

DS2

Estruturas de repetição

Estrutura "enquanto"

```
while ( condição ) {  
    comando 1  
    comando 2  
}
```

DS2

Estruturas de repetição

Digitar um número e mostrar sua raiz quadrada com três casas decimais, depois repetir o procedimento. Quando o usuário digitar um número negativo (podendo inclusive ser na primeira vez), mostrar uma mensagem "Número negativo" e terminar o programa.

```
Digite um número: 25
5.000
Digite outro número: 10
3.162
Digite outro número : 9
3.000
Digite outro número : -4
Número negativo!
```

```
using System;
using System.Globalization;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Console.Write("Digite um número: ");
            double x = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            while (x >= 0.0) {
                double raiz = Math.Sqrt(x);
                Console.WriteLine(raiz.ToString("F3", CultureInfo.InvariantCulture));
                Console.Write("Digite outro número: ");
                x = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            }

            Console.WriteLine("Número negativo!");
        }
    }
}
```

DS2

Exercícios

Escreva um programa que repita a leitura de uma senha até que ela seja válida. Para cada leitura de senha incorreta informada, escrever a mensagem "Senha Invalida". Quando a senha for informada corretamente deve ser impressa a mensagem "Acesso Permitido" e o algoritmo encerrado. Considere que a senha correta é o valor 2002.

Exemplo:

Entrada:	Saída:
2200	Senha Invalida
1020	Senha Invalida
2022	Senha Invalida
2002	Acesso Permitido

DS2

Exercícios

Escreva um programa para ler as coordenadas (X,Y) de uma quantidade indeterminada de pontos no sistema cartesiano. Para cada ponto escrever o quadrante a que ele pertence. O algoritmo será encerrado quando pelo menos uma de duas coordenadas for NULA (nesta situação sem escrever mensagem alguma).

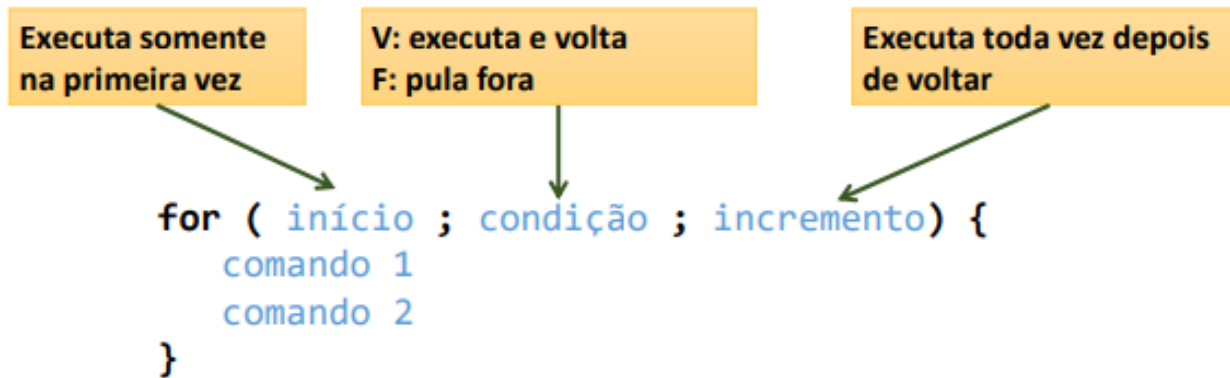
Exemplo:

Entrada:	Saída:
2 2	primeiro
3 -2	quarto
-8 -1	terceiro
-7 1	segundo
0 2	

DS2

Estruturas de repetição

Estrutura "para"



DS2

Estruturas de repetição

Digitar um número N e depois N valores inteiros. Mostrar a soma dos N valores digitados.

```
Quantos números inteiros você vai digitar? 3
Valor #1: 10
Valor #2: 7
Valor #3: 8
Soma = 25
```

```
using System;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Console.Write("Quantos números inteiros você vai digitar? ");
            int N = int.Parse(Console.ReadLine());

            int soma = 0;
            for (int i = 1; i <= N; i++) {
                Console.Write("Valor #{0}: ", i);
                int valor = int.Parse(Console.ReadLine());
                soma += valor;
            }

            Console.WriteLine("Soma = " + soma);
        }
    }
}
```

DS2

Estruturas de repetição

Leia um valor inteiro X ($1 \leq X \leq 1000$). Em seguida mostre os ímpares de 1 até X , um valor por linha, inclusive o X , se for o caso.

Exemplo:

Entrada:	Saída:
8	1 3 5 7

DS2

Estruturas de repetição

Leia um valor inteiro N. Este valor será a quantidade de valores inteiros X que serão lidos em seguida. Mostre quantos destes valores X estão dentro do intervalo [10,20] e quantos estão fora do intervalo, mostrando essas informações conforme exemplo (use a palavra "in" para dentro do intervalo, e "out" para fora do intervalo).

Exemplo:

Entrada:	Saída:
5	2 in
14	3 out
123	
10	
-25	
32	

DS2

Estruturas de repetição

Fazer um programa para ler um número inteiro positivo N. O programa deve então mostrar na tela N linhas, começando de 1 até N. Para cada linha, mostrar o número da linha, depois o quadrado e o cubo do valor, conforme exemplo.

Exemplo:

Entrada:	Saída:
5	1 1 1 2 4 8 3 9 27 4 16 64 5 25 125

DS2

Funções

- Representam um processamento que possui um significado
 - `Math.Sqrt(double)`
 - `Console.WriteLine(string)`
- Principais vantagens: modularização, delegação e reaproveitamento
- Dados de entrada e saída
 - Funções podem receber dados de entrada (parâmetros ou argumentos)
 - Funções podem ou não retornar uma saída
- Em orientação a objetos, funções em classes recebem o nome de "métodos"

DS2

Funções

```
using System;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Console.WriteLine("Digite três números:");
            int n1 = int.Parse(Console.ReadLine());
            int n2 = int.Parse(Console.ReadLine());
            int n3 = int.Parse(Console.ReadLine());

            double resultado = Maior(n1, n2, n3);

            Console.WriteLine("Maior = " + resultado);
        }

        static int Maior(int a, int b, int c) {
            int m;
            if (a > b && a > c) {
                m = a;
            }
            else if (b > c) {
                m = b;
            }
            else {
                m = c;
            }
            return m;
        }
    }
}
```

DS2

Exercício

EXERCÍCIO 01:

Fazer um programa para ler os dados de duas pessoas, depois mostrar o nome da pessoa mais velha.

Exemplo:

```
Dados da primeira pessoa:  
Nome: Maria  
Idade: 17  
Dados da segunda pessoa:  
Nome: Joao  
Idade: 16  
Pessoa mais velha: Maria
```

DS2

Exercício

EXERCÍCIO 02:

Fazer um programa para ler nome e salário de dois funcionários. Depois, mostrar o salário médio dos funcionários.

Exemplo:

```
Dados do primeiro funcionário:  
Nome: Carlos Silva  
Salário: 6300.00  
Dados do segundo funcionário:  
Nome: Ana Marques  
Salário: 6700.00  
Salário médio = 6500.00
```

DS2

Orientação a Objetos

Problema exemplo

Fazer um programa para ler as medidas dos lados de dois triângulos X e Y (suponha medidas válidas). Em seguida, mostrar o valor das áreas dos dois triângulos e dizer qual dos dois triângulos possui a maior área.

A fórmula para calcular a área de um triângulo a partir das medidas de seus lados a, b e c é a seguinte (fórmula de Heron):

$$area = \sqrt{p(p-a)(p-b)(p-c)} \quad \text{onde} \quad p = \frac{a+b+c}{2}$$

Exemplo:

Entre com as medidas do triângulo X:

3.00

4.00

5.00

Entre com as medidas do triângulo Y:

7.50

4.50

4.02

Área de X = 6.0000

Área de Y = 7.5638

Maior área: Y

DS2

Orientação a Objetos

```
using System;
using System.Globalization;

namespace Course {
    class Program {
        static void Main(string[] args) {

            double xA, xB, xC, yA, yB, yC;

            Console.WriteLine("Entre com as medidas do triângulo X:");
            xA = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            xB = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            xC = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            Console.WriteLine("Entre com as medidas do triângulo Y:");
            yA = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            yB = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            yC = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            double p = (xA + xB + xC) / 2.0;
            double areaX = Math.Sqrt(p * (p - xA) * (p - xB) * (p - xC));

            p = (yA + yB + yC) / 2.0;
            double areaY = Math.Sqrt(p * (p - yA) * (p - yB) * (p - yC));

            Console.WriteLine("Área de X = " + areaX.ToString("F4", CultureInfo.InvariantCulture));
            Console.WriteLine("Área de Y = " + areaY.ToString("F4", CultureInfo.InvariantCulture));

            if (areaX > areaY) {
                Console.WriteLine("Maior área: X");
            }
            else {
                Console.WriteLine("Maior área: Y");
            }
        }
    }
}
```

DS2

Orientação a Objetos

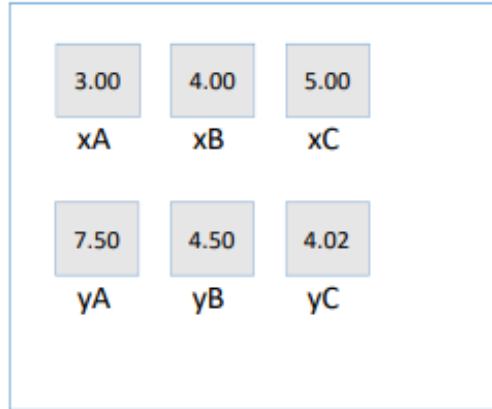
Triângulo é uma entidade com três atributos: a, b, c.

Estamos usando três variáveis distintas para representar cada triângulo:

```
double xA, xB, xC, yA, yB, yC;
```

Para melhorar isso, vamos usar uma CLASSE para representar um triângulo.

Memória:

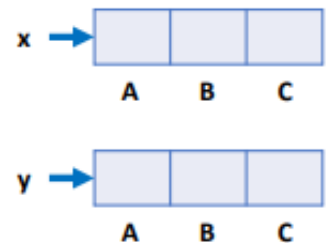


```
namespace Course {  
    class Triangulo {  
  
        public double A;  
        public double B;  
        public double C;  
  
    }  
}
```

```
double xA, xB, xC, yA, yB, yC;
```



```
Triangulo x, y;  
x = new Triangulo();  
y = new Triangulo();
```



DS2

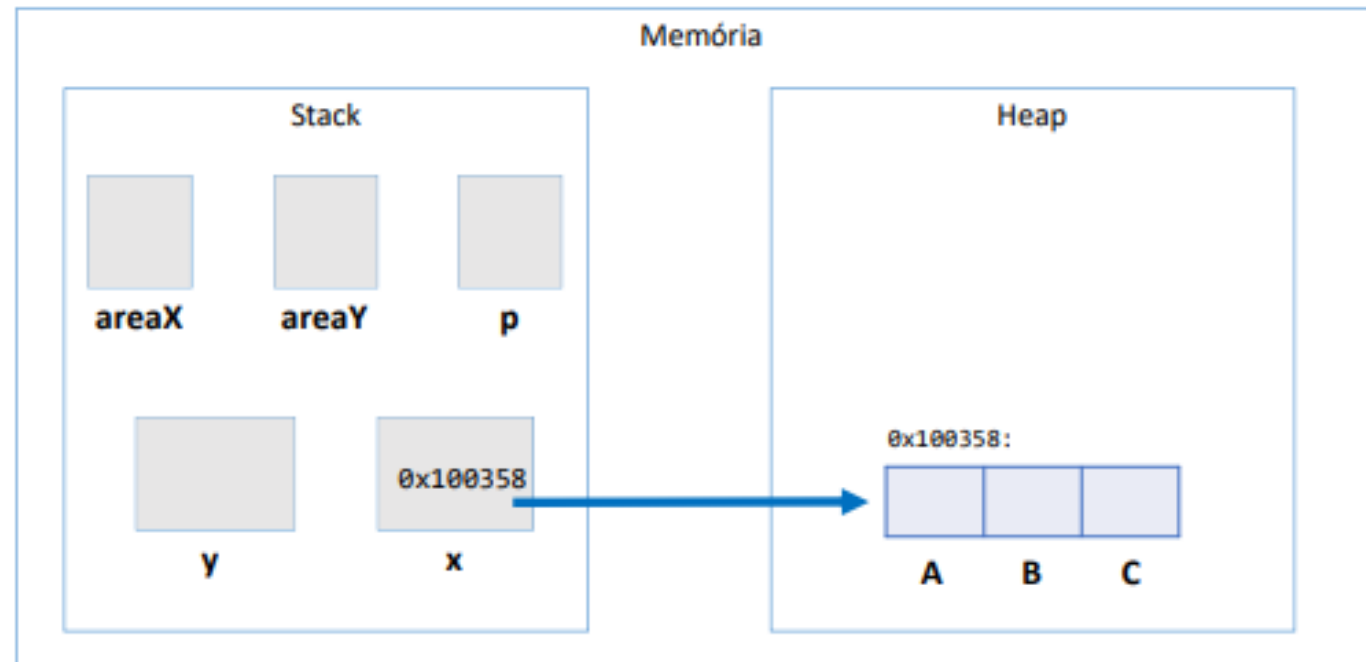
Orientação a Objetos

Instanciação

(alocação dinâmica de memória)

```
double areaX, areaY, p;  
Triangulo x, y;
```

```
x = new Triangulo();
```



DS2

Orientação a Objetos

```
using System;
using System.Globalization;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Triangulo x, y;

            x = new Triangulo();
            y = new Triangulo();

            Console.WriteLine("Entre com as medidas do triângulo X:");
            x.A = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            x.B = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            x.C = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            Console.WriteLine("Entre com as medidas do triângulo Y:");
            y.A = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            y.B = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            y.C = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            double p = (x.A + x.B + x.C) / 2.0;
            double areaX = Math.Sqrt(p * (p - x.A) * (p - x.B) * (p - x.C));

            p = (y.A + y.B + y.C) / 2.0;
            double areaY = Math.Sqrt(p * (p - y.A) * (p - y.B) * (p - y.C));

            Console.WriteLine("Área de X = " + areaX.ToString("F4", CultureInfo.InvariantCulture));
            Console.WriteLine("Área de Y = " + areaY.ToString("F4", CultureInfo.InvariantCulture));

            if (areaX > areaY) {
                Console.WriteLine("Maior área: X");
            }
            else {
                Console.WriteLine("Maior área: Y");
            }
        }
    }
}
```

DS2

Orientação a Objetos

Classe

- É um tipo estruturado que pode conter (membros):
 - Atributos (dados / campos)
 - Métodos (funções / operações)
- A classe também pode prover muitos outros recursos, tais como:
 - Construtores
 - Sobrecarga
 - Encapsulamento
 - Herança
 - Polimorfismo

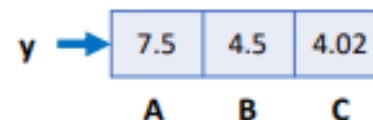
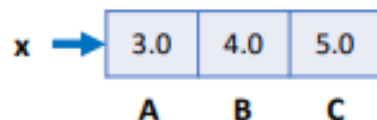
DS2

Orientação a Objetos

Classes, objetos, atributos

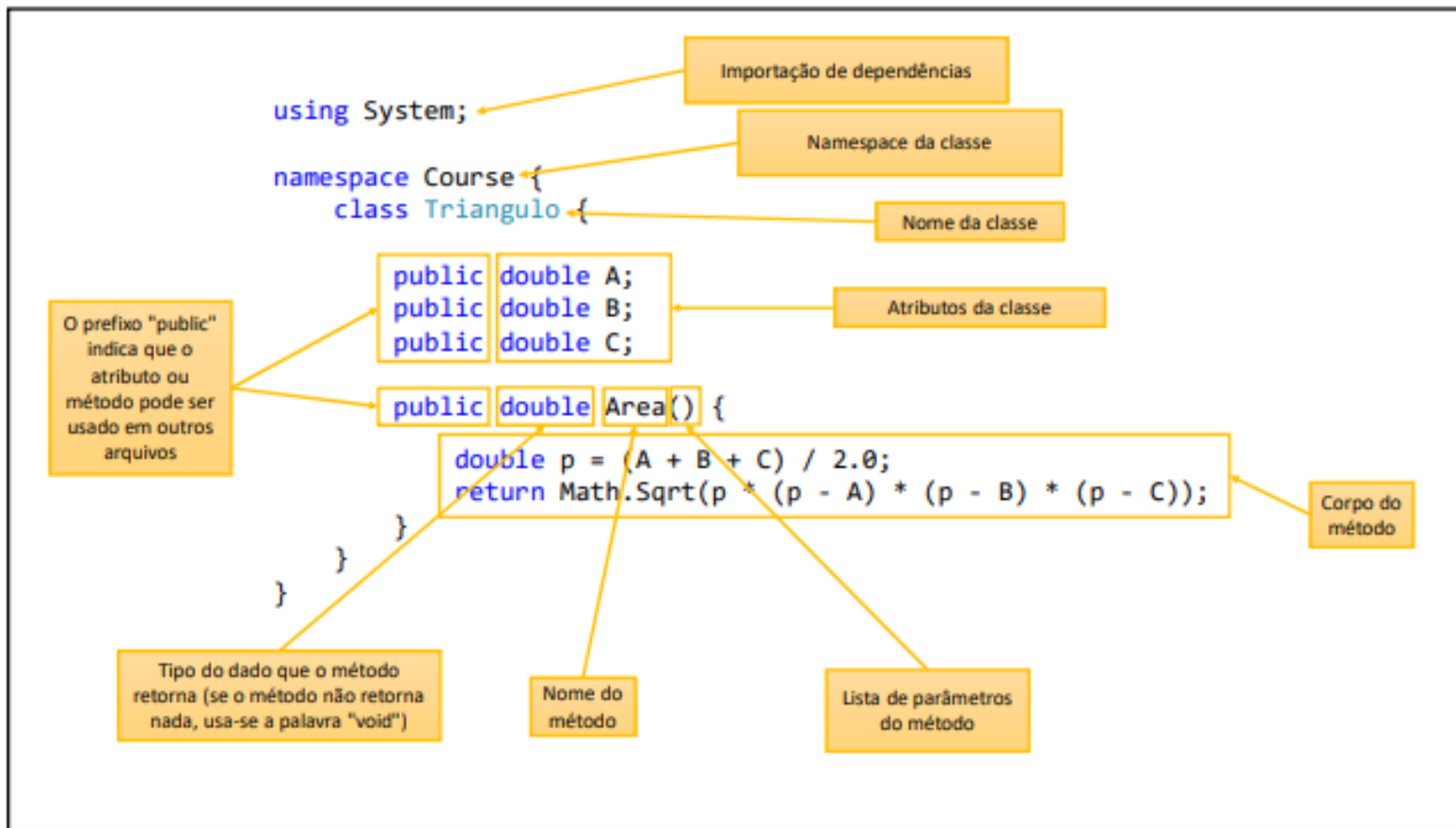
- Classe: é a definição do tipo
- Objetos: são instâncias da classe

```
namespace Course {  
    class Triangulo {  
  
        public double A;  
        public double B;  
        public double C;  
  
    }  
}
```



DS2

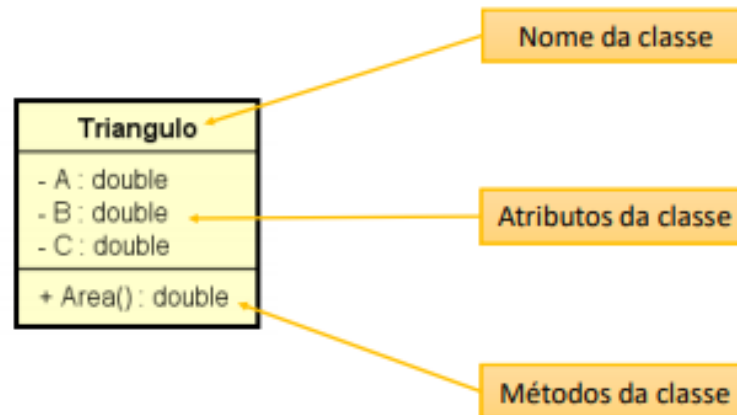
Orientação a Objetos



DS2

Orientação a Objetos

Projeto da classe (UML)



Orientação a Objetos

Quais são os benefícios de se calcular a área de um triângulo por meio de um MÉTODO dentro da CLASSE Triângulo?

- 1) Reaproveitamento de código:** nós eliminamos o código repetido (cálculo das áreas dos triângulos x e y) no programa principal.
- 2) Delegação de responsabilidades:** quem deve ser responsável por saber como calcular a área de um triângulo é o próprio triângulo. A lógica do cálculo da área não deve estar em outro lugar.

DS2

Orientação a Objetos

```
using System;

namespace Course {
    class Triangulo {

        public double A;
        public double B;
        public double C;

        public double Area() {
            double p = (A + B + C) / 2.0;
            return Math.Sqrt(p * (p - A) * (p - B) * (p - C));
        }
    }
}
```


DS2

Orientação a Objetos

```
using System;
using System.Globalization;

namespace Course {
    class Program {
        static void Main(string[] args) {

            Triangulo x, y;

            x = new Triangulo();
            y = new Triangulo();

            Console.WriteLine("Entre com as medidas do triângulo X:");
            x.A = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            x.B = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            x.C = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            Console.WriteLine("Entre com as medidas do triângulo Y:");
            y.A = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            y.B = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);
            y.C = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

            double areaX = x.Area();

            double areaY = y.Area();

            Console.WriteLine("Área de X = " + areaX.ToString("F4", CultureInfo.InvariantCulture));
            Console.WriteLine("Área de Y = " + areaY.ToString("F4", CultureInfo.InvariantCulture));

            if (areaX > areaY) {
                Console.WriteLine("Maior área: X");
            }
            else {
                Console.WriteLine("Maior área: Y");
            }
        }
    }
}
```

DS2

Orientação a Objetos

Fazer um programa para ler os dados de um produto em estoque (nome, preço e quantidade no estoque). Em seguida:

- Mostrar os dados do produto (nome, preço, quantidade no estoque, valor total no estoque)
- Realizar uma entrada no estoque e mostrar novamente os dados do produto
- Realizar uma saída no estoque e mostrar novamente os dados do produto

Produto
- Nome : string - Preço : double - Quantidade : int
+ ValorTotalEmEstoque() : double + AdicionarProdutos(quantidade : int) : void + RemoverProdutos(quantidade : int) : void

DS2

Orientação a Objetos

Entre os dados do produto:

Nome: **TV**

Preço: **900.00**

Quantidade no estoque: **10**

Dados do produto: TV, \$ 900.00, 10 unidades, Total: \$ 9000.00

Digite o número de produtos a ser adicionado ao estoque: **5**

Dados atualizados: TV, \$ 900.00, 15 unidades, Total: \$ 13500.00

Digite o número de produtos a ser removido do estoque: **3**

Dados atualizados: TV, \$ 900.00, 12 unidades, Total: \$ 10800.00

Produto
- Nome : string - Preço : double - Quantidade : int
+ ValorTotalEmEstoque() : double + AdicionarProdutos(quantidade : int) : void + RemoverProdutos(quantidade : int) : void

DS2

Exercício

EXERCÍCIO 01:

Fazer um programa para ler os dados de duas pessoas, depois mostrar o nome da pessoa mais velha.

Exemplo:

```
Dados da primeira pessoa:  
Nome: Maria  
Idade: 17  
Dados da segunda pessoa:  
Nome: Joao  
Idade: 16  
Pessoa mais velha: Maria
```

EXERCÍCIO 02:

Fazer um programa para ler nome e salário de dois funcionários. Depois, mostrar o salário médio dos funcionários.

Exemplo:

```
Dados do primeiro funcionário:  
Nome: Carlos Silva  
Salário: 6300.00  
Dados do segundo funcionário:  
Nome: Ana Marques  
Salário: 6700.00  
Salário médio = 6500.00
```