Técnicas de Programação

Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

Material Teórico

Construção de projeto em Java

Responsável pelo Conteúdo:
Prof. Ms. Douglas Almendro

Revisão Textual:
Prof. Ms. Fatima Furlan

UNIDADE Construção de projeto em Java

- O que é um pacote (package) em Java?
- Interfaces Gráficas

Objetivo de APRENDIZADO

- Nesta unidade, construiremos nosso primeiro projeto em linguagem Java.
- Vamos ver os detalhes da estrutura de um projeto. Recomendamos que fique atento ao material teórico e ao tutorial anexado. Todos os detalhes devem ser seguidos passo a passo como descrito no tutorial.
- Esperamos que você tenha um excelente aprendizado desse conteúdo que é muito importante para o curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

Atenção

Para um bom aproveitamento do curso, leia o material teórico atentamente antes de realizar as atividades. É importante também respeitar os prazos estabelecidos no cronograma.

Contextualização

Nesta unidade, construiremos um projeto na ferramenta NetBeans. Recomendamos que você siga atentamente o nosso tutorial de criação de projetos e pacotes.

Vamos abordar cada tipo de arquivo ou pacote a ser inserido dentro de um projeto. Com a criação de projeto e ou pacotes teremos como organizar melhor a estrutura de um programa de computador.

Durante o curso, veremos que existem várias situações nas quais podemos usar os conceitos de Pacotes e Arquivos. Veremos também todas as outras funcionalidades desta ferramenta. Esse material teórico servirá para sua vida como analista / programador.

O que é um pacote (package) em Java?

Um pacote em Java nada mais é que um agrupamento de classes em um diretório.

Usamos pacotes para organizar melhor nossas classes e certificar que classes diferentes, mas com o mesmo nome, possam ser usadas.

Os tipos que fazem parte da plataforma Java se encontram dentro de vários pacotes. Por exemplo, as classes principais estão no pacote java.lang, classes para leitura e escrita estão no pacote java.io e assim por diante.

Convenção de nomenclaturas de packages

Segundo a Oracle, os nomes de pacotes devem ser escritos em letras minúsculas para evitar conflito com os nomes de classes ou interfaces.

As empresas usam o seu nome de domínio Internet invertida para começar seus nomes para o pacote, por exemplo, com.example.mypackage é o nome de um pacote chamado mypackage criado por um programador em example.com .

Criando um Pacote

Para criar um pacote você deve inserir a declaração package seguido do nome de seu pacote. A declaração do pacote deve ser a primeira linha no arquivo da classe. Só pode haver uma instrução para cada classe.

Exemplo:

```
package br.com.empresa.banco;
public class Banco {
  public String nome;
  public Cliente clientes[] = new Cliente[2];
}
```

Usando o Pacote

Para termos acesso à classe, devemos referenciar todo o nome do pacote antes da classe. Esse método é conhecido como Fully Qualified Name , que seria o caminho completo da classe.

Exemplo:

```
class TesteDoBanco {
  public static void main(String args[]) {
    br.com.caelum.banco.Banco meuBanco = new br.com.caelum.banco.Banco();
    meuBanco.nome = "Banco do Brasil";
    System.out.println(meuBanco.nome);
  }
}
```

Você também pode importá-la com a seguinte instrução: import.

Exemplo:

```
import br.com.caelum.banco.Banco;
class TesteDoBanco {
  public static void main(String args[]) {
    Banco meuBanco = new Banco();
    meuBanco.nome = "Banco do Brasil";
  }
}
```

Interfaces Gráficas

A interface gráfica é que parte de um programa que interage com o usuário. Interfaces gráficas podem assumir muitas formas. Essas formas variam em complexidade de interfaces de linha de comando simples para as interfaces gráficas "touch" fornecidas por muitas aplicações modernas.

Segundo Deitel (2005), uma Interface Gráfica tem como objetivo oferecer ao usuário uma interação amigável com aplicativos. Esta dá ao aplicativo uma aparência e comportamentos distintos.

GUI no Java

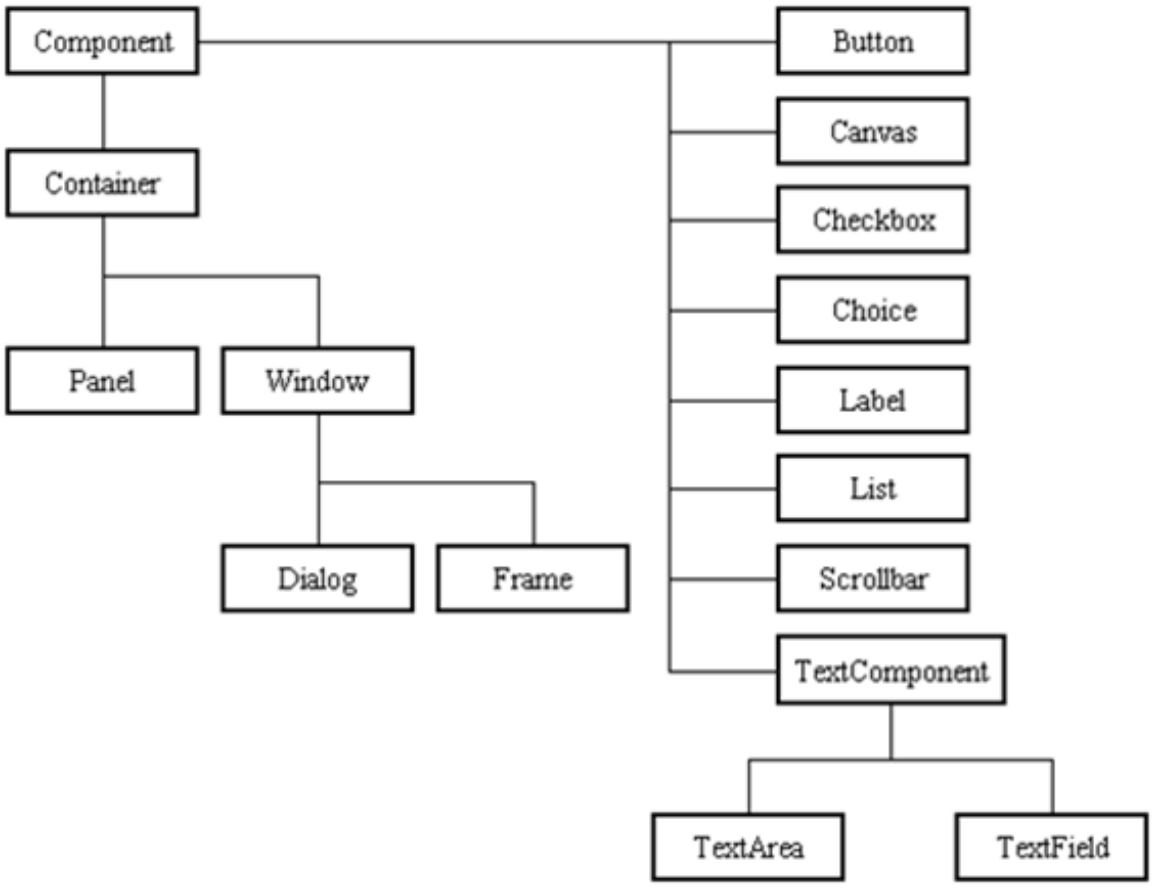
Java Foundation Classes (JFC) é um conjunto de componentes (GUI) da interface gráfica de usuário para aplicações Java que simplificam software. JFC contém o Abstract Window Toolkit (AWT), Java 2D e Swing.

AWT

A AWT foi a primeira API gráfica do Java, foi projetada para fornecer um conjunto comum de ferramentas para design de interface gráfica ao usuário que trabalha em uma variedade de plataformas. Os elementos da interface do usuário fornecidos pelo AWT são implementados usando kit de ferramentas GUI nativa de cada plataforma, preservando, assim, a aparência de cada plataforma. Este é um dos pontos mais fortes do AWT. A desvantagem dessa abordagem é o fato de que uma interface gráfica projetada em uma plataforma pode parecer diferente quando exibida em outra plataforma.

Uma interface gráfica de usuário é construída com elementos gráficos chamados de componentes. Os componentes típicos incluem itens como botões, barras de rolagem e campos de texto. Os componentes permitem ao usuário interagir com o programa e fornecendo um feedback visual sobre o estado do programa.

Estrutura da API



Containers

Containers é o local que abriga os componentes. Um Container proporciona um espaço onde um componente pode ser localizado. Na AWT, todos os Containers são instâncias da Classe Container ou um de seus subtipos.

A AWT fornece quatro classes de Containers, são elas:

- Panel;
- Windows;
- Dialog;
- Frame.

Componentes AWT

- Label (etiqueta/texto não editável);
- Button (botão);
- TextArea (área de texto);
- TextField (caixa de texto);
- CheckBox (Caixa de checagem).

Swing

Swing é um kit de ferramentas GUI avançado. Possui um rico conjunto de componentes que vão dos componentes básicos, como botões, etiquetas (labels), barras de rolagem até os mais avançados, como árvores (tree´s) e tabelas. O Swing é escrito em Java.

Como visto anteriormente, A API Swing é uma parte do JFC, Java Foundation Classes. É uma coleção de pacotes para a criação de aplicativos de desktop. JFC consiste em AWT, Swing, Accessibility, Java 2D e Drag and Drop. Swing foi lançado em 1997 com o JDK 1.2. É um conjunto de ferramentas bem maduro.

As principais características do conjunto da API Swing são:

- independente de plataforma;
- customizáveis;
- extensível;
- configurável;
- leve.

A API Swing tem 18 pacotes públicos:

- javax.accessibility
- javax.swing
- javax.swing.border
- javax.swing.colorchooser
- javax.swing.event
- javax.swing.filechooser
- javax.swing.plaf
- javax.swing.plaf.basic
- javax.swing.plaf.metal
- javax.swing.plaf.multi
- javax.swing.plaf.synth
- javax.swing.table
- javax.swing.text
- javax.swing.text.html
- javax.swing.text.html.parser
- javax.swing.text.rtf
- javax.swing.tree
- javax.swing.undo

Swing – Eventos

Os eventos são uma parte importante em qualquer programa de GUI. Todas as aplicações GUI são orientadas a eventos. Uma aplicação reage a diferentes tipos de eventos que são gerados durante seu ciclo de vida. Os eventos são gerados principalmente pelo usuário de um aplicativo, mas também podem ser produzidos por outros meios como , por exemplo, conexão à Internet, gerenciador de janelas, timer etc.

Tipos de Eventos

Os eventos podem ser classificados em duas categorias:

- **Eventos de primeiro plano (Foreground Events) -** Esses eventos requerem a interação direta do usuário. São gerados como consequência de uma pessoa interagindo com os componentes gráficos na interface gráfica do usuário. Por exemplo, ao clicar em um botão, mover o mouse, entrar em um personagem através do teclado, selecionando um item da lista, rolar a página etc.
- **Eventos de Segundo Plano (Background Events) -** Esses eventos não requerem a interação do usuário final. São gerados por interrupções do sistema operacional, falha de hardware ou software, um timer ou uma conclusão da operação etc.

Manipulando Eventos

Manipulação de eventos (Event Handling) é o mecanismo que controla o evento e decide o que deve acontecer se ocorrer um evento. Esse mecanismo tem o código que é conhecido como manipulador de eventos que é executado quando um evento ocorre. Java usa o modelo de evento Delegado*** (Event Delegation) para manipular os eventos. Este modelo define o mecanismo padrão para gerar e manipular eventos. Vejamos uma breve introdução a este modelo.

O modelo de evento Delegado* (*Event Delegation*) tem as seguintes propriedades-chave:**

- **Fonte (Source)-** A fonte é um objeto no qual o evento ocorre. Ela é responsável pela prestação de informações de que o evento ocorreu.
- **Ouvinte (Listener)**- É também conhecido como um evento Handler Listener (Manipulador Ouvinte). Ele é responsável pela geração de resposta a um evento. Do ponto de vista da implementação Java o ouvinte também é um objeto. Ouvinte espera até receber um evento. Uma vez que o evento é recebido, o ouvinte processa e retorna um evento.

A vantagem dessa abordagem é que a lógica de interface do usuário é completamente separada da lógica que gera o evento. O elemento de interface com o usuário é capaz de delegar o processamento de um evento para o pedaço de código. Neste modelo, o Ouvinte (Listener) precisa ser registrado com o objeto de origem para que possa receber a notificação de eventos. Esta é uma maneira eficiente de lidar com o caso porque as notificações de eventos são enviadas apenas para aqueles ouvintes que querem recebê-los.

Tutoriais

Instalando o Ambiente JAVA

http://arquivos.cruzeirodosulvirtual.com.br/materiais/disc_graduacao/2014/1sem/2mod/tec_pro/un_I/tutorial1.pdf

Pacotes - Packages JAVA

http://arquivos.cruzeirodosulvirtual.com.br/materiais/disc_graduacao/2014/1sem/2mod/tec_pro/un_I/tutorial2.pdf

Material Complementar

Para saber mais sobre o conteúdo estudado na Unidade, acesse:

Arquivo Tutorial de Criando o Projeto "pacotes Tut"

» http://docs.oracle.com/

Referências

DEITEL, Paul J.; DEITEL, Harvey M. **Java, como programar.** 6. ed. São Paulo: Pearson Education do Brasil, 2005

http://docs.oracle.com/

Anotações


Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo SP Brasil
Tel: (55 11) 3385-3000

Universidade Cruzeiro do Sul

UNICID Universidade Cidade de S. Paulo

UNIFRAN Universidade de Franca

UDF Centro Universitário

Módulo Centro Universitário