

PROGRAMAÇÃO ORIENTADA A OBJETOS

Eventos e GUIs

Prof. Emanuel Barreiros



Introdução ao tratamento de eventos

- Geralmente o usuário interage com a aplicação dando ordens sobre o que ela deve fazer
- Por exemplo, após digitar um email, pressionar o botão “Enviar” representa a ordem do usuário para que a aplicação envie o email
- Deste modo, GUIs são orientadas a eventos
- Quando o usuário interage com um componente GUI, a interação (o evento) leva o programa a fazer algo
- Este conceito é chamado de tratamento de eventos

Exemplo

- Vamos dar uma olhada na classe TextFieldFrame que implementamos...

Interfaces

- Interfaces definem e padronizam a maneira como coisas interagem
- Por exemplo, botões em um rádio permitem que o usuário interaja de alguma maneira com ele
- Para usar uma interface, uma classe deve “dizer” que a implementa
- Uma interface possui apenas métodos abstratos e pode também possuir constantes (valores que não podem ser mudados)

Interfaces no Java

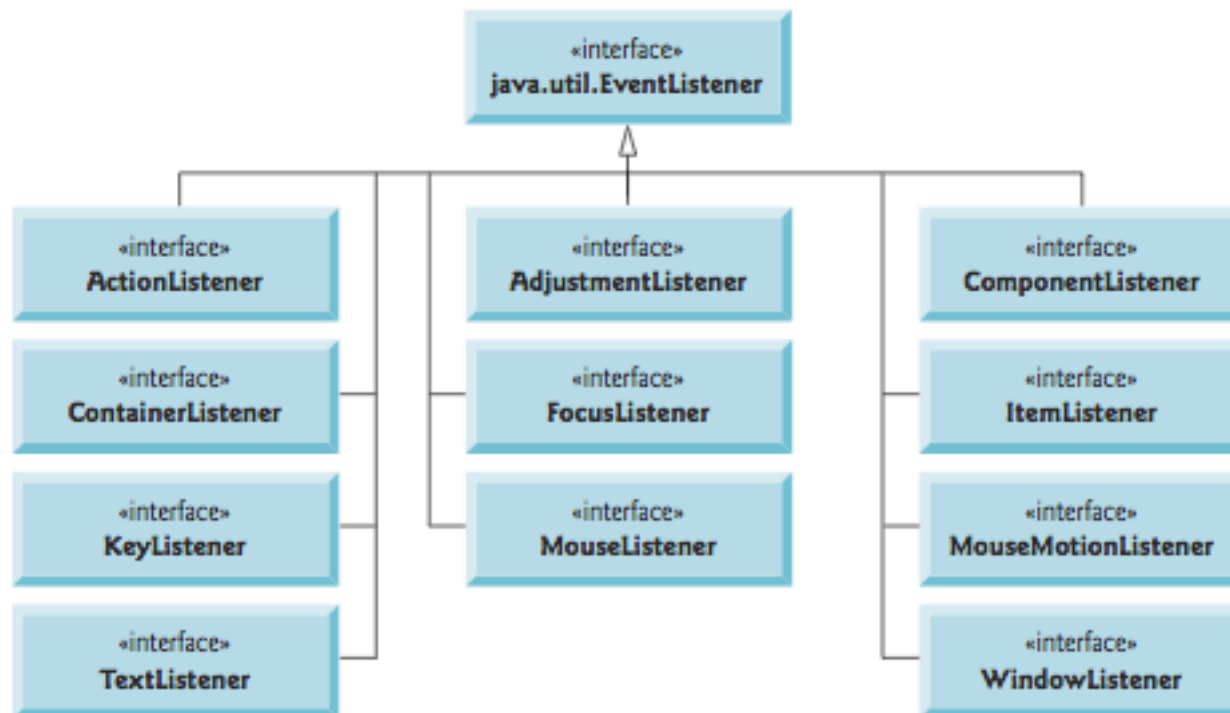
- Elas aparecem por todos os lados. Vamos tomar como exemplo a interface Comparable
- Java fornece vários operadores para comparação (<, >, <=, etc.) mas objetos não podem ser comparados utilizando-os
- A interface Comparable é utilizada para permitir que objetos sejam comparáveis através de um método definido por ela: compareTo(..)
- Uma classe que implemente essa interface é obrigada a fornecer implementação para o compareTo(..), o que torna objetos deste tipo comparáveis!

O Mecanismo de tratamento de eventos

- Existem 3 entidades que participam do mecanismo
 - Origem do evento
 - Componente (um objeto) com o qual o usuário interage
 - Objeto do evento
 - Encapsula a informação sobre o evento que ocorreu, como a referência para a fonte, e qualquer outra informação necessária para que o tratador consiga tratar o evento
 - Listener do evento
 - O objeto que é notificado pelo objeto de origem do evento, ou seja, ele fica escutando até que o evento seja “disparado”. O listener utiliza o objeto do evento para tratar o evento

Tipos de eventos

- Em geral, para cada tipo de evento, existe uma interface



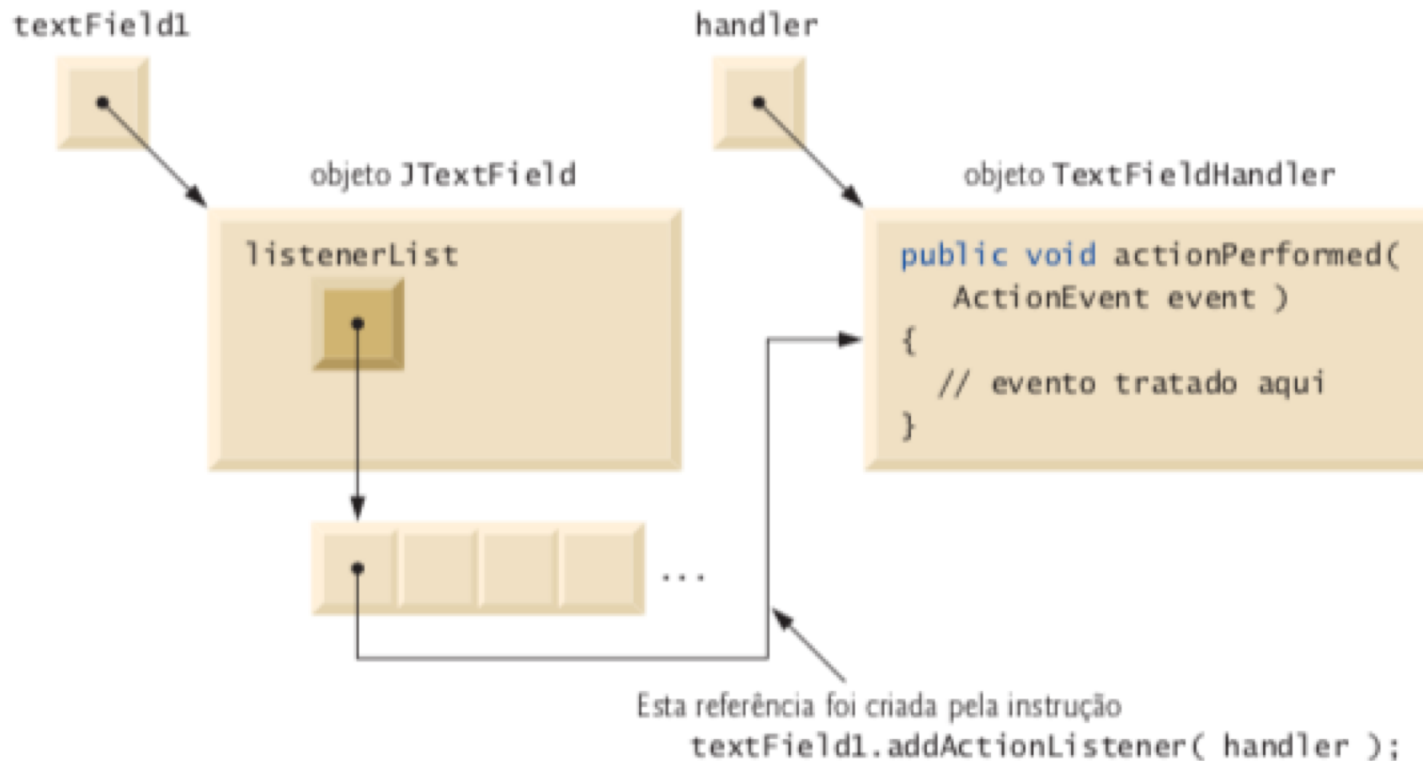
Como funciona o tratamento de eventos?

- Devemos nos registrar para sermos notificados de eventos, como pode ser visto na linha abaixo:
 - `botaoNormal.addActionListener(handler);`
- Todo componente possui vários eventos
- Cada tipo de evento possui uma interface para o listener do evento:
 - `ActionEvent` é tratado por um `ActionListener`
 - `MouseEvent` é tratado por um `MouseListener` e `MouseMotionListener`
 - `KeyEvent` é tratado por um `KeyListener`

Como funciona o tratamento de eventos? (cont.)

- Quando um evento ocorre, o componente GUI recebe (da JVM) um ID do evento especificando o tipo do evento
- O componente usa esse ID para “decidir” qual evento deve ser disparado e qual método deve ser chamado em cada listener (objeto cadastrado para receber a notificação)
 - EX: para um `ActionEvent`, todos os objetos `ActionListener` são notificados através do método `actionPerformed`
- Vamos ver como funciona para um botão no código!
 - `ButtonFrame` e `ButtonTest`

Como funciona o tratamento de eventos? (cont.)



Exemplo

- Mais um exemplo de código: JCheckBox
 - CheckBoxFrame e CheckBoxTest

Exemplo

- Tratando eventos de mouse!
 - PaintPanel e Painter

Exercício

- Vamos colocar a mão na massa e desenvolver uma calculadora
- A calculadora é simples, mas vai precisar utilizar o conhecimento de interface que vimos até agora. Quero algo bem simples como a figura abaixo:
 - Inicialmente pode fazer apenas com inteiros, e se conseguir adicione o suporte a números decimais
 - Quem conseguir incrementar mais vai ser muito bem vindo 😊

Mãos à obra!

