

Java - Swing:

Como controlar a ordem do “foco” nos componentes?

Diferente de outras linguagens de programação como Delphi ou mesmo o C#, em Java, controlar a ordem do foco nos componentes *swing* dá mais trabalho do que você possa imaginar.

Pois bem, para controlar o ordem do foco nos componentes em Java você poderá usar de diversos recursos da própria linguagem, dentre eles:

- no evento de perda de foco, `focusLost`, de cada componente você poderá requerer o foco para o próximo componente (`nomeDoProximoComponente.requestFocus();`) ou
- herdar e reescrever alguns métodos da classe `FocusTraversalPolicy`.

Nas pesquisas que fiz vi outras maneiras de se fazer a mesma coisa, no entanto, estas duas são as que achei mais fáceis de serem utilizadas. Sendo que, apesar da facilidade da primeira opção, a que mais me agradou foi a segunda. Inicialmente dá um certo trabalho para se criar a classe herdeira de `FocusTraversalPolicy`, mas você terá que fazer isto uma única vez. E depois de pronto, basta algumas linhas de código no método construtor de cada janela (`JFrame`), que você terá a ordem desejada. Desta maneira o código de orientação do foco ficará todo em um único lugar, deixando o programa mais limpo e mais fácil de ser entendido.

Bem, agora chega de blá... blá... blá... e vamos ao código:

- primeiro construa a classe herdeira de `FocusTraversalPolicy`. No meu caso eu dei o nome de `MyOwnFocusTraversalPolicy` (Figura 1), seguindo o padrão dos exemplos que achei na internet;
- depois da classe `MyOwnFocusTraversalPolicy` pronta, basta adicionar as seguintes linhas de código mostradas na Figura 2, no método construtor de cada janela (`JFrame`).

É isto! Mais um tutorial concluído.

Boa leitura e bons programas!

Programação Orientada a Objetos
Java – Swing – Controlando a Ordem do Foco nos Componentes da Tela
Prof. Edwar Saliba Júnior – versão 1.0

```
1 package pv_exercicioemdupla_02_n1_m01_a;
2
3 import java.awt.Component;
4 import java.awt.Container;
5 import java.awt.FocusTraversalPolicy;
6 import java.util.ArrayList;
7
8 public class MyOwnFocusTraversalPolicy extends FocusTraversalPolicy {
9
10     ArrayList<Component> order;
11
12     public MyOwnFocusTraversalPolicy(ArrayList<Component> order) {
13         this.order = new ArrayList<>(order.size());
14         this.order.addAll(order);
15     }
16
17     @Override
18     public Component GetComponentAfter(Container focusCycleRoot,
19         Component aComponent) {
20         int idx = (order.indexOf(aComponent) + 1) % order.size();
21         return order.get(idx);
22     }
23
24     @Override
25     public Component GetComponentBefore(Container focusCycleRoot,
26         Component aComponent) {
27         int idx = order.indexOf(aComponent) - 1;
28         if (idx < 0) {
29             idx = order.size() - 1;
30         }
31         return order.get(idx);
32     }
33
34     @Override
35     public Component getDefaultComponent(Container focusCycleRoot) {
36         return order.get(0);
37     }
38
39     @Override
40     public Component getLastComponent(Container focusCycleRoot) {
41         return order.get(order.size() - 1);
42     }
43
44     @Override
45     public Component getFirstComponent(Container focusCycleRoot) {
46         return order.get(0);
47     }
48 }
```

Figura 1: Classe MyOwnFocusTraversalPolicy

```
11 public class FazUmCalculoQualquer extends javax.swing.JFrame {
12
13     private GerenciaPessoas gp;
14     private int dia;
15     private int mes;
16     private int ano;
17
18     public FazUmCalculoQualquer() {
19         initComponents();
20         gp = new GerenciaPessoas();
21
22         dia = 0;
23         mes = 0;
24         ano = 0;
25
26         ArrayList<Component> order = new ArrayList<>(7);
27         order.add(tfDia);
28         order.add(tfMes);
29         order.add(tfAno);
30         order.add(tfNome);
31         order.add(tfEndereco);
32         order.add(tfTelefone);
33         order.add(tfSalario);
34         order.add(tfIdade);
35         MyOwnFocusTraversalPolicy newPolicy =
36             new MyOwnFocusTraversalPolicy(order);
37         setFocusTraversalPolicy(newPolicy);
38     }
```

Figura 2: Método construtor da classe FazUmCalculoQualquer

Basicamente:

- Cria-se e instancia-se um `ArrayList<Component>`,
- adiciona-se ao `ArrayList` os componentes na ordem que você quer que o foco percorra a tela (Ex.: `tfDia`, `tfMes`, ...),
- cria-se um objeto da classe `MyOwnFocusTraversalPolicy` passando como parâmetro ao método construtor desta classe, o `ArrayList` criado anteriormente e já preenchido (`order`) e
- por fim, chama-se o método “`setFocusTraversalPolicy`” da janela em questão e passa-se como parâmetro o objeto do tipo “`MyOwnFocusTraversalPolicy`” criado, ou seja, “`newPolicy`”.

Obs.: Todos estes passos têm que ser feitos no método construtor da janela que se quer organizar o foco.

Bibliografia

GUJ.com.br. **Como setar o próximo componente a receber foco?**.

<<http://www.guj.com.br/java/254661-como-setar-o-proximo-componente-a-receber-foco>> Acesso em: 22 nov. 2013

Oracle. **Introduction to the Focus Subsystem.**

<<http://docs.oracle.com/javase/tutorial/uiswing/misc/focus.html>> Acesso em: 22 nov. 2013

JAVAA. **Controlling the focus when navigating with the TAB and shift-TAB keys Java code example.** <<http://www.javafaq.nu/java-example-code-786.html>> Acesso em: 22 nov. 2013

PERERA, Gayan. **Controlling TabIndex in Java Swing.**

<<http://gaps-blog.blogspot.com.br/2008/12/controlling-tabindex-in-java-swing.html>> Acesso em: 22 nov. 2013