

Mãos à obra: Calculador rápido

Vamos executar a multiplicação em uma nova thread. O primeiro passo é criar a tarefa que implementa a interface `Runnable` :

1) Crie uma nova classe `TarefaMultiplicacao` no pacote `br.com.caelum.threads` :

```
package br.com.alura.threads;

public class TarefaMultiplicacao implements Runnable {

    @Override
    public void run() {
        //esse método o nosso thread executará
    }

}
```

2) Na classe `TarefaMultiplicacao` adicione os atributos para recuperar os dados da interface gráfica. Gere também o construtor:

```
private JTextField primeiro;
private JTextField segundo;
private JLabel resultado;

public TarefaMultiplicacao(JTextField primeiro, JTextField segundo,
JLabel resultado) {
    this.primeiro = primeiro;
    this.segundo = segundo;
    this.resultado = resultado;
}
```

3) Copie toda multiplicação da classe `AcaoBotao` para o método `run` da classe `TarefaMultiplicacao` :

```
@Override
public void run() {

    long valor1 = Long.parseLong(primeiro.getText());
    long valor2 = Long.parseLong(segundo.getText());
    BigInteger calculo = new BigInteger("0");

    for (int i = 0; i < valor1; i++) {
        for (int j = 0; j < valor2; j++) {
            calculo = calculo.add(new BigInteger("1"));
        }
    }

    resultado.setText(calculo.toString());
}
```

4) Na classe `AcaoBotao`, dentro do método `actionPerformed` instancie a tarefa e crie uma nova thread. Não esquece de chamar o método `start`:

```
public void actionPerformed(ActionEvent e) {  
  
    TarefaMultiplicacao tarefa = new TarefaMultiplicacao(primeiro, segundo, resultado);  
    Thread threadMultiplicador = new Thread(tarefa);  
  
    //thread começa a trabalhar  
    threadMultiplicador.start();  
}
```

5) Rode novamente a classe `TelaCalculador` e execute uma multiplicação demorada. A tela continua travando?