

Introdução à Programação Orientada a Objetos

Aula 9

Entrada de dados simples (modo grafico)

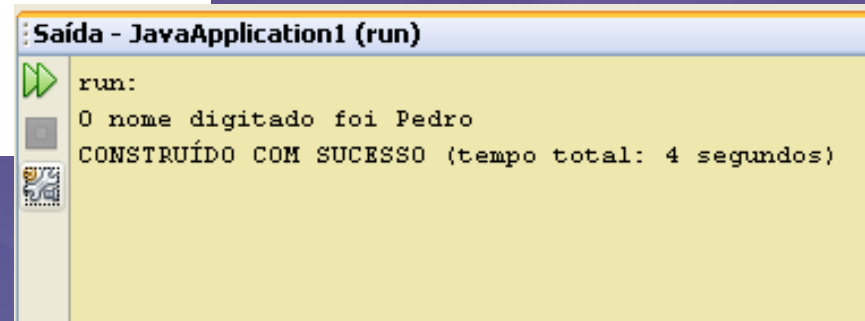
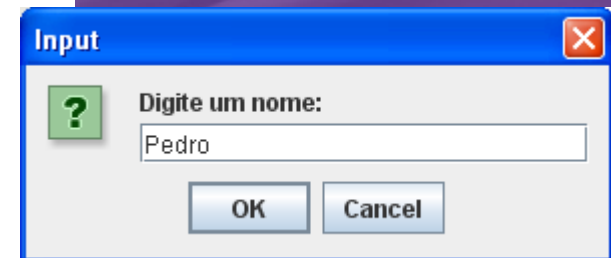
- Para pegar uma entrada de dados do teclado através de uma caixa de diálogo, usa-se o método “showInputDialog” da classe “JOptionPane” do pacote “javax.swing”.

```
package javaapplication1;

import javax.swing.JOptionPane;

public class Main {

    public static void main(String[] args) {
        String nome;
        nome = JOptionPane.showInputDialog("Digite um nome:");
        System.out.println("O nome digitado foi " + nome);
    }
}
```



Saída de dados simples (modo grafico)

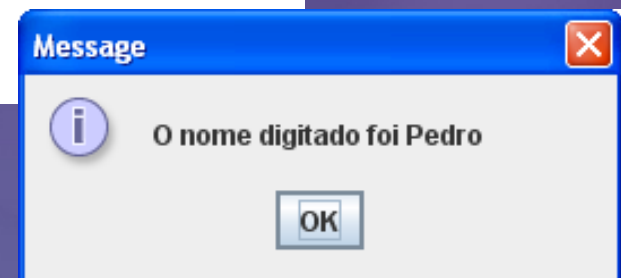
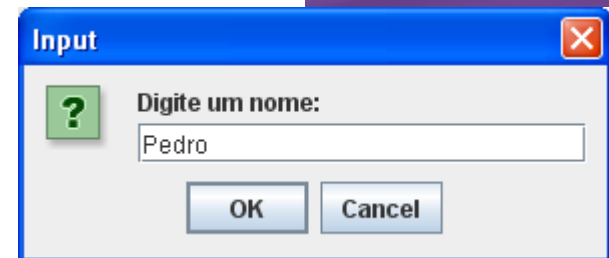
- Para mostrar um dado na tela através de uma caixa de diálogo, TAMBÉM se usa a classe “JOptionPane” do pacote “javax.swing”, porém o método é “showMessageDialog”

```
package javaapplication1;

import javax.swing.JOptionPane;

public class Main {

    public static void main(String[] args) {
        String nome;
        nome = JOptionPane.showInputDialog("Digite um nome:");
        JOptionPane.showMessageDialog(null, "O nome digitado foi " + nome);
    }
}
```



Conversão de valores String

- Para converter uma String em um numero inteiro ou real usa-se os metodos “parse” da classe correspondente ao tipo do numero:

```
1  package javaapplication15;
2
3  import javax.swing.JOptionPane;
4
5  public class Main {
6
7      public static void main(String[] args) {
8          String s;
9          s = JOptionPane.showInputDialog("Digite um número inteiro:");
10         int i = Integer.parseInt(s);
11         JOptionPane.showMessageDialog(null, "numero digitado: " + i);
12
13         s = JOptionPane.showInputDialog("Digite um número real:");
14         double d = Double.parseDouble(s);
15         JOptionPane.showMessageDialog(null, "numero digitado: " + d);
16     }
17
18 }
```


Sobrecarga de métodos

- Mecanismo que permite a um método ter o nome igual ao de outro método existente.
- Apesar de terem o mesmo nome, métodos são considerados diferentes quando têm assinaturas diferentes.
- O nome do método e seus parâmetros são chamados de assinatura do método.
- O tipo de retorno do método não faz parte da sua assinatura.

Exemplo

```
3 public class Data {
4     private int dia, mes, ano;
5
6     public void setData(){
7         dia = 1;
8         mes = 1;
9         ano = 1980;
10    }
11
12    public void setData(int d, int m, int a){
13        dia = d;
14        mes = m;
15        ano = a;
16    }
17
18    public String getData(){
19        return dia + "/" + mes + "/" + ano;
20    }
21 }
```

```
run:
1/1/1980
31/12/2008
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

```
5 public class Main {
6
7     public static void main(String[] args) {
8         Data nascim = new Data();
9         nascim.setData();
10        System.out.println(nascim.getData());
11        nascim.setData(31, 12, 2008);
12        System.out.println(nascim.getData());
13    }
14 }
```

A palavra-chave this

- Representa a referencia à própria instância da classe.

```
3 public class Data {
4     private int dia, mes, ano;
5
6     public void setData(){
7         dia = 1;
8         mes = 1;
9         ano = 1980;
10    }
11
12    public void setData(int dia, int mes, int ano){
13        if(dataEhValida(dia, mes, ano) ){
14            this.dia = dia;
15            this.mes = mes;
16            this.ano = ano;
17        }
18    }
19
20    private boolean dataEhValida(int d, int m, int a){
21        if(d >= 1 && d <= 31 && m >= 1 && m <= 12 && a >= 1800)
22            return true;
23        else
24            return false;
25    }
}
```

Sobrecarga de construtores

```
3 public class Data {
4     private int dia, mes, ano;
5
6     public Data(){
7         setData();
8     }
9
10    public Data(int d, int m, int a){
11        this();
12        setData(d, m, a);
13    }
14
15    public void setData(){
16        dia = 1;
17        mes = 1;
18        ano = 1980;
19    }
20
21    public void setData(int dia, int mes, int ano){
22        if(dataEhValida(dia, mes, ano) ){
23            this.dia = dia;
24            this.mes = mes;
25            this.ano = ano;
26        }
27    }

```

```
run:
1/1/1980
31/12/2008
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

```
7 public static void main(String[] args) {
8     Data nascim = new Data();
9     System.out.println(nascim.getData());
10    Data entrada = new Data(31,12,2008);
11    System.out.println(entrada.getData());
12 }
13 }
```

Regras para chamada de construtores

1. Construtores não são chamados pelo nome e sim pela palavra `this`.
2. Somente construtores pode chamar construtores.
3. Se um construtor for chamado a partir de outro, a chamada deve ser a primeira linha de código dentro do corpo do construtor.
4. Construtores podem chamar outros métodos mas, métodos não podem chamar construtores.
5. Um construtor só pode chamar diretamente outro construtor e não a si próprio.