Interface gráfica de usuário

FURB
UNIVERSIDADE DE BLUMENAU

Introdução

- Uma **interface gráfica de usuário** (GUI – *Graphics User Interface*) permite ao usuário interagir com a aplicação utilizando ícones e desenhos, ao contrário de interfaces baseadas apenas em texto
  - Fornecem apresentação mais amigável ao usuário.

- GUIs podem ser construídas reutilizando componentes de bibliotecas gráficas.
  - Os componentes também são conhecidos por “controles GUI” ou simplesmente “controles”;

- As principais bibliotecas nativas do Java para construção de GUIs são:
  - AWT
  - Swing
  - JavaFX

AWT

AWT – Design e comportamentos distintos entre os sistemas operacionais

Alguns componentes GUI

Classe do Componente
JLabel
JTextField
JButton
JCheckBox
JComboBox
JList
JPanel
JTable

Programação orientada a eventos

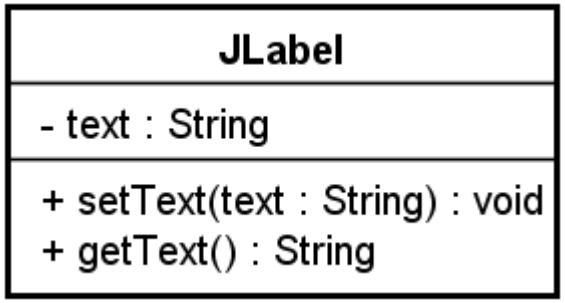
- É um estilo de programação na qual o fluxo de execução do programa é determinado por eventos

- Um evento é um sinal recebido pelo programa indicando que algo aconteceu

- São considerados eventos:
  - **Ações do usuário: movimentos do mouse, teclado**
  - Mensagens de outros programas
  - Periféricos enviando sinais ao programa

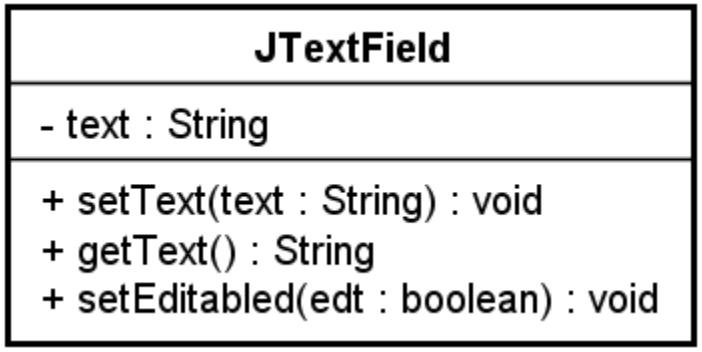
- O programa pode escolher responder ou ignorar o evento

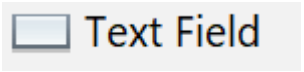
JLabel



Método	Descrição
`getText()`	Obtém o valor exibido no componente
`setText()`	Altera o valor exibido pelo componente

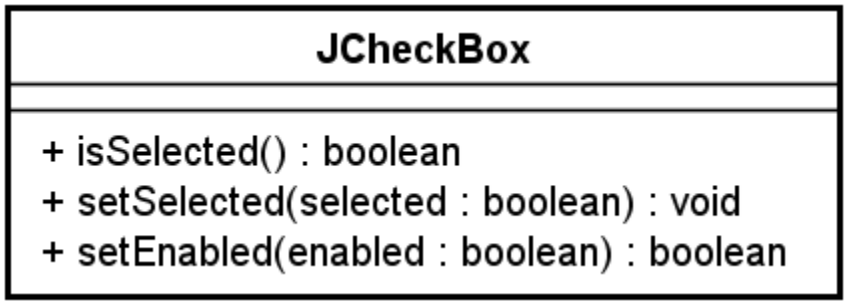
JTextField

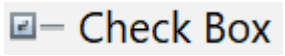




Método	Descrição
`getText()`	Obtém o valor exibido no componente
`setText()`	Altera o valor exibido pelo componente
`setEditabled()`	Habilita ou desabilita a edição

JCheckBox





Método	Descrição
`isSelected()`	Retorna true se o componente estiver marcado
`setSelected()`	Marca ou desmarca a caixa de seleção
`setEnabled()`	Habilita ou desabilita a edição

Arquitetura de software em camadas

FURB
UNIVERSIDADE DE BLUMENAU

Arquitetura de software

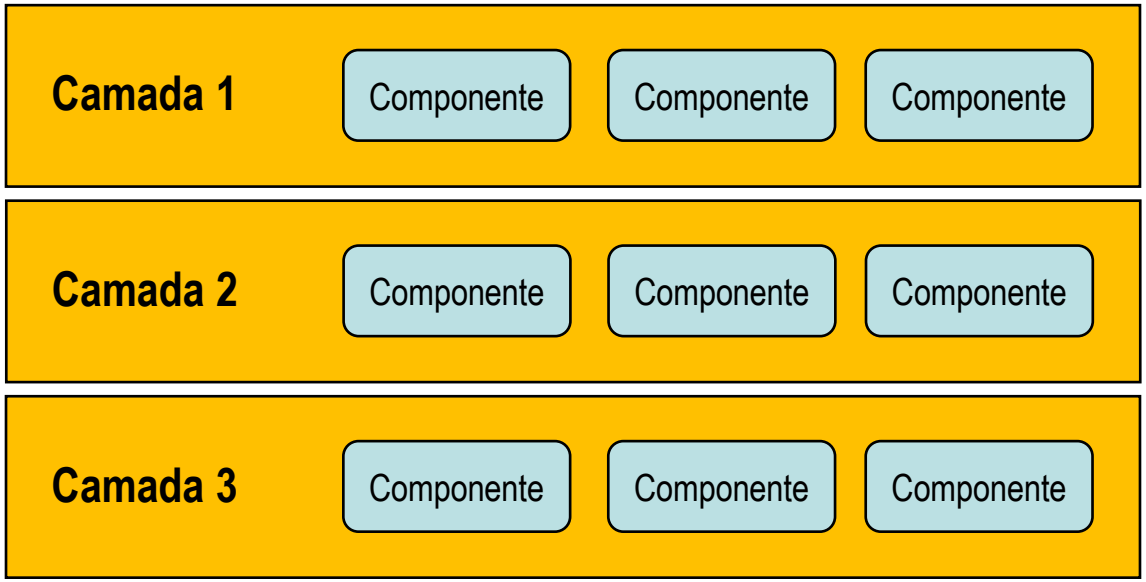
- A arquitetura de software descreve a forma como os componentes que compõe o software se relacionam.

- A arquitetura de software é “a organização fundamental de um software materializada pelos seus componentes, seus relacionamentos e o ambiente, e os princípios que regem a sua concepção e evolução” [1].

- Em OOP, é popular a arquitetura conhecida como “arquitetura em camadas”

Arquitetura em camadas

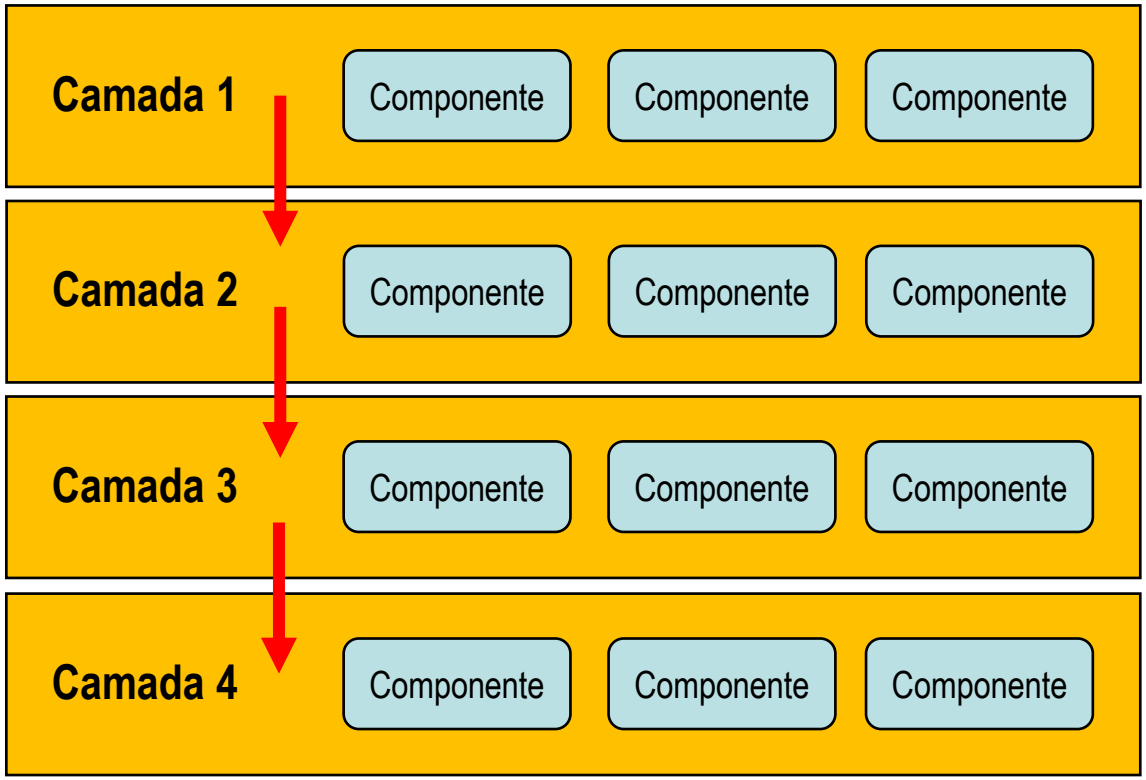
- Na “arquitetura de software em camadas” o software é organizado em camadas horizontais.
- Cada camada possui uma função específica na aplicação.



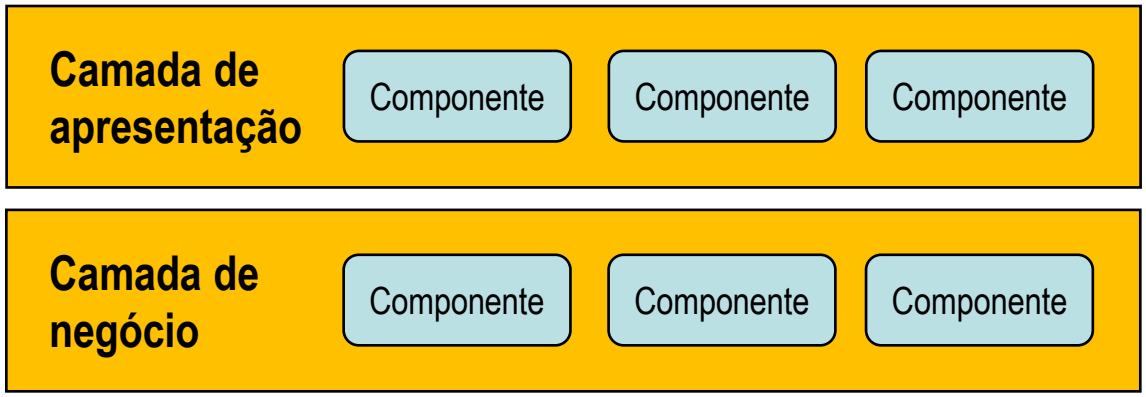
- Não há um número determinado de camadas

Arquitetura em camadas

- Em geral, as camadas são fechadas de forma que a camada conhece apenas a camada imediatamente inferior.

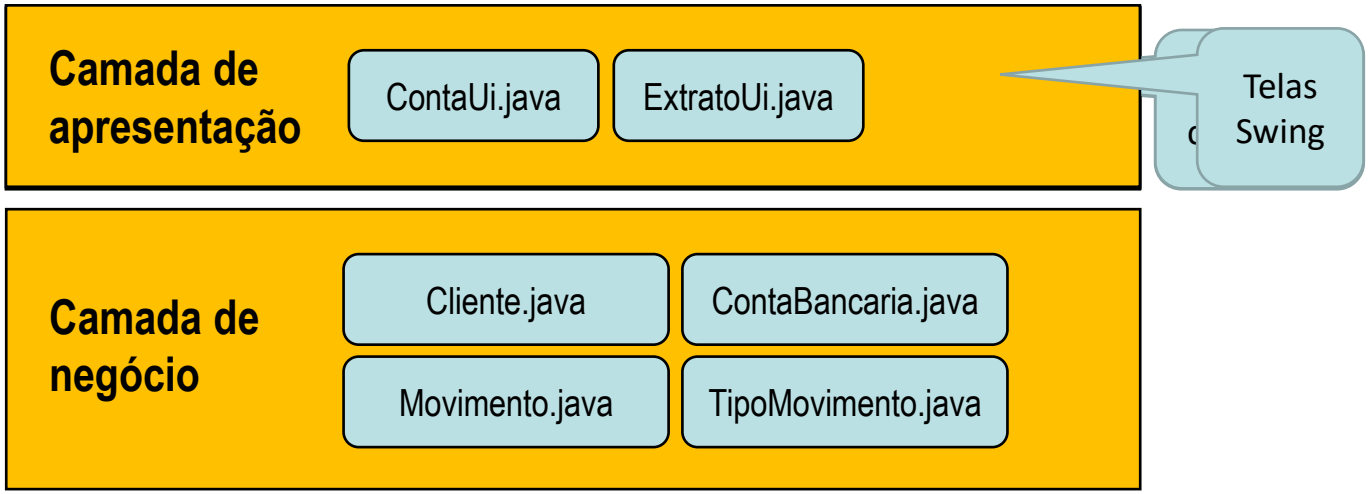


Exemplo de arquitetura em duas camadas



- Camada de apresentação
  - Responsável por se comunicar com o usuário, exibindo informações e lidando com as ações do usuário.
- Camada de negócio
  - Lida com as regras de negócio.
  - Desconhece como funciona a camada de apresentação

Exemplo de arquitetura em camadas



Vantagens da arquitetura em camadas

- Modularização
  - Princípio denominado “separação de conceitos”.
  - Cada camada mantém o foco num aspecto.
    - Os componentes dentro de uma camada se preocupam com a lógica pertinente àquela camada.

- Torna fácil desenvolver, testar e manter a aplicação

Organização de projetos Java

- Alguns desenvolvedores organizam as classes Java em pacotes de acordo com as camadas da arquitetura.
- Exemplo: