

Módulos e Matrizes

Bem-vindo ao estudo sobre Matrizes e Módulos no Java. Neste módulo serão mostrados exemplos práticos em um programa Java. Vamos conhecer estes exemplos práticos sobre o que foi estudado durante toda a disciplina. Lembrando que, a prática é indispensável para você aprender a resolver problemas através da lógica de programação. Portanto, você será capaz de desenvolver algoritmos que resolvam problemas, e posteriormente, desenvolver um programa a partir desses algoritmos.

Matrizes

As matrizes são um conjunto de informações do mesmo tipo. A matriz mais usual é a matriz de uma dimensão que é conhecido como vetor ou array. Você pode desenvolver matrizes de várias dimensões, porém, quanto mais dimensões, mais complicado de entender a organização dos dados.

Vamos exemplificar o uso de matrizes por meio do desenvolvimento de um programa java que declara um vetor unidimensional de 5 posições e uma matriz de dimensões 2x3 (duas linhas e 3 colunas), colocando informações nessas matrizes e apresentando os resultados.

```

1 //salvar como ProgMatriz.java
2 import javax.swing.*;
3
4 class ProgMatriz
5 {
6     public static void main (String entrada[])
7     {
8         int vetor[] = {2, 4, 6, 8, 10};
9         int matriz[][] = new int [2][3];
10        String msg = "vetor = ";
11
12        //vetor = ;
13        for (int i = 0 ; i < vetor.length ; i++)
14        {
15            msg = msg + vetor[i] + " ";
16        }
17        JOptionPane.showMessageDialog(null, msg);
18
19        msg = "Matriz = \n\n";
20        for (int i = 0 ; i < matriz.length ; i++)
21        {
22            for (int j = 0 ; j < matriz[0].length ; j++)
23            {
24                matriz[i][j] = Integer.parseInt(JOptionPane.showInputDialog("Digite um inteiro para posicao " + i + " e " + j));
25                msg = msg + matriz[i][j] + " ";
26            }
27            msg = msg + "\n";
28        }
29
30        JOptionPane.showMessageDialog(null, msg);
31        System.exit(0);
32    }
33 }

```

Na linha 8, criamos a matriz `vetor[]` de uma única dimensão, já colocando os valores {2, 4, 6, 7, 10} na inicialização do vetor. Na linha 9, criamos um vetor de duas dimensões, duas linhas e três colunas.

Na linha 15, o `vetor[i]` é acessado e cada uma das suas informações são concatenados em `msg` e ao final da repetição de acesso do vetor, essa mensagem é apresentada.

Nas linhas de 20 até 28, a `matriz[i][j]` recebe informações digitadas pelo usuário e suas informações são acessadas e concatenadas em `msg` que ao final da repetição encadeada do `for`, o conteúdo de `msg` é apresentada.



```
C:\teste>javac ProgMatriz.java
```

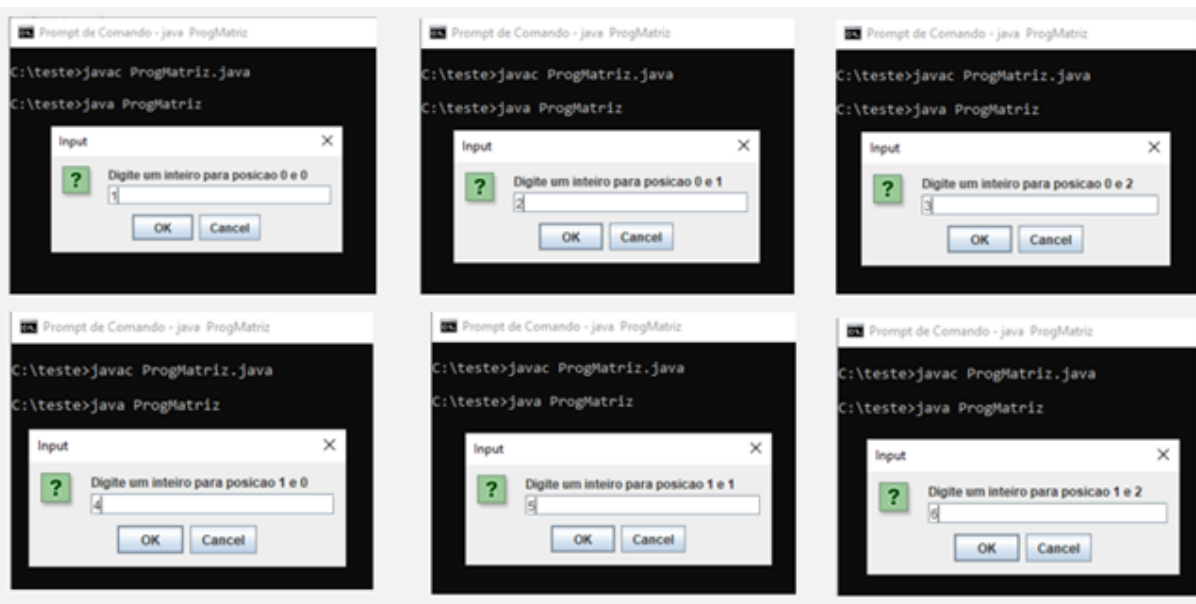
```
C:\teste>java ProgMatriz
```

Message



vetor = 2 4 6 8 10

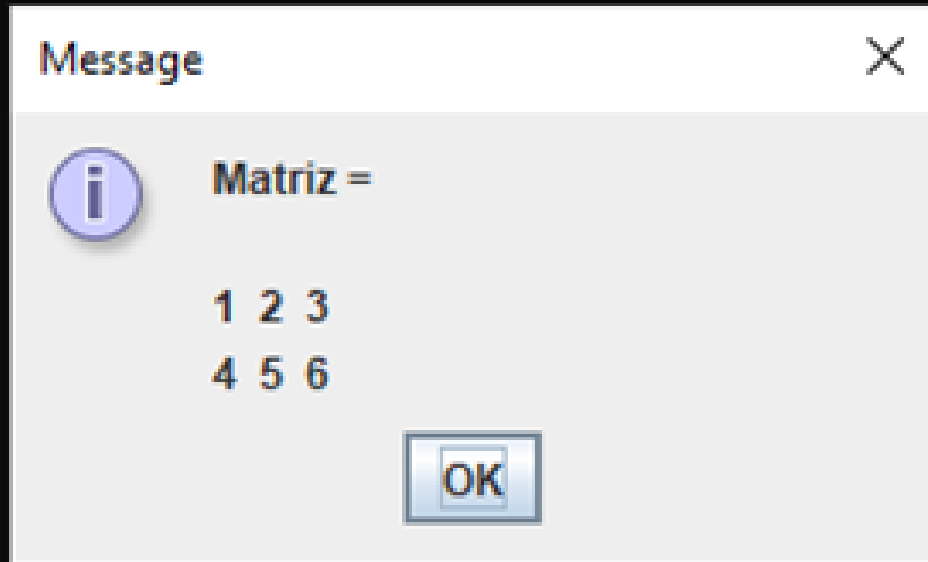
OK





```
C:\teste>javac ProgMatriz.java
```

```
C:\teste>java ProgMatriz
```



Modularização

A modularização serve para dividir um programa maior em pequenas tarefas para serem executadas quando chamadas. Os módulos no Java são chamados de métodos e temos quatro tipos de métodos no Java.

Temos o método procedimento sem parâmetro que não recebe argumentos e não retorna valores. O método procedimento com parâmetro recebe argumentos e não retorna valores.

Temos a função sem parâmetro que não recebe argumentos e retorna algum valor. O método função com parâmetro recebe argumentos e retorna valor.

Vamos exemplificar desenvolvendo um programa que, utilizando os métodos função e procedimento, calcula a soma, o produto, a diferença e o resto da divisão de dois números inteiros, mostrando os resultados.

```
6 public static void soma ()
7 {
8     int n1, n2;
9     n1 = Integer.parseInt(JOptionPane.showInputDialog("Digite um numero"));
10    n2 = Integer.parseInt(JOptionPane.showInputDialog("Digite outro numero"));
11
12    JOptionPane.showMessageDialog(null, "A soma eh "+ (n1+n2));
13 }
14
15 public static void subtracao (int x, int y)
16 {
17     int s;
18
19     s = x - y;
20
21    JOptionPane.showMessageDialog(null, "A diferenca eh "+ s);
22 }
23
24 public static int produto ()
25 {
26     int n1, n2;
27     n1 = Integer.parseInt(JOptionPane.showInputDialog("Digite um numero"));
28     n2 = Integer.parseInt(JOptionPane.showInputDialog("Digite outro numero"));
29
30     return (n1 * n2);
31 }
32
33 public static double divisao (int x, int y)
34 {
35     double d;
36
37     d = (double)x / (double)y;
38
39     return d;
40 }
```

Da linha 6 à linha 13, temos um procedimento soma() sem argumentos como parâmetros sem retorno de valores.

Da linha 15 à linha 22, temos um procedimento subtração (int x, int y) com os argumentos x e y no parâmetro e sem retorno de valores.

Da linha 24 à linha 31, temos uma função produto() sem argumentos como parâmetro e com retorno do resultado de n1*n2.

Da linha 33 à linha 40, temos uma função divisão(int x, int y) com os argumentos x e y no parâmetro e com retorno do valor da variável d.

```

1 //salvar como ProgMod.java
2 import javax.swing.*;
3
4 class ProgMod
5 {
6     public static void soma ()
7     {
14
15     public static void subtracao (int x, int y)
16     {
23
24     public static int produto ()
25     {
32
33     public static double divisao (int x, int y)
34     {
41     public static void main (String entrada[])
42     {
43         int n1, n2, s;
44         double r;
45
46         soma();
47         n1 = Integer.parseInt(JOptionPane.showInputDialog("Digite um numero"));
48         n2 = Integer.parseInt(JOptionPane.showInputDialog("Digite outro numero"));
49         subtracao(n1,n2);
50         s = produto();
51         JOptionPane.showMessageDialog(null, "O produto eh "+ s);
52         r = divisao(n1,n2);
53         JOptionPane.showMessageDialog(null, "A divisao eh "+ r);
54         System.exit(0);
55     }
56 }

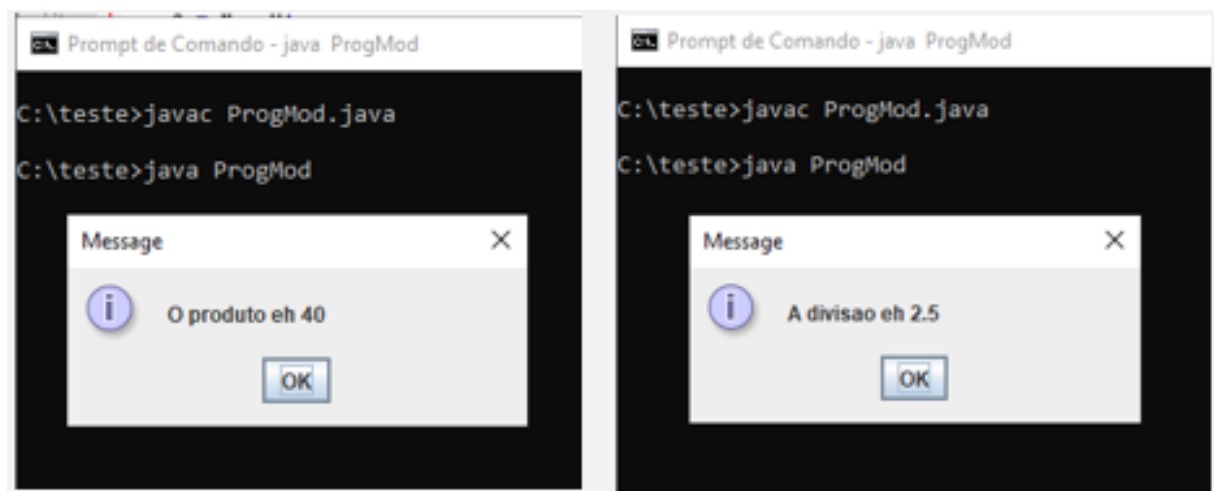
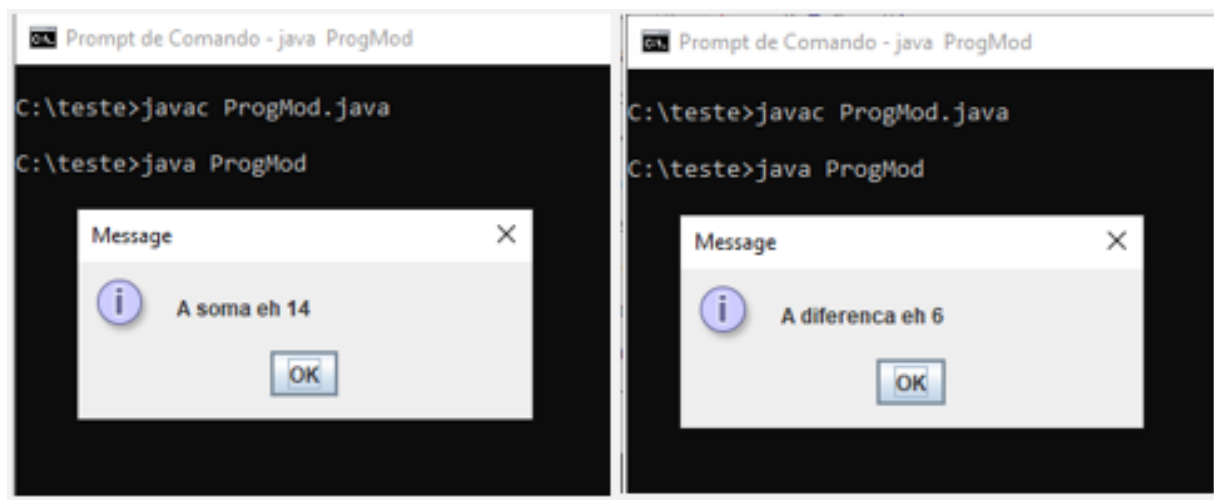
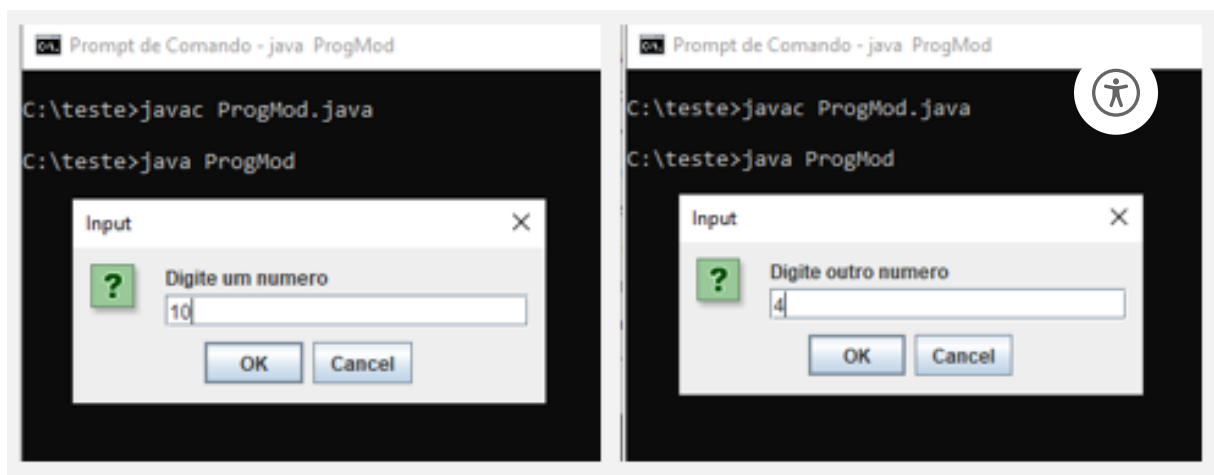
```

Na linha 46, temos a chamada do método procedimento sem argumentos no parâmetro soma() e sem retorno de valores.


Na linha 49, temos a chamada do método procedimento com argumentos n1 e n2 no parâmetro subtração(n1, n2) e sem retorno de valores.

Na linha 50, temos a chamada do método função sem argumentos no parâmetro s=produto() e com retorno do conteúdo da variável s.

Na linha 52, temos a chamada do método função com argumentos n1 e n2 no parâmetro r=divisão(n1,n2) e com retorno do conteúdo da variável r.



Atividade extra

Assista ao filme “Ela” Estrelado por Joaquin Phoenix, Amy Adams e Scarlett Johansson, esse filme de Spike Jonzen levou Oscar de melhor roteiro  ginal em 2014. Mostra o drama de um escritor solitário que se apaixona pelo sistema operacional projetado para atender a todas as suas necessidades. O desenvolvimento desse romance improvável e seu desfecho, você vai ter de assistir para conferir.

Referência Bibliográfica

- GUEDES, S. (Org.). **Lógica de programação algorítmica**. Pearson: 2014.
- MANZANO, J. A. N. G.; OLIVEIRA, J. F. **Estudo Dirigido de Algoritmos**. 15. ed. São Paulo: Érica, 2012
- PUGA, S.; RISSETTI, G. **Lógica de programação e estruturas de dados, com aplicações em Java**. Pearson: 2016.
- RIBEIRO, J. A. **Introdução à programação e aos algoritmos**. 1. ed. Rio de Janeiro: LTC, 2019

Atividade Prática – Aula 16

Título da Prática: Módulo e Matrizes

Aulas Envolvidas nesta Prática: Módulos e Matrizes no Java

Objetivos: Praticar lógica de programação e desenvolvimento de algoritmos e programas.

Materiais, Métodos e Ferramentas: Para realizar este exercício, vamos utilizar Bloco de nota e Prompt de comandos



Atividade Prática

Desenvolva um programa Java que declara matriz unidimensional (Vetor) de inteiros, receba números inteiros num vetor de 5 posições, calcula a somatória (ex.: $s = s + \text{vet}[i]$) e a produtória (ex.: $p = p * \text{vet}[i]$) desses números com um método similar a um procedimento e um método similar a uma função, ambos utilizando parâmetros, por fim, mostre essas informações na tela.

Após desenvolver seu código conforme a descrição acima, copie e cole na caixa de texto (a resposta da Atividade Prática sempre será em código (Java)).

Gabarito Atividade Prática

```

1 //salvar como Prog08.java
2 import javax.swing.*;
3
4 class Prog08
5 {
6     public static void soma (int vet[])
7     {
8         int s=0;
9         for (int i=0 ; i<vet.length; i++)
10         {
11             s = s + vet[i];
12         }
13         JOptionPane.showMessageDialog(null, "A somatoria eh " + s);
14     }
15
16     public static int produto (int vet[])
17     {
18         int p=1;
19         for (int i=0 ; i<vet.length; i++)
20         {
21             p = p * vet[i];
22         }
23         return p;
24     }
25     public static void main (String entrada[])
26     {
27         int s=0, vetor[] = {2, 4, 6, 8, 10};
28         int r;;
29
30         soma(vetor);
31         r = produto(vetor);
32         JOptionPane.showMessageDialog(null, "A produtoria eh " + r);
33
34         System.exit(0);
35     }
36 }

```

//salvar como Prog08.java

import javax.swing.*;

class Prog08

{

public static void soma (int vet[])

{

```
int s=0;
```



```
for (int i=0 ; i<vet.length; i++)
```

```
{
```

```
    s = s + vet[i];
```

```
}
```

```
OptionPane.showMessageDialog(null, "A somatoria eh " + s);
```

```
}
```

```
public static int produto (int vet[])
```

```
{
```

```
    int p=1;
```

```
    for (int i=0 ; i<vet.length; i++)
```

```
    {
```

```
        p = p * vet[i];
```

```
    }
```

```
    return p;
```

```
}
```

```
public static void main (String entrada[])
```

```
{
```

```
int s=0, vetor[] = {2, 4, 6, 8, 10};
```



```
int r;;
```

```
soma(vetor);
```

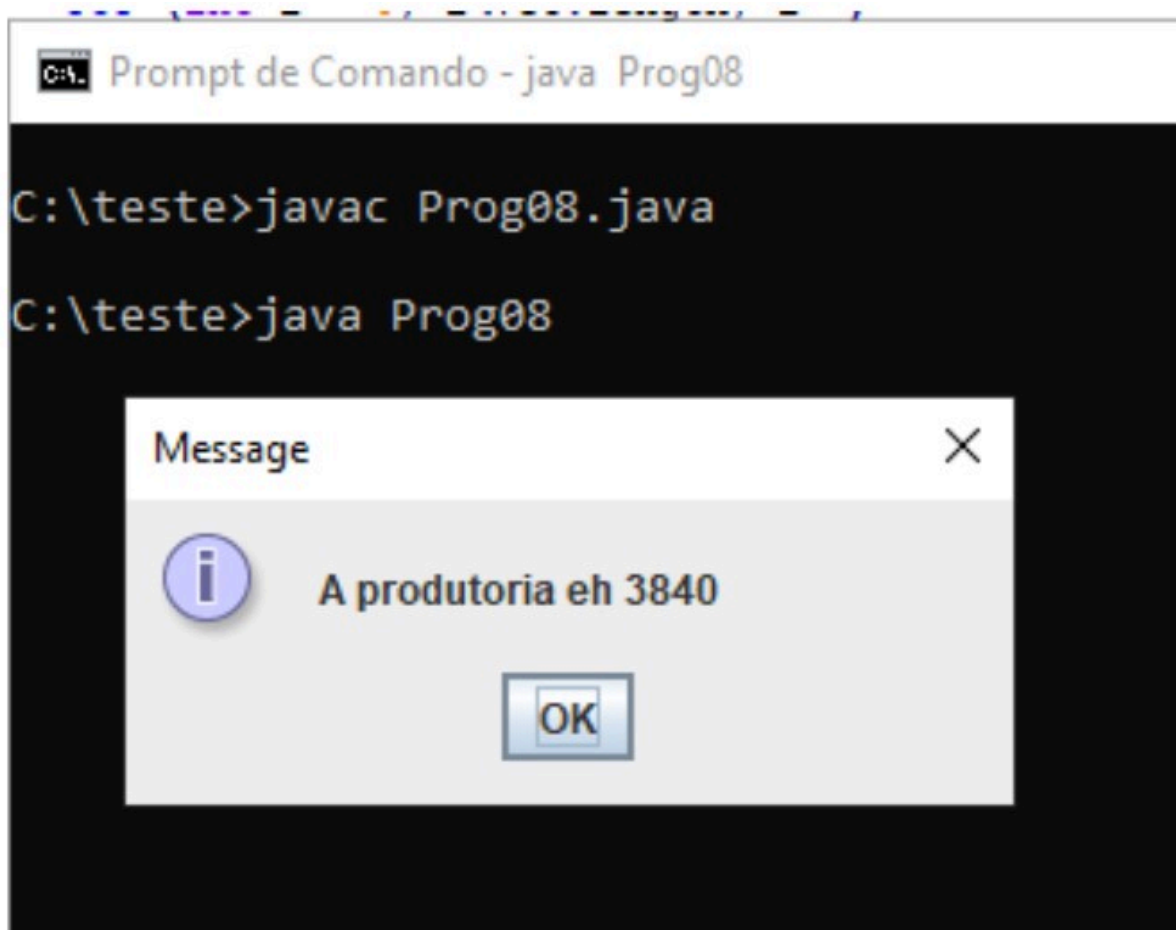
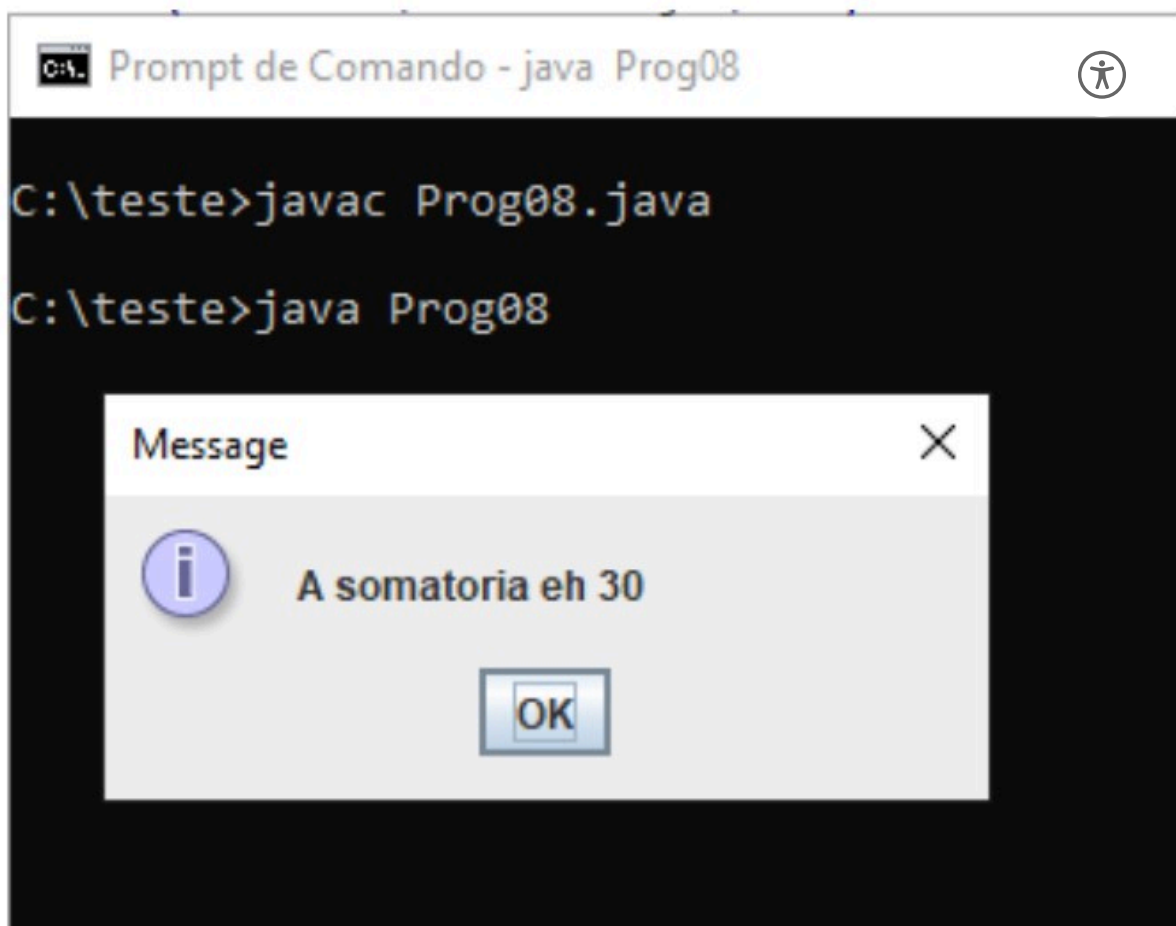
```
r = produto(vetor);
```

```
JOptionPane.showMessageDialog(null, "A produtoria eh " + r);
```

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

```
}
```

```
}
```



Ir para exercício

