Universidade de Brasília Faculdade UnB Gama

Orientação a Objetos

Aula 9 - Janelas de Diálogo e Classes Empacotadoras

Daniel Porto

daniel.porto@unb.br

APRESENTAÇÃO

Janelas de Diálogo

Classes Empacotadoras

JANELAS DE DIÁLOGO

Geralmente, elaborar aplicativos necessita de comunicação entre o sistema e o usuário, em que algumas vezes o usuário fornece algum tipo de informação para que o sistema prossiga corretamente em sua execução.

Esse tipo de rápida interação é realizada através de caixas ou janelas de dialogo, sendo 4 abordadas neste material. A implementação da **Swing** fornece uma classe de fácil uso para este tipo de interação, chamada JOptionPane.

A JOptionPane possibilita uma variação de janelas de dialogo, por meio de métodos estáticos específicos, tais como:

MessageDialog, ConfirmDialog, OptionDialog, InputDialog

JANELAS DE DIÁLOGO

As características de cada uma das janelas de dialogo implementada pela Swing serão abordadas a seguir, porém uma importante propriedade, no ambiente de janelas (gráfico) consiste em:

- **modal** - não permite o acesso a outras janelas, enquanto esta permanecer em execução na aplicação. Exemplo: a necessidade da identificação do sexo do usuário para que o sistema possa prosseguir adequadamente em sua execução;
- **Não modal** (sem modo) - permite o acesso a janela não modal ou a outra janela qualquer em seu computador. Exemplo: janela como barra de ferramentas.

A janela de dialogo pode ser ou não modal, geralmente sendo modal, como será abordado neste material.

JANELAS DE DIÁLOGO

A partir de métodos da JOptionPane são acionadas 4 formas de dialogos diferentes, sendo cada uma destas janelas compostas por:

a) Título na janela de dialogo
b) Mensagem destinada ao contato com usuário
c) Ícone representativo do tipo de mensagem
    1. Informação
    2. Pergunta
    3. Alerta (ou advertência)
    4. Erro
    5. Um outro definido pelo usuário
d) Um ou mais botões de interação com usuário

Também é possível a ausência de qualquer ícone, sendo apresentada somente uma mensagem orientadora.

JANELAS DE DIÁLOGO

MessageDialog

Mostra somente uma informação para o usuário.

Não retorna nenhum valor após sua utilização.

Sua sintaxe geral consiste em:

JOptionPane.showMessageDialog(Componente, Mensagem, Título da Mensagem, Tipo de Mensagem)

- Componente: referente ao objeto do tipo contêiner que permite definir a posição na tela em que a janela de dialogo aparecerá. Normalmente seu valor é null para que a janela se apresente no centro da tela
- Mensagem: mensagem que a janela mostrará ao usuário
- Título da Mensagem: título que será mostrado na janela
- Tipo de Mensagem: defini o ícone padrão que será apresentado na janela de dialogo, ou um outro ícone qualquer definido no sistema, ou ainda sem ícone algum

JANELAS DE DIÁLOGO

Os ícones padrões são indicados pelas constantes:

- INFORMATION_MESSAGE 
- ERROR_MESSAGE 
- WARNING_MESSAGE 
- QUESTION_MESSAGE 
- PLAIN_MESSAGE (sem ícone)

Para cada um dos tipos de dialogo existe também um método que permite trocar o ícone padrão por um outro ícone qualquer, definido pelo desenvolvedor do programa.

JANELAS DE DIÁLOGO

ConfirmDialog

Mostra uma mensagem e alguns botões pré-definidos (Yes, No, OK, Cancel), solicitando a confirmação do usuário.

Retorna um valor inteiro para identificação de qual botão foi pressionado (escolhido) pelo usuário.

Várias são as opções de indicação de botões para esta janela de dialogo, sendo as mais comumente usadas:

- DEFAULT_OPTION
- YES_NO_OPTION
- YES_NO_CANCEL_OPTION
- OK_CANCEL_OPTION

Conforme o botão pressionado é retornado um valor inteiro que inicia em zero, do botão mais a esquerda, e segue sequencialmente para 1 e 2 dos botões vindo para direita.

JANELAS DE DIÁLOGO

JOptionPane.showConfirmDialog(Componente, Mensagem, Título da Mensagem, Botões, Tipo de Mensagem)

- Componente: referente ao objeto do tipo contêiner que permite definir a posição na tela em que a janela de dialogo aparecerá. Geralmente é deixado como null para janela ser apresentada no centro da tela
- Mensagem: mensagem que a janela mostrará ao usuário
- Título da Mensagem: título que será mostrado na janela
- Botões: constante que defini os botões que estarão presente nesta janela de dialogo
- Tipo de Mensagem: defini o ícone padrão que será apresentado na janela de dialogo, ou um outro ícone definido no sistema e ainda sem ícone algum

JANELAS DE DIÁLOGO

InputDialog

Solicita uma informação ao usuário que deverá responder com um texto (String) na própria janela de dialogo.

Retorna um valor String para ser utilizado.

JOptionPane.showInputDialog(Componente, Mensagem, Título da Mensagem, Tipo de Mensagem)

- Componente: referente ao objeto do tipo contêiner que permite definir a posição da janela de dialogo na tela sendo comumente definido como null para esta janela ser mostrada no centro da tela
- Mensagem: mensagem orientadora para o usuário
- Título da Mensagem: título da janela de dialogo
- Tipo de Mensagem: defini o ícone que será mostrado nesta janela, podendo ser os padrões, definido pelo usuário ou nenhum ícone (só a mensagem)

JANELAS DE DIÁLOGO

InputDialog

Algumas situações especiais são possíveis na interação com esta janela de dialogo, sendo elas:

- Pressionar o botão OK sem fornecer nenhuma resposta atribuirá vazio (””) na String de resposta
- Pressionar o botão Cancel sem fornecer nenhuma resposta atribuirá nulo (null) na String de resposta
- Pressionar o botão OK com alguma resposta atribui o valor informado na String indicada como resposta
- Pressionar o botão Cancel com alguma resposta atribuirá nulo (null) na String de resposta

JANELAS DE DIÁLOGO

OptionDialog

Janela de dialogo mais complexa de ser utilizada, pois é capaz de combinar todos os recursos destas outras janelas.

Retorna um valor inteiro, conforme o botão escolhido.

JOptionPane.showOptionDialog(Componente,Mensagem, Título da Mensagem, Botões, Tipo de Mensagem, Ícone, Array de objetos, Seleção padrão)

- Componente: referente ao objeto do tipo contêiner, que sendo null posicionará esta janela no centro da tela
- Mensagem: mensagem orientadora para o usuário
- Título da Mensagem: título da janela de dialogo
- Botões: usa os botões padrões ou novos botões

JANELAS DE DIÁLOGO

JOptionPane.showOptionDialog(Componente,Mensagem, Título da Mensagem, Botões, Tipo de Mensagem, Ícone, Array de objetos, Seleção padrão)

- Tipo de Mensagem: umas das mensagens padrões que possibilitam indicar o ícone padrão em outras janelas
- Ícone: objeto ImageIcon que permite a inclusão de outro ícone diferente do padrão, ou null para ficar sem ícone
- Array de objetos: array que indicará as possíveis escolhas nesta caixa de diálogo, caso YES_NO_OPTION, OK_CANCEL_OPTION, ... não estejam sendo usados
- Seleção padrão: objeto de padrão inicial selecionado, quando as constantes padrões não estiverem em uso

JANELAS DE DIÁLOGO

```java
/** Síntese
 *    Objetivo: Identificar a faixa de idade do usuário
 *    Entrada:  faixa de idade
 *    Saída:    confirmação da faixa de idade indicada
 */
import javax.swing.*;
public class OpcaoDialogo {
    public static void main(String[] args) {
        String [] escolha = {"Entre 1 e 20 anos",
                "Entre 21 e 40 anos","Mais de 40 anos"};
        String idade = null;
        try {
            int resposta = JOptionPane.showOptionDialog(null,
                    "Qual sua idade?","Idade", 0,
                    JOptionPane.INFORMATION_MESSAGE,
                    null, escolha, escolha[0]);
            idade = (escolha[resposta]);
            System.out.print("Faixa de Idade = " + idade);
        } catch (ArrayIndexOutOfBoundsException ex) {
            System.out.println("Programa encerrado.");
        }
    }
}
```

JANELAS DE DIÁLOGO

Para cada uma destas janelas de dialogo existe um método que permite definir um outro ícone qualquer.

A organização da mensagem de orientação a ser apresentada por uma janela de dialogo pode ser melhor elaborado com saltos de linha '\n'.

Apesar da facilidade de uso destas janelas de interação, seus tamanhos são limitados para manipulação de vários componentes gráficos, sendo as mesmas empregadas somente na atividade de "dialogo" ágil com os usuários.

O estudo de outros recursos gráficos possíveis pela classe Swing e AWT será efetivado mais a frente em nossa disciplina, onde vários componentes serão usados.

JANELAS DE DIÁLOGO

Todas estas opções podem parecer confusas para interagir com os usuários, mas a prática mostrará que esta aparência esta errada. Siga os passos para elaborar uma interação adequada e agradável com seu usuário:

1. Escolha o tipo de dialogo (mensagem, confirmação, entrada ou opção)
2. Selecione um ícone (informação, erro, advertência, pergunta, nenhum ou um outro qualquer personalizado)
3. Reforce o intuito do dialogo no título coerente desta janela
4. No dialogo de confirmação defina as opções possíveis (Yes/No, Yes/No/Cancel, Ok/Cancel, padrão, ...)
5. Para janela de opções defina os componentes desejados para interação ágil e adequada
6. Seja direto na janela de entrada para que usuário forneça o retorno esperado pelo programa
7. Encontre o método apropriado da JOptionPane a ser chamado

CLASSES EMPACOTADORAS (WRAPPERS)

Na Programação poderão existir situações onde seja necessária a conversão (cast) de um tipo de dado primitivo em um objeto (empacotar o tipo primitivo).

Esta conversão não pode ser feita implicitamente (automática), sendo preciso o uso das classes referentes a cada tipo, que estão disponíveis na **java.lang**.

Classe	Tipo Primitivo
Byte	⇒ byte
Short	⇒ short
Integer	⇒ int
Long	⇒ long
Float	⇒ float
Double	⇒ double

superclasse Number

Classe	Tipo Primitivo
Character	⇒ char
Boolean	⇒ boolean
Void	⇒ void

CLASSES EMPACOTADORAS (WRAPPERS)

O uso destas classes permitem a referência a valores de tipos primitivos, o que não é possível para estes tipos de dados diretamente.

```
public static void main(String[] args) {
        Integer preco = 10;
        Float taxa = 0.25F;
        System.out.print("Aumento= " + (preco * taxa));
}
```

Por meio destas classes ainda são implementados métodos que manipulam, adequadamente, estes tipos primitivos, convertidos em objetos e vice-versa (objetos $\rightarrow$ tipos primitivos), sendo alguns deles relacionados a seguir:

CLASSES EMPACOTADORAS (WRAPPERS)

Para classe **Integer**:

MÉTODO	FUNCIONALIDADE
intValue()	retorna o valor do objeto Integer na forma de int
toString(int)	retorna objeto String representando o valor inteiro
parseInt(String)	retorna o valor inteiro referente a String
valueOf(String)	retorna um objeto Integer inicializado com o valor inteiro que será convertido a String (parâmetro)
:	:

CLASSES EMPACOTADORAS (WRAPPERS)

Exemplo:

```
int idade = Integer.parseInt(str); // str se torna int
```

Para cada um dos tipos primitivos existem métodos similares implementados em suas respectivas classes, como em:

```
double salario= Double.parseDouble(str); // str para double
```

Todos estes tipos possuem suas classes e métodos.

CLASSES EMPACOTADORAS (WRAPPERS)

Além de métodos estas classes também possuem alguns atributos, tais como:

MIN_VALUE - menor valor de seu tipo
MAX_VALUE - maior valor de seu tipo

```
int valor = Integer.MAX_VALUE; // atributo constante
System.out.print("Maior Inteiro = " + valor);
```

Exemplo de manipulações destas classes:

```
Integer anoCorrente = new Integer(2008); // mét. construtor
int ano = anoCorrente.intValue(); // converte Integer em int

String anos = "42"; // comum converter String em int
int idade = Integer.parseInt(anos);

String condicional = "false";
boolean condicao = Boolean.getBoolean(condicional);
```

CLASSES EMPACOTADORAS (WRAPPERS)

A partir do Java 5 foi implementado o Autoboxing, permitindo uma sintaxe mais elegante sobre os mesmos recurso. Isso possibilitou a seguinte codificação:

```
Integer anoCorrente = 2008;
int ano = anoCorrente;
```

No entanto, é importante ressaltar que tipo primitivo e objeto não são a mesma coisa, mas a partir desta versão (Java 5) a codificação pôde ser facilitada, inclusive sobre a superclasse Object, passando um tipo primitivo para um método que recebe **Object** como argumento:

```
Object valor = 3;
```

EXERCÍCIOS DE FIXAÇÃO

1) Faça um programa orientado a objeto que armazene o ano, maior que 1900 e menor que o ano atual, e o nome completo do presidente brasileiro neste ano. Estes dados serão cadastrados enquanto o usuário quiser e indicarão anos onde movimentações nos astros por nossa galáxia foram expressivos, por exemplo: ano e presidente em que o cometa Halley passou e foi visto da Terra, eclipses, etc. Todos os dados de cada entidade em sua solução não poderão ser de tipos de dados primitivos, devendo ser usadas as **classes empacotadoras**. Toda interação com o usuário para coleta destes dados deverá acontecer por **janela gráfica interativa**, mas o relatório final e tabelar será mostrado na console com todos os registros feitos pelo usuário. Este programa só poderá ser encerrado se ao menos um cadastro válido for realizado.

EXERCÍCIOS DE FIXAÇÃO

2) Elabore um programa que permita o controle de uma coleção de quadros e de seus pintores. Para cada quadro será fornecido um código de identificação, a identificação do pintor, preço e ano de aquisição do mesmo. Para cada pintor será cadastrado o nome, seu código de identificação pessoal e o ano de nascimento. Faça um programa que apresente um menu simples as opções de: cadastramento de um novo quadro; cadastramento de um novo pintor; listagem de todas as telas de um pintor informado, com o valor total da soma dos valores dos mesmos; apresentação de todos os quadros cadastrados até o momento no programa; opção de encerrar o programa. Identifique quais classes serão necessárias nesta solução e despreze as diferenças entre letras maiúsculas e minúsculas na pesquisa do pintor. Todos os dados de cada entidade em sua solução não poderão ser de tipos de dados primitivos (**usar classe empacotadora**).