

ESTRUTURAS BÁSICAS E DE SELEÇÃO

Profa. Dra. Marta Adriana da Silva Cristiano

1 ESTRUTURAS BÁSICAS

1.1 JAVA APPLICATION PROGRAMMING INTERFACE – API DO JAVA

A linguagem de programação Java contém um conjunto de classes (programas reutilizáveis) predefinidas que facilitam o trabalho do programador, auxiliando-o a melhorar sem ter de reescrever muitos códigos que já foram anteriormente desenvolvidos e disponibilizados para reuso junto à linguagem Java.

Um exemplo que podemos usar é a biblioteca gráfica `javax.swing.JOptionPane`. Observe o código abaixo para imprimir a frase “Nosso primeiro código Java!”.

```
1 package javaapplication2;
2 public class JavaApplication2 {
3     public static void main(String[] args) {
4         System.out.println("Nosso primeiro código Java!");
5     }
6 }
7 }
```

O mesmo que o comando
ESCREVAL do VisualG

De acordo com o código acima, temos como resultado a seguinte mensagem:

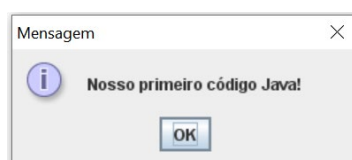
```
Nosso primeiro código Java!
CONSTRUÍDO COM SUCESSO (tempo total: 11 segundos)
```

Agora, observe o mesmo código utilizando a biblioteca gráfica `JOptionPane` (linha 2).

```
1 package javaapplication2;
2 import javax.swing.JOptionPane;
3 public class JavaApplication2 {
4     public static void main(String[] args) {
5         JOptionPane.showMessageDialog(null, "Nosso primeiro código Java!");
6     }
7 }
8 }
```

Comando para mostrar na tela
de acordo com a classe usada

Utilizando o comando `import`, é possível identificar a classe que se deseja utilizar. No nosso exemplo, estamos usando a classe `JOptionPane` do pacote `javax.swing` (linha 2). Veja como é mostrada, na tela, a mensagem “Nosso primeiro código Java!”:



Abaixo, seguem as APIs mais utilizadas na programação Java.

Pacote	Descrição
java.lang	Pacote usual da linguagem
java.util	Utilitários da linguagem
java.io	Pacote de entrada e saída
java.awt	Pacote de interface gráfica
java.net	Pacote de rede
java.sql	Pacote de banco de dados



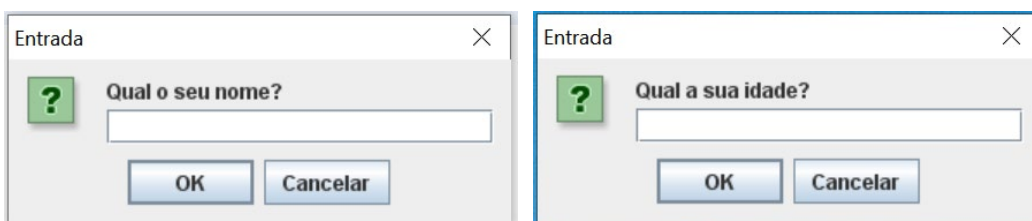
EXEMPLO

Suponha que desejamos solicitar ao usuário seu nome e idade e mostrar na tela os dados recebidos. Para tanto, precisaremos utilizar variáveis.

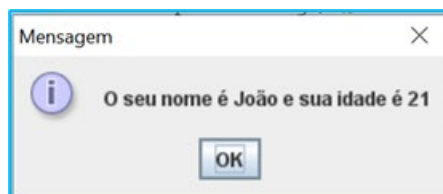
```
1 package javasimples;
2 import javax.swing.JOptionPane;
3 public class JavaSimples {
4     public static void main(String[] args) {
5         String nome;
6         int idade;
7
8         nome = JOptionPane.showInputDialog("Qual o seu nome? ");
9         idade = Integer.parseInt(JOptionPane.showInputDialog("Qual a sua idade? "));
10
11         JOptionPane.showMessageDialog(null, "O seu nome é " + nome + " e sua idade é " + idade);
12     }
13 }
14 }
```

Observe que foram criadas as variáveis **nome** do tipo *String* (linha 5) e **idade** do tipo *int* (linha 6). Como padrão da linguagem Java, o tipo é definido antes do nome da variável.

Ao executar, aparecerá uma janela solicitando os dados de entrada.



Logo após, imprime na tela os dados de saída.





ATIVIDADE DE ASSIMILAÇÃO 1

VAMOS EXERCITAR?

Sabendo que a área de um círculo é obtida pela fórmula $\text{Área} = \pi * R^2$, desenvolva um programa que receba o valor do raio e imprima a área do círculo. Lembre-se de que o valor de π é 3,1415.

2 ESTRUTURA CONDICIONAL

As estruturas condicionais, sejam elas simples ou encadeadas, executam uma sequência de comandos quando uma condição lógica ou matemática for verdadeira.

SE condição
ENTÃO comandos
SENÃO comandos

Na linguagem Java, a estrutura condicional utiliza o termo em inglês *if* (SE) e *else* (SE-NÃO). Quando a condição for verdadeira, executa a primeira sequência de instruções, caso contrário, executa a segunda sequência de instruções.

```
if (condição) {  
    <sequência de comandos>  
} else {  
    <sequência de comandos>  
}
```

2.1 ESTRUTURA CONDICIONAL ENCADEADA

Na estrutura condicional encadeada, uma condição pode levar à outra condição de tomada de decisão. Para tanto, as estruturas de seleção **(SE)** são disponibilizadas uma dentro da outra.

SE condição
ENTÃO
 SE condição
 ENTÃO comandos
 SENÃO comandos
SENÃO comandos

A linguagem de programação Java utiliza os termos em inglês *if* e *else* para a formação da estrutura condicional encadeada.

```
If (condição) {  
    If (condição) {  
        <sequência de comandos>  
    }  
    else {  
        <sequência de comandos>  
    }  
}  
else  
    <sequência de comandos>  
}
```



EXEMPLO

A empresa XXX solicita um programa que, a partir da digitação do salário atual de seu funcionário, calcule o novo salário acrescido de bonificação e vale alimentação, de acordo com a tabela abaixo:

Salário	Bonificação
Até R\$1.000,00	5% do salário atual
Entre R\$1.000,00 e R\$2.000,00	12% do salário atual
Acima de R\$2.000,00	Nenhuma bonificação
Salário	Vale alimentação
Até R\$1.500,00	R\$150,00
Acima de R\$1.500,00	R\$ 100,00

```
1 package javasimples;  
2 import java.util.*;  
3 public class JavaSimples {  
4     public static void main(String[] args) {  
5         float sal, novo_sal, boni=0, vale=0;  
6         Scanner entrada;  
7         //Instancia o objeto da classe Scanner  
8         entrada = new Scanner(System.in);  
9         //Mostra mensagem solicitando o valor do salário  
10        System.out.println("Digite o valor do salário: ");  
11        //Recebe o valor do salário  
12        sal = entrada.nextFloat();  
13        // Calcula a bonificação  
14        if (sal <= 1000)  
15            boni= sal * 5/100;  
16        else  
17            if (sal < 2000)  
18                boni=sal * 12/100;  
19            else  
20                boni=0;  
21        // Calcula o auxilio escola  
22        if (sal <= 1500)  
23            vale=150;  
24        else  
25            vale=100;  
26        //Calcula o novo salário  
27        novo_sal = sal + boni + vale;  
28        //Mostra o valor do novo salário  
29        System.out.println("Novo salário = " + novo_sal);  
30    }  
31 }
```

A classe *Scanner* facilita a entrada de dados pelo Console.



ATIVIDADE DE ASSIMILAÇÃO 2

VAMOS EXERCITAR?

1. Os alunos da Escola ABC têm três notas: uma nota referente ao trabalho da disciplina, uma nota da prova e outra nota de participação. Calcule e mostre a média aritmética simples, seguida da mensagem constante na tabela abaixo. Para os alunos que ficarem para recuperação, apresente a nota que precisam tirar para serem aprovados, considerando que a média para aprovação é 6,0.

Média aritmética	Mensagem
0,0 – 3,0	Reprovado
3,0 – 6,0	Recuperação
6,0 – 10,0	Aprovado

2.2 ESTRUTURA CONDICIONAL DE MÚLTIPLA ESCOLHA

A estrutura de seleção de múltipla escolha é uma especialização da estrutura de seleção encadeada. Esta estrutura oferece um conjunto de opções de escolha.

```
ESCOLHA variável  
CASO OPÇÃO: comandos  
CASO OPÇÃO: comandos  
CASO OPÇÃO: comandos  
FIMESCOLHA
```

Na linguagem Java, a palavra ESCOLHA é substituída pelo termo em inglês *SWITCH*, e CASO substituído pelo termo em inglês *CASE*.

```
Switch (<variável>){  
    Case <opção 1>:  
        [sequência de comandos];  
        Break;  
    Case <opção 2>:  
        [sequência de comandos];  
        Break;  
    Case <opção 3>:  
        [sequência de comandos];  
        Break;  
    Default:  
        [sequência de comandos];  
}
```



EXEMPLO

Maria é aluna de Análise e Desenvolvimento de Sistemas, e uma de suas atividades é desenvolver um código em que o usuário poderá somar dois números ou calcular a raiz quadrada de um número. Para tanto, será necessário criar um menu de opções para que o usuário possa escolher o processamento desejado.

```
1 package javasimples;
2 import java.util.*;
3 import java.text.*;
4 public class JavaSimples {
5     public static void main(String[] args) {
6         double num1, num2, soma, raiz;
7         int op;
8         Scanner entrada;
9         entrada = new Scanner(System.in);
10        //Menu
11        System.out.println("1- Somar dois números");
12        System.out.println("2- Raiz quadrada de um número");
13        System.out.print("Digite sua opção: ");
14        op = entrada.nextInt(); //Recebe a opção do usuário
15        switch (op)
16        {
17            case 1 : System.out.print("Digite um valor para o primeiro número: ");
18                    num1 = entrada.nextDouble();
19                    System.out.print("Digite um valor para o segundo número: ");
20                    num2 = entrada.nextDouble();
21                    soma = num1 + num2;
22                    System.out.println("A soma de " + num1 + " + " + num2 + " = " + soma);
23                    break;
24            case 2 : System.out.print("Digite um número: ");
25                    num1 = entrada.nextDouble();
26                    raiz = Math.sqrt(num1);
27                    System.out.println("A raiz quadrada de " + num1 + " é " + raiz);
28                    break;
29            default : System.out.println("Opção inválida!");
30        }
31    }
32 }
```



ATIVIDADE DE ASSIMILAÇÃO 3

1. Desenvolva um programa que simule uma calculadora com as operações básicas: soma, subtração, multiplicação e divisão. O usuário deverá entrar com dois números para cada operação. Caso a escolha seja divisão, verifique se o divisor (número 1) é maior que o dividendo (número 2), caso contrário, imprima uma mensagem de erro.
2. Faça um programa que receba um número do usuário, calcule e verifique se é par ou ímpar.
3. A empresa XXX deve dar um aumento aos seus colaboradores, variando o percentual de acordo com o cargo. Faça um pseudocódigo em que, a partir do código da função de cada trabalhador, sejam apresentados o cargo, o salário atual, o aumento e o novo salário (observe tabela abaixo).

Código	Função	Aumento
1	Serviços Gerais	50%
2	Atendente	30%
3	Supervisor	20%
4	Gerente	10%

4. A rede de supermercados BEMBÃO precisa reajustar os preços de seus produtos de acordo com a seguinte tabela:

Valor atual	Venda média (mensal)	Perc. aumento	Perc. desconto
< R\$ 50,00	< 500 unidades	10 %	
>= R\$ 50,00 e < R\$ 100,00	>= 500 unid. e < 1000 unid.	20%	
> R\$ 100,00	> 1000 unid.		20 %

Faça um pseudocódigo que receba o preço atual, a venda média do produto, calcule e mostre o novo preço.

CONCLUSÃO

Uma das principais características da linguagem de programação Java é que ela contém um conjunto de classes predefinidas (orientando ao reuso de códigos) que facilitam o trabalho do programador. Essas classes formam uma biblioteca de funções prontas que podem ser importadas para o programa a ser desenvolvido, economizando tempo, pois muitas funções de que vamos precisar já estão criadas e funcionais.

Para os códigos que devem ser desenvolvidos, cujas classes ainda não estão disponibilizadas nas bibliotecas, as estruturas condicionais são essenciais para começar. As estruturas de seleção executam uma sequência de comandos quando uma condição lógica ou matemática for verdadeira, sejam elas simples, compostas, encadeadas ou de múltipla escolha.

Para tanto, a linguagem de programação Java utiliza os termos em inglês *if* e *else* para a formação da estrutura condicional. Além disso, quando utilizamos estruturas de múltipla escolha, a palavra ESCOLHA é substituída pelo termo em inglês *SWITCH*, e CASO substituído pelo termo em inglês *CASE*.

Não se esqueça de que a escolha do tipo de estrutura condicional determinará a qualidade do seu código.

APORTES BIBLIOGRÁFICOS

ASCENCIO, Ana Fernandes Gomes. **Fundamentos da programação de computadores**. São Paulo: Pearson Prentice Hall, 2007.

DEITEL, Paul; DEITEL, Harvey. **Java: como programar**. 6. ed. São Paulo: Pearson Prentice Hall, 2005.

MANZANO, José Augusto N. G. **Programação de computadores com Java**. São Paulo: Erica, 2014.

PUGA, Sandra; RISSETI, Gerson. **Lógica de programação e estrutura de dados, com aplicações em Java**. 2. ed. São Paulo: Pearson Prentice Hall, 2009.

SCHILDT, Herbert. **Java para iniciantes**. 6. ed. Porto Alegre: Bookman, 2015.