ThreadProgressBar extends Thread {

	private JProgressBar progressBar;

	public ThreadProgressBar(JProgressBar progressBar) {
		this.progressBar = progressBar;
	}

	private void preencheBarra() {
		for (int i = 1; i <= 100; i++) {
			progressBar.setValue(i);
			try {
				Thread.sleep(20);
			} catch (InterruptedException ex) {
				ex.printStackTrace();
			}
		}
	}

	@Override
	public void run() {
		preencheBarra();
	}


}
```

```java
package controller;

import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JProgressBar;

public class ThreadBanner extends Thread {

	private JLabel lblNewLabel;
	private JProgressBar progressBar;
	private JButton btnIniciar;

	public ThreadBanner(JLabel lblNewLabel, JProgressBar progressBar,
			JButton btnIniciar) {
		this.lblNewLabel = lblNewLabel;
		this.progressBar = progressBar;
		this.btnIniciar = btnIniciar;
	}

	private void geraBanner() {
		btnIniciar.setEnabled(false);

		Thread tBarra = new ThreadProgressBar(progressBar);
		tBarra.start();

		int contador = 1;
		String texto = "";
		while (tBarra.isAlive()) {
			switch (contador) {
			case 1:
				texto = "Boa";
				break;
			case 2:
				texto = "Tarde";
				break;
			case 3:
				texto = "Galera";
			}
			lblNewLabel.setText(texto);

			contador++;
			if (contador == 4) {
				contador = 1;
			}
			try {
				Thread.sleep(100);
			} catch (InterruptedException ex) {
				ex.printStackTrace();
			}
		}
		btnIniciar.setEnabled(true);
	}

	@Override
	public void run() {
		geraBanner();
	}


}
```

Exercícios

Exercícios:

1) Fazer uma aplicação que rode 5 Threads que cada uma delas imprima no console o seu número.
2) Fazer uma aplicação que insira números aleatórios em uma matriz 3 x 5 e tenha 3 Threads, onde cada Thread calcula a soma dos valores de cada linha, imprimindo a identificação da linha e o resultado da soma.

Exercícios

Exercícios:

3) Fazer uma aplicação de uma corrida de sapos, com 5 Threads, cada Thread controlando 1 sapo. Deve haver um tamanho máximo para cada pulo do sapo (em metros) e a distância máxima para que os sapos percorram. A cada salto, um sapo pode dar uma salto de 0 até o tamanho máximo do salto (valor aleatório). Após dar um salto, a Thread, para cada sapo, deve mostrar no console, qual foi o tamanho do salto e quanto o sapo percorreu. Assim que o sapo percorrer a distância máxima, a Thread deve apresentar que o sapo chegou e qual sua colocação.

Exercícios

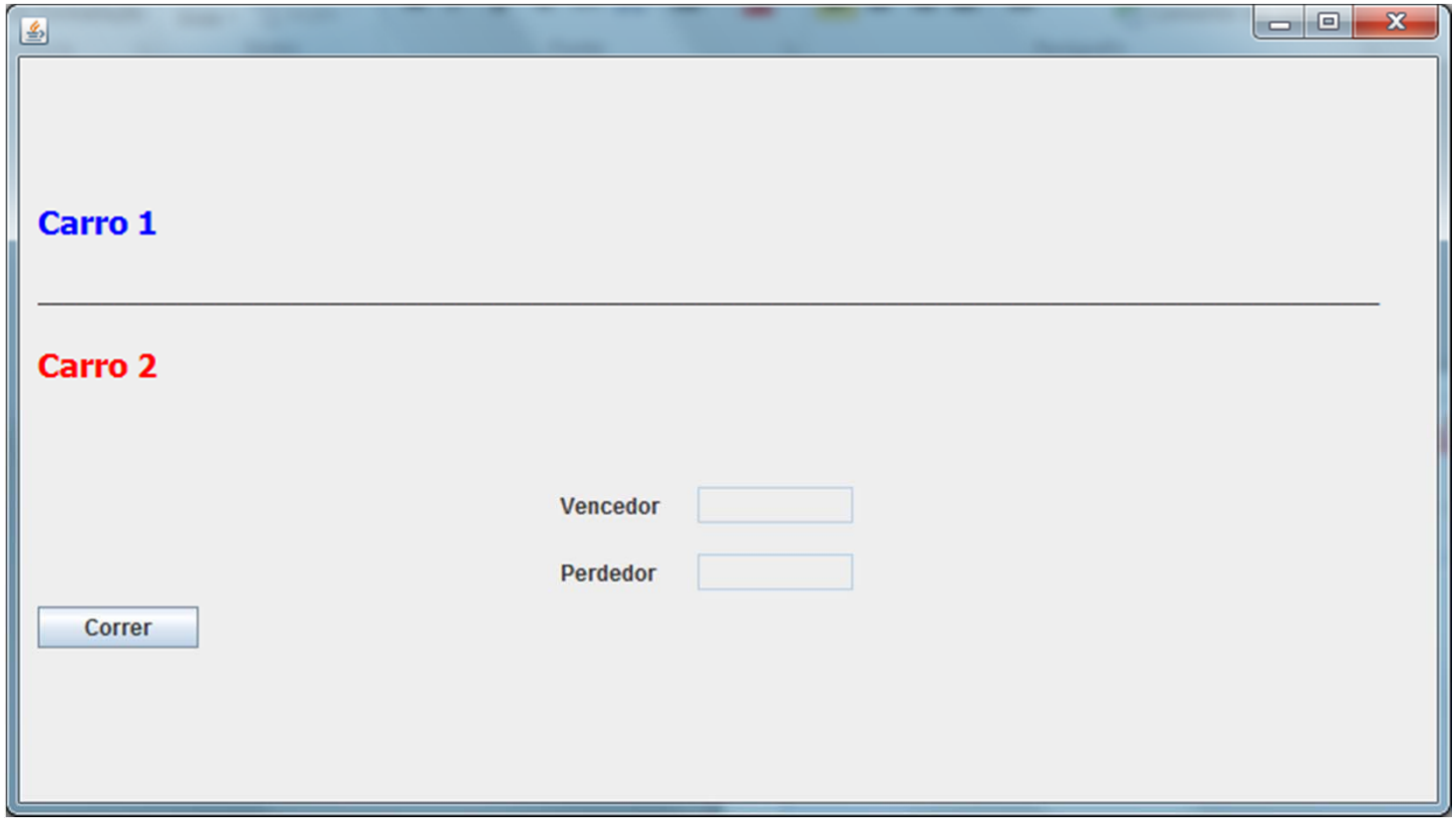
Exercícios:

4) Utilizando o Java SWING, criar uma tela, semelhante à tela abaixo, para criar uma corrida de carros, tipo *drag race*. A aplicação deve ter a distância que os carros devem correr e a velocidade máxima dos carros. Os carros (Jlabel) devem, a cada 100 mS, dar uma arrancada de velocidade que pode estar entre 0 e a velocidade máxima (definida aleatoriamente). Assim que o primeiro carro chegar, o JTextField Vencedor deve receber o nome deste e o JTextField Perdedor receberá o nome do outro carro. Assim que a corrida se inicia, o botão Correr deve sumir.

Exercícios

Exercícios

4)



Exercícios

Exercícios:

5) Fazer, com o Java SWING, uma aplicação de caça-níquel, conforme figura abaixo. O caça níquel tem 3 JTextFields, independentes, que giram, aleatoriamente, de 1 a 150 vezes e apresentará um número de 1 a 7. Quando iniciado, o botão Jogar deve desaparecer.