

Prof. Me. Renato Alves Ferreira
email: renato.ferreira@fmu.br

Disciplina:

Programação Orientada a Objetos

Agenda da aula

- Correção da Atividade Anterior
- Múltiplos Construtores em uma Classe
- Manipulação de elementos gráficos (classe JOptionPane)

Correção da Atividade anterior (Faça um novo programa MenuDigitaConta ou acrescentar o menu ao programa DigitaConta para permitir a seleção de qual classe será manipulada, se Poupança ou Conta Corrente. Uso de estruturas IF, WHILE ou DO WHILE)

```
1 package exemplodeclasses;           // nome do pacote "pasta" do projeto
2 import java.util.Scanner;           // importar classe Scanner para tratar a entrada de dados via teclado
3
4 public class DigitaConta {
5     public static void main (String args[]) {
6         Scanner tc = new Scanner(System.in); //instância da classe Scanner, gerando o objeto tc
7         int opcao = 9;
8
9         while (opcao !=0 ) {
10             System.out.print("Escolha <1> para Poupança, <2> para Corrente <0> para Sair : ");
11             opcao = tc.nextInt();
12
13             if (opcao == 1) {
14                 ContaPoupanca cp = new ContaPoupanca(); //instância da classe ContaPoupanca, gerando o objeto cp
15                                     // ***** segue a digitação dos dados *****
16                 System.out.println("\n\n Cadastro em Conta Poupança");
17
18                 while(cp.numeroConta == 0) {           // exemplo de consistência com While
19                     System.out.print("Digite o numero da conta :");
20                     cp.numeroConta = tc.nextInt();
21                 }
22
23                 do {                                   // exemplo de consistência com DO While
24                     System.out.print("Digite o numero do cliente:");
25                     cp.numeroCliente = tc.nextInt();
26                 } while (cp.numeroCliente == 0);
27
28                 System.out.print("Digite a data de abertura:");
29                 cp.dataAbertura = tc.next();
30                                     // chamada de métodos saque e depósito com valor passado por digitação
31                 System.out.print("Digite o valor do depósito:");
32                 cp.depositoConta(tc.nextDouble());
33
34                 System.out.print("Digite o valor do saque:");
```

```
35 cp.saqueConta(tc.nextDouble());
36                                     //lembrando que o saldo inicia com 100,00
37 System.out.println("Agora, seu saldo em conta poupança é :" + cp.saldo);
38 } // fim do SE
39
40 if (opcao == 2) {
41     ContaCorrente cc = new ContaCorrente(); //instância da classe ContaCorrente, gerando o objeto cc
42                                     // ***** segue a digitação dos dados *****
43     System.out.println("\n\n Cadastro em Conta Corrente");
44     System.out.print("Digite o numero da conta :");
45     cc.numeroConta = tc.nextInt();
46
47     System.out.print("Digite o numero do cliente:");
48     cc.numeroCliente = tc.nextInt();
49
50     System.out.print("Digite a data de abertura:");
51     cc.dataAbertura = tc.next();
52     // chamada de métodos saque e depósito com valor passado por digitação
53     System.out.print("Digite o valor do depósito:");
54     cc.depositoConta(tc.nextDouble());
55
56     System.out.print("Digite o valor do saque:");
57     cc.saqueConta(tc.nextDouble());
58     //lembrando que o saldo inicia com 100,00
59     System.out.println("Agora, seu saldo em conta corrente é :" + cc.saldo);
60 } // fim do 2o Se
61 } // fim do While
62 // fim do main
63 }
```

Resultado (escolhido 2 no menu e posteriormente 0)

Saida



ExemplodeClasses (run) x

ExemplodeClasses (run) #2 x



run:



Escolha <1> para Poupança, <2> para Corrente <0> para Sair : 2



Cadastro em Conta Corrente

Digite o numero da conta :20020

Digite o numero do cliente:1501

Digite a data de abertura:08/09/2019

Digite o valor do depósito:900

Digite o valor do saque:200

Agora, seu saldo em conta corrente é :800.0

Escolha <1> para Poupança, <2> para Corrente <0> para Sair : 0


CONSTRUÍDO COM SUCESSO (tempo total: 1 minuto 10 segundos)

Aplicando Construtores Múltiplos à Classe

Classe com Construtores múltiplos

```
1 package exemplodeclasses;
2
3 public class ContaPoupanca extends Conta{
4     String dataAniversarios;
5     void saqueConta(double valor){
6         if((saldo - valor) < 0)
7             System.out.println("ERRO: saldo ficará abaixo de zero! ");
8         else
9             saldo=saldo - valor;
10    }
11
12    ContaPoupanca () {           // construtor sem parâmetros
13    }
14
15    ContaPoupanca (String x, double y) {           // construtor com dois parâmetros
16        dataAbertura = x;
17        saldo = y;
18    }
19
20    ContaPoupanca (int numeroConta, String dataAbertura, double saldo) {           // construtor com N parâmetros
21        this.numeroConta = numeroConta;
22        this.dataAbertura = dataAbertura;
23        this.saldo = saldo;
24    }
25 }
```

Acrescente os novos Construtores



Aplicando Construtores múltiplos

Classe TestaExemplodeClasses invocando 3 construtores

```
1 package exemplodeclasses;
2
3 public class TestaExemplodeClasses {
4     public static void main(String[] args) {
5         ContaPoupanca cp = new ContaPoupanca();    //invocando o 1o construtor
6         // ou
7         ContaPoupanca cp2 = new ContaPoupanca("09/08/2019", 500.00);    //invocando o 2o construtor
8         // ou
9         ContaPoupanca cp3 = new ContaPoupanca(10020, "09/08/2019", 800.00); //invocando o 3o construtor
10
11         cp3.saqueConta(30.00);
12         System.out.println("Dados iniciais da Popança. N.Conta: " + cp3.numeroConta + " Dt.Abertura " + cp3.dataAbertura + " Saldo: " + cp3.saldo);
13
14         ContaCorrente cr = new ContaCorrente(); //Objeto cr para conta corrente
15         cr.saqueConta(500.00);
16         System.out.println("\n\n Saldo em Conta Corrente:" + cr.saldo);
17     }
18 }
```

Criação de múltiplos objetos com construtores diferentes.

Saída



ExemplodeClasses (run) x

ExemplodeClasses (run) #2 x



run:



Dados iniciais da Popança. Conta: 10020 Dt.Abertura 09/08/2019 Saldo: 770.0



Saldo em Conta Corrente:-400.0

CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)

Manipulando Objetos Gráficos com a classe JOptionPane do pacote javax.swing

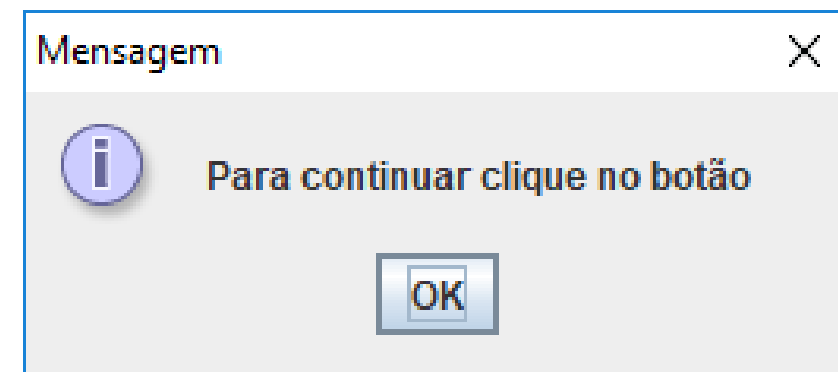
(Crie um projeto novo ou aplique em algum programa do seu projeto atual)

Classe para manipulação de objetos gráficos para entrada de dados – classe JOptionPane

(Necessário importar a classe **JOptionPane** do pacote **javax.swing**. Ex. `import javax.swing.JOptionPane`)

- Apenas apresenta caixa de mensagem

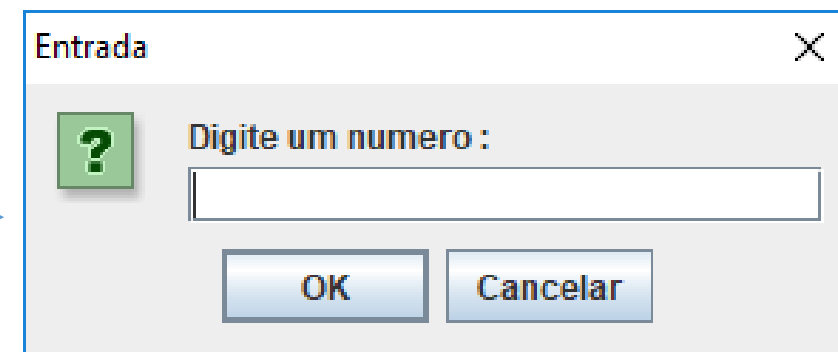
JOptionPane.showMessageDialog (**null**, "Para continuar clique no botão ");



- Mostrar caixa de diálogo e recebe uma string digitada

String **texto**;

texto = **JOptionPane.showInputDialog** (**null**, "Digite um numero :");



- Converter string em inteiro

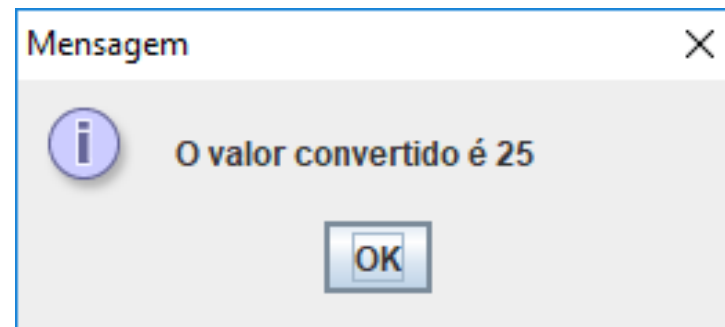
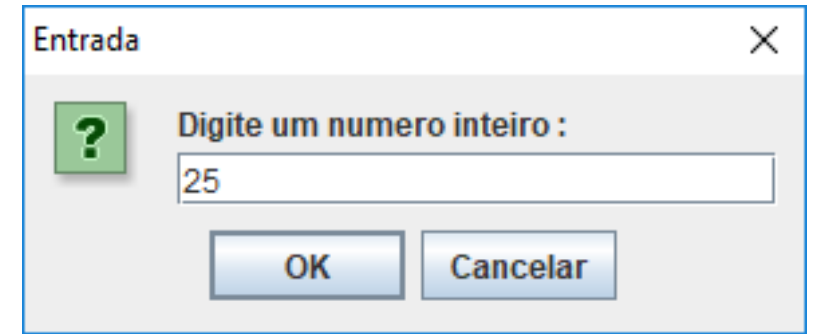
int i;

i = **Integer.parseInt**(**texto**);

// classe.método que converte string para inteiro

Exemplo de objeto gráfico para entrada de dados

```
2
3 import javax.swing.JOptionPane;
4
5 public class EntradaGrafica {
6     public static void main (String args[]) {
7
8         String texto;
9         texto = JOptionPane.showInputDialog (null, "Digite um numero inteiro :");
10
11         int i;
12         i = Integer.parseInt(texto);           // classe.método que converte de string para inteiro
13
14         JOptionPane.showMessageDialog(null, "O valor convertido é " + i );
15     }
16 }
```



Atividade com objeto gráfico: JOptionpane.showInputDialog() e JOptionPane.showMessageDialog()

- Faça um programa que receba 4 valores via teclado no modo gráfico.
 - Teste antes de prosseguir.
 - Depois, acrescente os demais itens:
-
- ✓ Tais dados devem ser convertidos para tipo numérico, inteiro ou double.
 - ✓ Os outros 3 valores não poderão receber 0. Colocar consistência.
 - ✓ Apresente a média dos valores e inicie o programa.
 - ✓ O programa deverá encerrar apenas se o 1º valor digitado for 0.



Durante o curso:

- Leitura do artigo indicado (Recurso 1)
- Leitura do livro indicado (Recurso 2)

Recurso 1

Artigo Devmedia: “Principais conceitos da Programação Orientada a Objetos” Disponível em:

<https://www.devmedia.com.br/principais-conceitos-da-programacao-orientada-a-objetos/32285>

Recurso 2

Livro: FÉLIX, R. Programação Orientada a Objetos. São Paulo: Pearson Education do Brasil, 2016. 164p.
[Biblioteca Virtual Universitária] pp 1-4.

Indicações

- Tiexpert
- Devmedia
- GUJ
- Video-aulas youtube

Referências

Livro: FURGERI, S. Java 8 - Ensino Didático - Desenvolvimento e Implementação de Aplicações. São Paulo: Érica, 2015. 320p. [Minha Biblioteca]. Capítulos 1 e 2.

Te espero na próxima aula!

