

A Linguagem Java

Java Básico

Profa. Sandra Bianca Henriques Geroldo
Email: profsandra.fatec@gmail.com

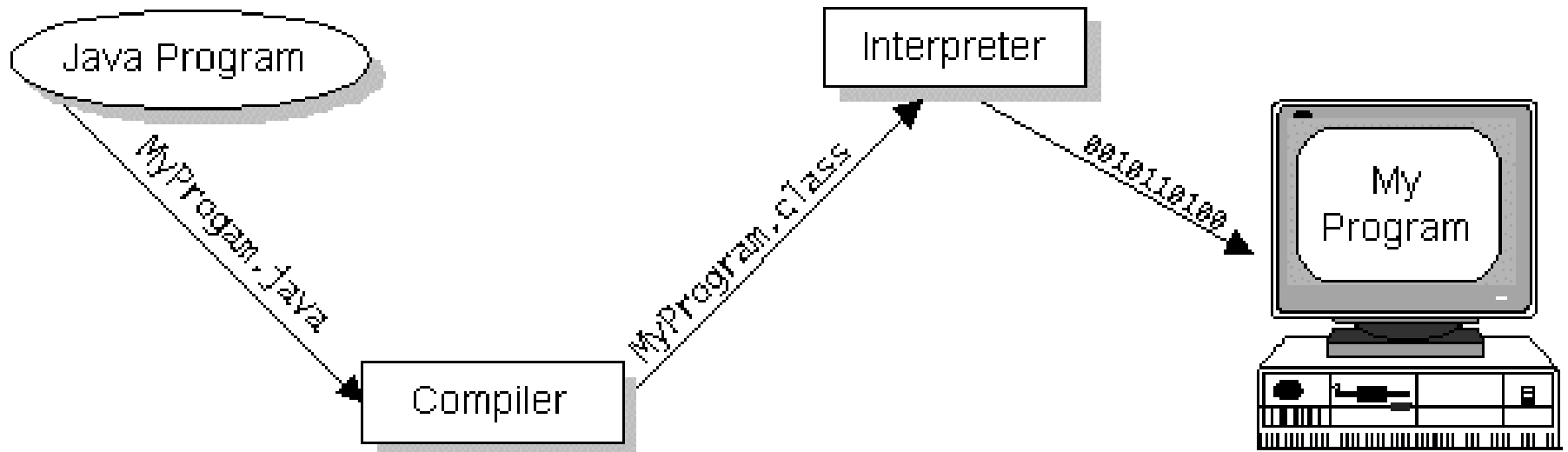
Linguagem Java

► Linguagem Orientada a Objetos

- Todos os programas Java são compilados e interpretados
- O compilador transforma o programa em *bytecodes* independentes de plataforma
- O interpretador testa e executa os *bytecodes*

Linguagem Java

► Compilada e interpretada



Linguagem Java

► Comentários

- `/* texto */`

O compilador ignora tudo entre `/*` e `*/`

- `/** documentacao */`

indica um comentário para documentação.
Utilizado pela ferramenta *javadoc*

- `// texto`

O compilador ignora todos os caracteres de `//` até o final da linha

Linguagem Java

► Operadores

- Atribuição: `=`
- Aritméticos: `+` `-` `*` `/` `%`

Saída de informação

- ▶ Para exibir uma mensagem de saída no modo console usa-se o comando:

- `System.out.println("Mensagem");`

Ou

- `System.out.println("Mensagem"+variavel);`

- Obs: In ao final do print faz pular uma linha

Entrada de Dados (Classe Scanner)

- ▶ A classe Scanner permite que seja realizada a entrada de dados no modo Console.
- ▶ Para usar a classe é necessário adicionar um import

- `import java.util.Scanner;`

Entrada de Dados (Classe Scanner)

- ▶ Para permitir a digitação será necessário criar um objeto a fim de manipular a entrada

Scanner **entra** = new Scanner(System.in);

onde: **entra** é o objeto que será utilizado. System.in é o que faz a leitura do que se escreve no teclado.

Entrada de Dados (Classe Scanner)

► Exemplo 1:

```
1  import java.util.Scanner;
2  public class Entrada{
3
4      public static void main(String args[]){
5
6          Scanner entra = new Scanner(System.in);
7          int soma, num1, num2;
8
9          System.out.print("Digite o primeiro valor: ");
10         num1 = entra.nextInt();
11         System.out.print("Digite o segundo valor: ");
12         num2 = entra.nextInt();
13
14         soma=num1+num2;
15         System.out.println("Resultado da soma: "+soma);
16
17     }
18
19 }
```

Entrada de Dados (Classe Scanner)

- ▶ Métodos usados para os tipos de dados mais usados:
- ▶ Para o tipo de dado double:
 nomeDoObjeto.**nextDouble()**;
- ▶ Para o tipo de dado String:
 nomeDoObjeto.**nextLine()**;

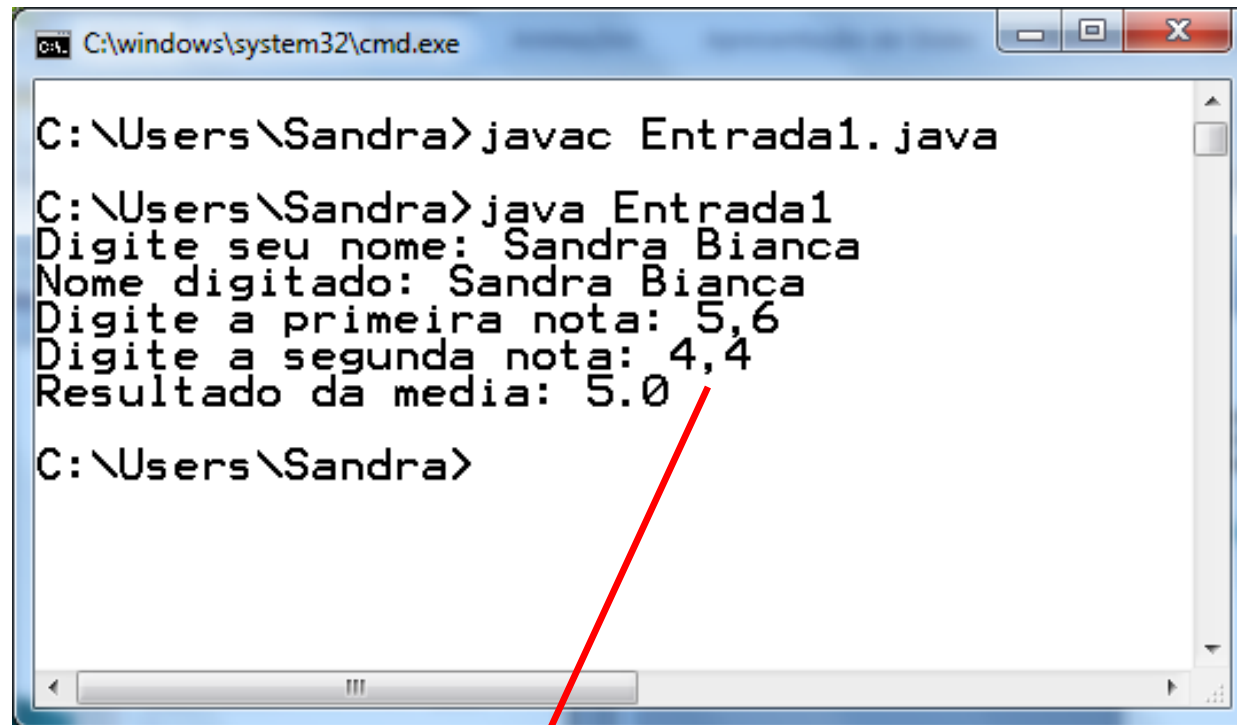
Entrada de Dados (Classe Scanner)

▶ Exemplo 2:

```
1  import java.util.Scanner;
2  public class Entrada1{
3
4      public static void main(String args[]){
5
6          Scanner entra = new Scanner(System.in);
7          String nome;
8          double nota1, nota2, media;
9
10         System.out.print("Digite seu nome: ");
11         nome = entra.nextLine();
12
13         System.out.println("Nome digitado: "+nome);
14
15         System.out.print("Digite a primeira nota: ");
16         nota1 = entra.nextDouble();
17         System.out.print("Digite a segunda nota: ");
18         nota2 = entra.nextDouble();
19         media=(nota1+nota2)/2;
20         System.out.println("Resultado da media: "+media);
21
22     }
23
24 }
```

Entrada de Dados (Classe Scanner)

► Exemplo 2:



```
C:\windows\system32\cmd.exe

C:\Users\Sandra>javac Entrada1.java

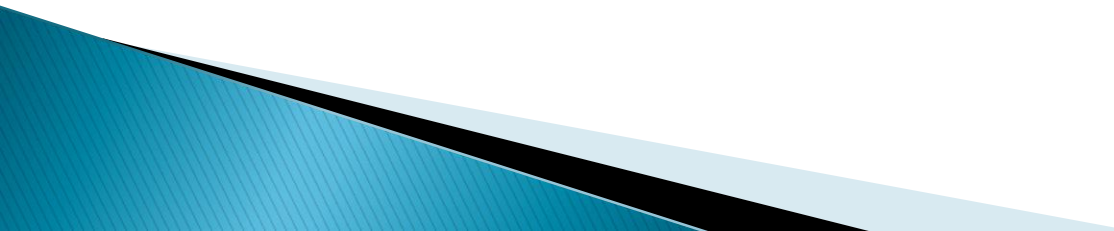
C:\Users\Sandra>java Entrada1
Digite seu nome: Sandra Bianca
Nome digitado: Sandra Bianca
Digite a primeira nota: 5,6
Digite a segunda nota: 4,4
Resultado da media: 5.0

C:\Users\Sandra>
```

A red arrow points from the text 'Obs: ao realizar a digitação de um valor do tipo double no console use a vírgula para entrada de dados' to the comma in the input '5,6'.

Obs: ao realizar a digitação de um valor do tipo double no console use a **vírgula** para entrada de dados

Entrada e Saída : GUI simples

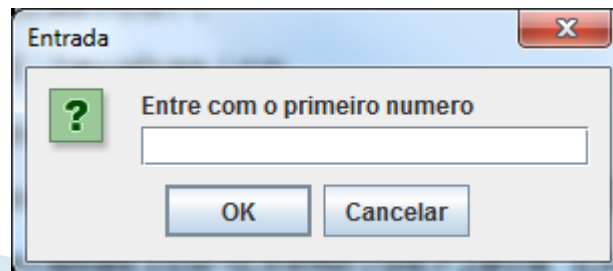
- ▶ Na linguagem Java pode-se usar a classe `JOptionPane`
 - ▶ A `JOptionPane` é encontrada no pacote `javax.swing`
 - ▶ Fornece caixas de diálogo tanto para entrada como para saída
 - ▶ Esses diálogos são exibidos invocando métodos `JOptionPane static`
- 

JOptionPane.showInputDialog

- ▶ Para usar uma entrada usa-se o método `showInputDialog`
- ▶ Para utilizar esse recurso é necessário adicionar a linha a seguir no programa .java a fim de importar a classe `JOptionPane`
 - `import javax.swing.JOptionPane;`

JOptionPane.showInputDialog

- ▶ O método `showInputDialog` permite a entrada de um dado, sendo esse definido em uma string
- ▶ Para a exibição da caixa de diálogo abaixo e necessário adicionar algumas linhas de código (as quais poder ser visualizadas no slide a seguir)



JOptionPane.showInputDialog

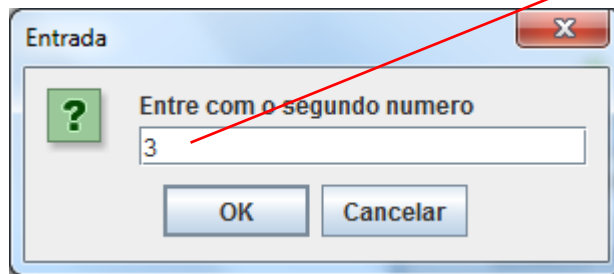
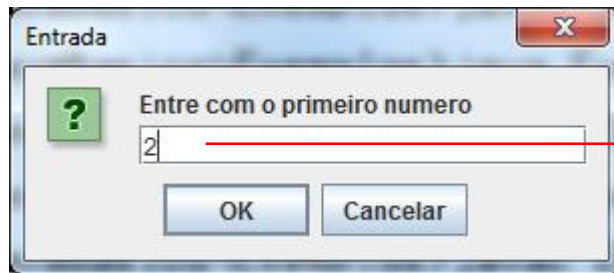
```
1  import javax.swing.JOptionPane; // import class JOptionPane
2
3  public class Exemplo1 {
4      public static void main( String args[] )
5      {
6          String primeironumero,
7              segundonumero;
8          int numero1,
9              numero2,
10             resultsoma;
11
12         // le o primeiro numero como uma string
13         primeironumero = JOptionPane.showInputDialog( "Entre com o primeiro numero" );
14
15         // le o segundo numero como uma string
16         segundonumero = JOptionPane.showInputDialog( "Entre com o segundo numero" );
17
18         // converte numeros do tipo string para inteiro
19         numero1 = Integer.parseInt( primeironumero );
20         numero2 = Integer.parseInt( segundonumero );
21
22         // soma os numeros
23         resultsoma = numero1 + numero2;
24
25         // mostra o resultado
26         JOptionPane.showMessageDialog( null, "Soma igual a: " + resultsoma, "Resultado", JOptionPane.PLAIN_MESSAGE );
27
28         System.exit( 0 ); // termina o programa
29     }
30 }
```


JOptionPane.showInputDialog

- ▶ Percebe-se que no exemplo anterior os seguintes detalhes:
 - Linha 1: importando a classe JOptionPane;
 - Linha 6: declarando strings locais;
 - Linha 13: atribuindo à string local `primeironumero` o resultado do método `static` da chamada `JOptionPane.showInputDialog`. Esse método exibe um diálogo de entrada, utilizando um `String` do método (“Entre com o primeiro numero”) como um prompt para o usuário;
 - Linhas 19 e 20: responsáveis por converter a string de entrada em valores do tipo inteiro (`int`).

JOptionPane.showInputDialog

- ▶ Caixas de diálogo exibidas pelas linhas 13 e 16 do exemplo:



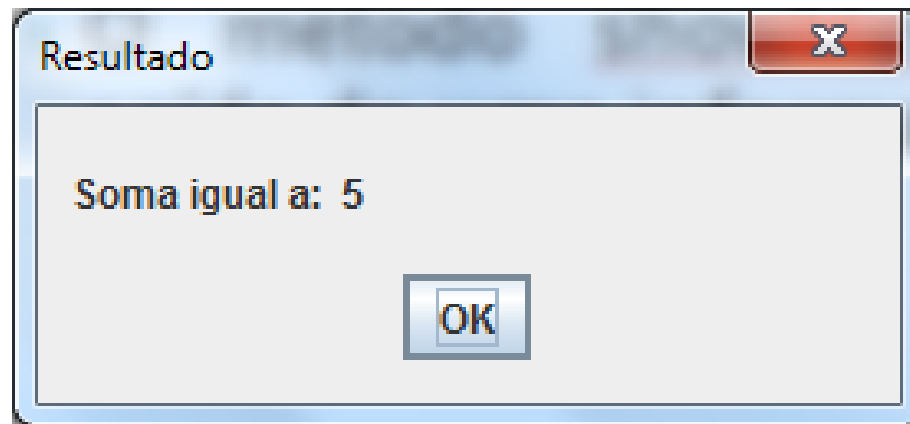
Após digitar um valor de entrada, o usuário irá clicar em OK. O `showInputDialog` retorna para o programa os valores 2 e 3 como `String`. Para que uma soma seja realizada é necessário converter as `String` em `int`. Isso é realizado pelas linhas 19 e 20

```
18  
19  
20  
21
```

```
// converte numeros do tipo string para inteiro  
numero1 = Integer.parseInt( primeironumero);  
numero2 = Integer.parseInt( segundonumero);
```

JOptionPane.showMessageDialog

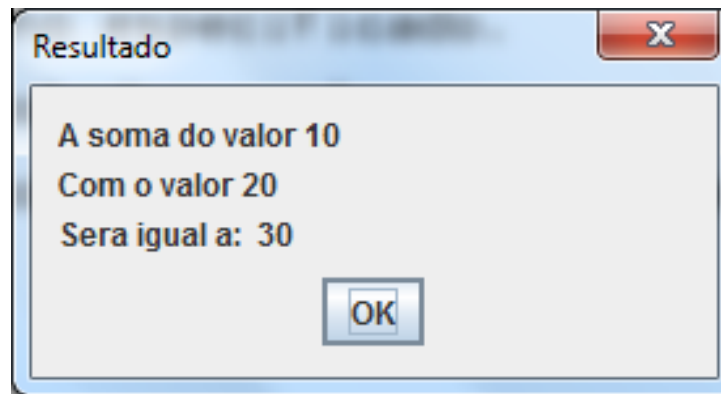
- ▶ O método `showMessageDialog` permite a saída de uma informação



- ▶ Assim, na linha 26 percebe-se que o método `static JOptionPane showMessageDialog` permite exibir um diálogo de mensagem

JOptionPane.showMessageDialog

- ▶ No método showMessageDialog pode-se pular linha
- ▶ Para isso use \n entre aspas: “\n”
 - `JOptionPane.showMessageDialog(null, "A soma do valor " + numero1 + "\nCom o valor " + numero2 + "\nSera igual a: " + resultsoma, "Resultado", JOptionPane.PLAIN_MESSAGE);`



Estruturas Condicionais

- ▶ No dia-a-dia, surge a necessidade de tomar decisões. Na lógica de programação, e na criação de programas, isso também ocorre. Para isso se utilizam Estruturas de Seleção, conhecidas também como Estrutura de Controle Decisória ou de Decisão, ou ainda conhecidas como Estrutura de Controle Condicional.
- ▶ As Estruturas Condicionais permitem a execução de um ou vários comandos. Para isso uma condição, ou várias condições, devem ser satisfeitas.
- ▶ A Estrutura de Seleção pode ainda ser classificada em: Simples, Composta ou Aninhada (Encadeada).

Operadores Relacionais

- ▶ Utilizamos os operadores relacionais para realizar comparações entre dois valores do mesmo tipo primitivo (ex: inteiro, real). Esses valores são representados por constantes, variáveis ou expressões aritméticas.

Operador	Operação	Exemplo
==	igualdade	if (x==10)
!=	diferente	if (a!=b)
<	menor que	if (y<10)
>	maior que	if (z>100)
<=	menor ou igual que	if (cont<=10)
>=	maior ou igual que	if (c>=1)

Operadores Lógicos

- ▶ Conectivos (palavras que fazem ligação entre duas frases) serão utilizados para a formação de novas proposições a partir de outras já conhecidas.
- ▶ Em programas usa-se os seguintes conectivos:

Operador (conectivo)	Operação	Exemplo
&&	Conjunção (E)	if ((x>10) && (y<5))
	Disjunção (OU)	if ((x>10) (y<5))
!	Negação (NÃO)	if (!(x>10))

Tabela da Verdade – operador E

► Operador Lógico E (&&)

Expressão	Resultado
V && V	V
V && F	F
F && V	F
F && F	F

- Ex: considere a expressão $((x > 10) \&\& (y < 5))$ e as possibilidades apontadas pela tabela abaixo:

valor de x	valor de y	Resultado
15 (V)	3 (V)	V
20 (V)	10 (F)	F
5 (F)	4 (V)	F
8 (F)	20 (F)	F

Tabela da Verdade – operador **OU**

- ▶ Operador Lógico OU (||)

Expressão	Resultado
V V	V
V F	V
F V	V
F F	F

- ▶ Ex: considere a expressão $((x > 10) \text{ || } (y < 5))$ e as possibilidades apontadas pela tabela abaixo:

valor de x	valor de y	Resultado
15 (V)	3 (V)	V
20 (V)	10 (F)	V
5 (F)	4 (V)	V
8 (F)	20 (F)	F

Estrutura de Seleção Simples

- ▶ A estrutura de seleção simples usa o comando if.
- ▶ O comando if instrui o computador a tomar uma decisão simples, de forma que um bloco de instruções é executado uma única vez se a **expressão de teste for verdadeira**.

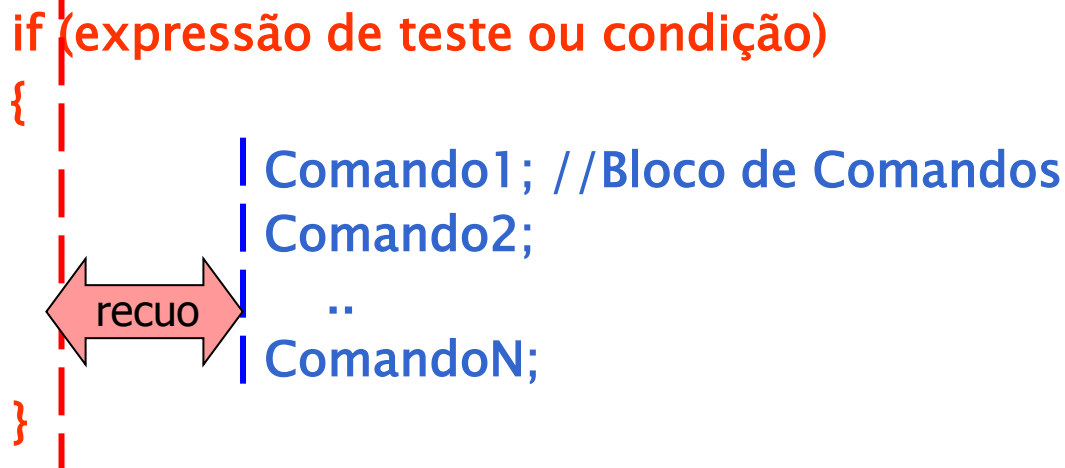
- ▶ **Sintaxe**

// a condição deve ser verdadeira para executar os comandos

Recur por
critério de
identação
(para
organização
do código)

```
if (expressão de teste ou condição)
{
    Comando1; //Bloco de Comandos
    Comando2;
    ..
    ComandoN;
}
```

recuo



Exemplo de Seleção Simples

```
import java.lang.*;
import java.io.*;
import javax.swing.JOptionPane; // import class JOptionPane

public class Exemplo2 {
    public static void main( String args[] ) throws java.lang.Exception
    {
        String primeironumero;
        int x;

        primeironumero = JOptionPane.showInputDialog( "Entre com o valor de x: " );
        x = Integer.parseInt( primeironumero );

        if (x>=0 && x<=10)
        {
            JOptionPane.showMessageDialog(null, "O valor " + x + " \u00e9 maior ou igual a 10 ", "Resultado", JOptionPane.PLAIN_MESSAGE );
        }

        System.exit( 0 );
    }
}
```

Estrutura de Seleção Composta

- ▶ A estrutura de seleção composta usa o comando **if-else**.
- ▶ O comando **else** quando associado ao **if**, executará uma instrução ou grupo de instruções entre chaves, se a expressão de teste do comando **if** for falsa.
- ▶ Sintaxe:

```
if (expressão de teste ou condição)
{
    // Comandos do bloco Verdade;
}
else
{
    // Comandos do bloco Falso;
}
```

Estrutura de Seleção Composta

```
import java.lang.*;
import javax.swing.JOptionPane; // import class JOptionPane


public class Exemplo3 {
    public static void main( String args[] ) throws java.lang.Exception
    {
        String primeironumero;
        int x;

        primeironumero = JOptionPane.showInputDialog( "Entre com o valor de x: " );
        x = Integer.parseInt( primeironumero );

        if (x>=0 && x<=10)
        {
            JOptionPane.showMessageDialog(null, "O valor " + x + "esta no intervalo de 0 a 10", "Resultado", JOptionPane.PLAIN_MESSAGE );
        }
        else
        {
            JOptionPane.showMessageDialog(null, "Valor fora do intervalo de 0 a 10", "Resultado", JOptionPane.PLAIN_MESSAGE );
        }

        System.exit( 0 );
    }
}
```

Repetindo Ações

- ▶ Às vezes é necessário repetir a mesma tarefa para chegar a um resultado final.
 - ▶ Por exemplo, para encher uma caixa, você coloca objetos dentro dela e então verifica se na mesma pode ser adicionado mais objetos.
 - ▶ Enquanto constatar que a caixa ainda não está cheia, você coloca mais um objeto dentro dela. A ação se repetirá até atingir o objetivo: encher a caixa.
 - ▶ Quando isso acontecer, você vai parar de colocar objetos na caixa.
 - ▶ Para repetir tarefas em um programa, evitando escrever várias vezes as mesmas instruções, você pode utilizar a Estruturas de Controle de Repetição, ou Estruturas de Repetição.
- 

Estrutura de Repetição

- ▶ Permite que um fluxo/instrução (ou um trecho de instruções) seja executado várias vezes.
- ▶ As repetições também são chamadas de looping, justamente porque têm por finalidade efetuar o processamento de um determinado trecho, tantas vezes quantas forem necessárias.

- ▶ **Tipos de Estruturas de Repetição:**

- 1– Repetição com Teste no Início (while);
- 2– Repetição com Variável de Controle (for).
- 3– Repetição com Teste no Final (do-while);

1 – Estrutura while

- ▶ **Repetição com Teste no Início:** também conhecido como estrutura **while**.
- ▶ Consiste em uma estrutura de controle (de fluxo de execução) que permite repetir diversas vezes um mesmo trecho do programa, porém, sempre verificando **antes** de cada execução se é “permitido” executar o mesmo trecho novamente.
- ▶ A estrutura **while** permite que um bloco, ou uma ação primitiva, seja repetida...
“**enquanto uma determinada <expressão de teste ou condição> for verdadeira**”.

1 – Estrutura while

- ▶ Sintaxe geral:

```
while (expressão de teste ou condição) {  
    comando1;  
    comando2;  
    .  
    comandoN;  
}
```

Assim, enquanto a expressão de teste for verdadeira, o corpo do laço **while** é executado uma vez e a expressão de teste é avaliada novamente. Esse ciclo de teste e execução é repetido até que a expressão de teste se torne falsa, então o laço termina e o controle do programa passa para a linha seguinte ao laço.

Uso de Contador

- ▶ Para que não ocorra erros na execução de um programa pode-se utilizar variáveis que desempenham o papel de **contador**.
- ▶ Um contador tem a função de controlar, ou contar, a quantidade de vezes que um bloco de comandos foi executado.
- ▶ Obs: na estrutura **while** é necessário inicializar (geralmente com o valor zero) o contador e depois dentro dessa estrutura (do loop) atualizar o mesmo, a partir por exemplo de um incremento.
- ▶ Sintaxe:

```
inicialização_do_contador
while (expressão de teste baseada no contador) {
    comandos;
    atualização_do_contador;
}
```

Uso de Contador

- ▶ **Resumo e sintaxe do contador**
- ▶ **CONTADOR:** é a variável cujo conteúdo é alterado pelo seu próprio valor adicionado (incrementado) ou subtraído (decrementado) de um valor constante.
- ▶ Usado para contar a quantidade de vezes que o programa executa as instruções de um loop.

Sintaxe:

Nome_do_contador = Nome_do_contador (+ ou -) k;
onde k é uma constante.

Exemplo: `cont = cont + 1;`



Observações do Uso de Contador

- ▶ OBS1: como o contador é uma variável ele deve ser declarado;
Ex: `int cont;`
- ▶ OBS2: o contador deve ser iniciado com um valor;
Ex: `cont=1;`
- ▶ OBS3: normalmente o contador deve ser usado na condição (expressão de teste) do loop;
Ex: `while (cont<=100)`
- ▶ OBS4: o contador deve ser atualizado dentro do loop para evitar um loop infinito.
Ex: `cont = cont + 1;`

```
int cont;  
cont=1;  
while(cont<=100)  
{  
    /*instruções;*/  
    cont=cont+1;  
}
```

Exemplo

```
import java.lang.*;
import javax.swing.JOptionPane; // import class JOptionPane

public class Exemplo4 {
    public static void main( String args[] ) throws java.lang.Exception
    {
        int x, cont, soma;

        cont=1;
        soma=0;
        while (cont<=10)
        {
            x = Integer.parseInt(JOptionPane.showInputDialog( "Digite um número: " ));
            soma=soma+x;
            cont++;
        }
        JOptionPane.showMessageDialog(null, "Total da soma: "+soma, "Resultado", JOptionPane.PLAIN_MESSAGE );
        System.exit( 0 );
    }
}
```

Exercícios

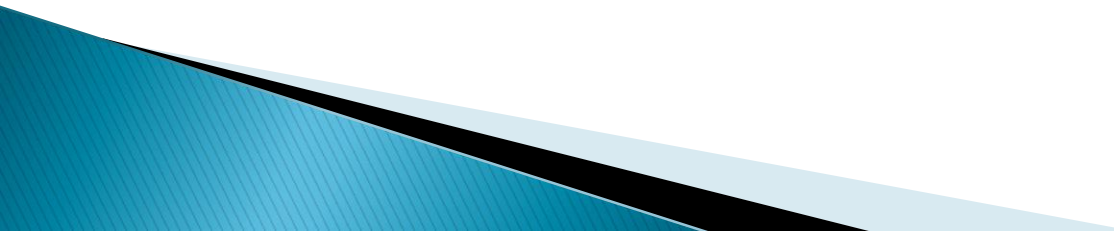
- 1) Faça um programa em linguagem Java para ler a idade de uma pessoa expressa em anos, meses e dias e imprime essa idade apenas em dias. Dessa forma, será necessário realizar a leitura de 3 valores inteiros. Obs: Para efetuar o cálculo considere que 1 ano tem 365 dias, e 1 mês tem 30 dias.
- 2) A imobiliária IMOV vende apenas terrenos retangulares. Faça um programa em Java para ler as dimensões de um terreno e depois exibir a área do mesmo. Para calcular a área de um retângulo use a fórmula: $\text{área} = b * h$

Exercícios

3) Considere a descrição a seguir:

“O custo de um carro novo para o consumidor é a soma do custo de fábrica com a percentagem do distribuidor (em cima do custo de fábrica) e dos impostos (também aplicados ao custo de fábrica)”.

Supondo que a percentagem do distribuidor seja de 32% e os impostos de 41%, escrever um programa em linguagem Java para ler o custo de fábrica de um carro e imprimir o custo ao consumidor.



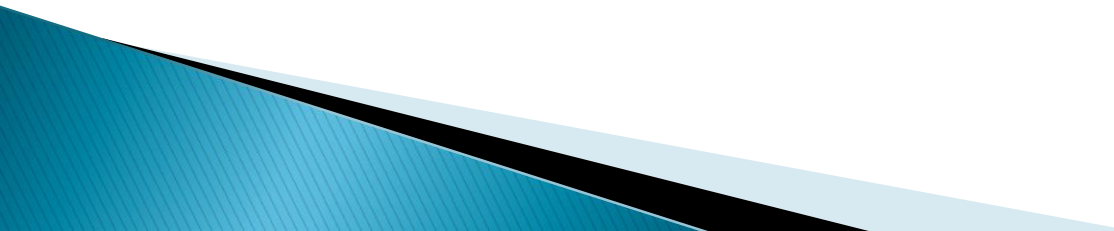
Exercícios

4) Crie um programa em Java que tendo como dados de entrada a **altura** (h) e o **sexo** de uma pessoa (1 – Masculino e 2 – Feminino), calcular seu peso ideal, utilizando as seguintes fórmulas:

- Para masculino

$$\text{PesoIdeal} = (72.7 * h) - 58;$$

- Para feminino

$$\text{PesoIdeal} = (62.1 * h) - 44.7;$$


Exercícios

5) Nesse ano ocorrerão as eleições para prefeitos, vices e vereadores. De acordo com o Tribunal Superior Eleitoral é considerada a idade da pessoa para que seja verificada a situação da pessoa, ou seja, se a mesma pode votar, ou se o seu voto é opcional, ou se a pessoa não pode votar. A tabela em questão e exibida a seguir:

Idade	Situação
Menor de 16 anos	Não pode votar
De 18 anos a 64 anos	Voto Obrigatório
Maior ou igual a 16 anos e menor que 18 anos	Voto Opcional
Maior ou igual a 65 anos	Voto Opcional

Exercícios

5) (continuação)


Considerando a necessidade de verificar a situação de uma pessoa, crie um programa em Java que dada a idade de uma pessoa, determine sua condição de Eleitor.

6) Crie um programa para ler um número como sendo a quantidade de computadores gerenciados por uma empresa. Verificar se o número (quantidade) de computadores é par ou ímpar.

Obs: para o número ser par, o resto de sua divisão por dois deve ser igual a zero. Usar o operador %.

ex: `resto = N%2;`

Exercícios

- 7) Criar um programa usando a estrutura while para imprimir os 100 primeiros múltiplos de 13.
 - 8) Criar um programa usando a estrutura while para ler 100 números inteiros e somar apenas os números pares. Ao final mostrar o total da soma.
 - 9) Uma árvore (árvore 1) tem 1,50 m e cresce 2 centímetros por ano. Uma outra árvore tem 1,10 m (árvore 2) e cresce 3 centímetros por ano. Construa um programa que calcule em quantos anos a árvore 2 será maior que árvore 1.
- 

Exercícios

10) Escrever um programa (usando a estrutura for) para ler 20 valores, um de cada vez, para a variável X. Contar quantos destes valores são negativos, mostrando os mesmos e escrevendo esta a quantidade ao final.