

Lógica e Pensamento Computacional II

com C#

Prof. Waldeck Lindoso Jr.

Operadores aritméticos

Operador	Significado
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão

NOTAS:

1) * / % tem precedência maior que + -

2) Exemplos:

$3 + 4 * 2 \longrightarrow$ Resultado: 11

$(3 + 4) * 2 \longrightarrow$ Resultado: 14

3) Pode-se usar parêntesis à vontade

4) Exemplo com mod:

$17 \% 3 \longrightarrow$ Resultado: 2

$$\begin{array}{r} 17 \text{ } | \text{ } 3 \\ 2 \text{ } 5 \end{array}$$

praticando...

```
int n1 = 3 + 4 * 2;  
int n2 = (3 + 4) * 2;  
int n3 = 17 % 3;  
double n4 = 10.0 / 8.0;  
  
double a = 1.0, b = -3.0, c = -4.0;  
  
double delta = Math.Pow(b, 2.0) - 4.0 * a * c;  
  
double x1 = (-b + Math.Sqrt(delta)) / (2.0 * a);  
double x2 = (-b - Math.Sqrt(delta)) / (2.0 * a);  
  
Console.WriteLine(n1);  
Console.WriteLine(n2);  
Console.WriteLine(n3);  
Console.WriteLine(n4);  
Console.WriteLine(delta);  
Console.WriteLine(x1);  
Console.WriteLine(x2);
```


$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Operadores de atribuição

Operador	Exemplo	Significado
=	a = 10;	a RECEBE 10
+=	a += 2;	a RECEBE a + 2;
-=	a -= 2;	a RECEBE a - 2;
*=	a *= 2;	a RECEBE a * 2;
/=	a /= 2;	a RECEBE a / 2;
%=	a %= 3;	a RECEBE a % 3;

praticando...

```
int a = 10;  
Console.WriteLine(a);  
  
a += 2;  
Console.WriteLine(a);  
  
a *= 3;  
Console.WriteLine(a);  
  
string s = "ABC";  
Console.WriteLine(s);  
  
s += "DEF";  
Console.WriteLine(s);
```

 Microsoft Visual Studio Debu

```
10  
12  
36  
ABC  
ABCDEF
```

Operadores aritméticos / atribuição

Operador	Exemplo	Significado
++	a++; ou ++a;	a = a + 1;
--	a--; ou --a;	a = a - 1;

```
int a = 10;  
a++;  
Console.WriteLine(a);
```

SAÍDA:
11

```
int a = 10;  
int b = a++;  
Console.WriteLine(a);  
Console.WriteLine(b);
```

SAÍDA:
11
10

```
int a = 10;  
int b = ++a;  
Console.WriteLine(a);  
Console.WriteLine(b);
```

SAÍDA:
11
11

Entrada de dados (**console**)

Comando

- `Console.ReadLine();`
 - Lê da entrada padrão **até a quebra de linha**.
 - Retorna os dados lidos **em forma de string**.

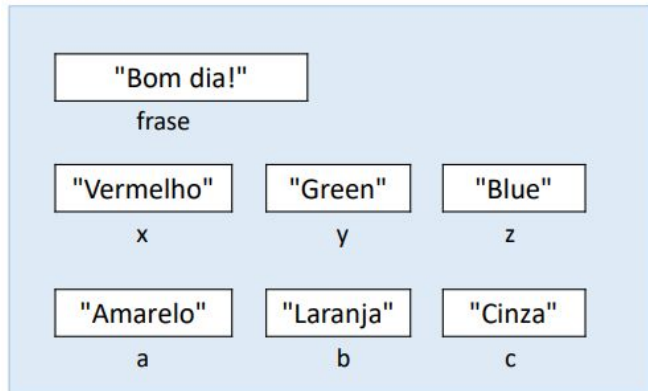
Entrada de dados (**console**)

- Ler um texto até a quebra de linha e armazenar em uma variável
- Ler três palavras, uma em cada linha, armazenando cada uma em uma variável
- Ler três palavras na mesma linha, separadas por espaço, armazenando cada uma em uma variável

Console:

```
Bom dia!  
Vermelho  
Verde  
Azul  
Amarelo Laranja Cinza
```

Memória:



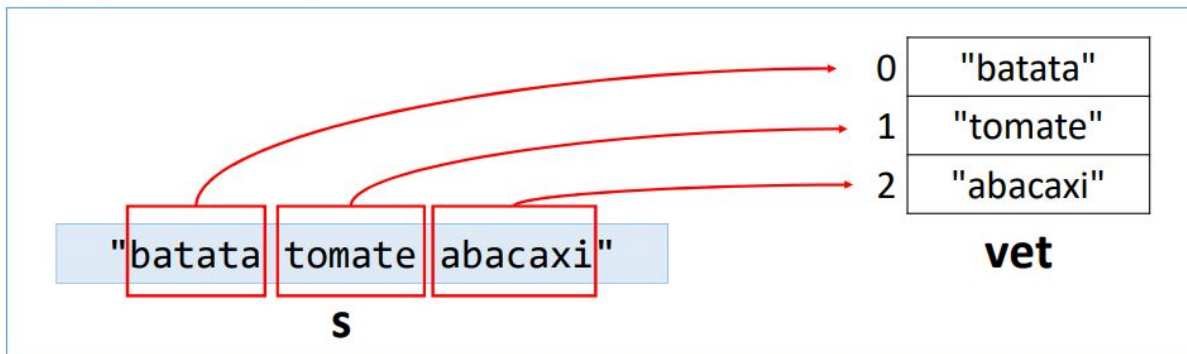
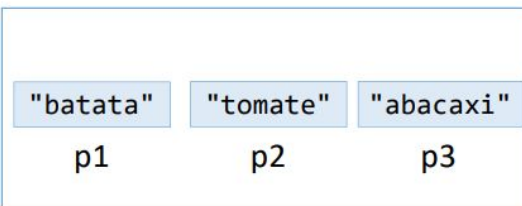
Entrada de dados (**console**)

Split

```
string s = Console.ReadLine();  
  
string[] vet = s.Split(' ');  
string p1 = vet[0];  
string p2 = vet[1];  
string p3 = vet[2];
```

Console:

```
batata tomate abacaxi
```



praticando...

```
string frase = Console.ReadLine();
string x = Console.ReadLine();
string y = Console.ReadLine();
string z = Console.ReadLine();

string[] v = Console.ReadLine().Split(' ');
string a = v[0];
string b = v[1];
string c = v[2];

Console.WriteLine("Você digitou: ");
Console.WriteLine(frase);
Console.WriteLine(x);
Console.WriteLine(y);
Console.WriteLine(z);
Console.WriteLine(a);
Console.WriteLine(b);
Console.WriteLine(c);
```

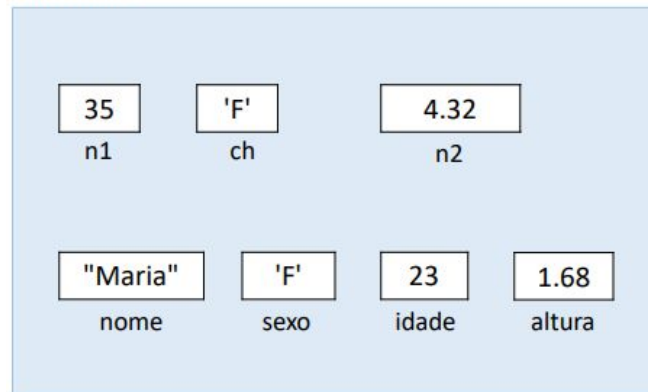
Entrada de dados (**console**)

- Ler um número inteiro
- Ler um caractere
- Ler um número double
- Ler um nome (única palavra), sexo (caractere F ou M), idade (inteiro) e altura (double) na mesma linha, armazenando-os em quatro variáveis com os devidos tipos

Console:

```
35
F
4.32
Maria F 23 1.68
```

Memória:



praticando...

```
int n1 = int.Parse(Console.ReadLine());
char ch = char.Parse(Console.ReadLine());
double n2 = double.Parse(Console.ReadLine(), CultureInfo.InvariantCulture);

string[] vet = Console.ReadLine().Split(' ');
string nome = vet[0];
char sexo = char.Parse(vet[1]);
int idade = int.Parse(vet[2]);
double altura = double.Parse(vet[3], CultureInfo.InvariantCulture);

Console.WriteLine("Você digitou:");
Console.WriteLine(n1);
Console.WriteLine(ch);
Console.WriteLine(n2.ToString("F2", CultureInfo.InvariantCulture));
Console.WriteLine(nome);
Console.WriteLine(sexo);
Console.WriteLine(idade);
Console.WriteLine(altura.ToString("F2", CultureInfo.InvariantCulture));
```

Exercício de fixação

Fazer um programa para executar a seguinte interação com o usuário, lendo os valores destacados em vermelho, e depois mostrar os dados na tela:

Entre com seu nome completo:

Alex Green

Quantos quartos tem na sua casa?

3

Entre com o preço de um produto:

500.50

Entre seu último nome, idade e altura (mesma linha):

Green 21 1.73

**SAÍDA ESPERADA (NÚMEROS REAIS
COM DUAS CASAS DECIMAIS):**

Alex Green

3

500.50

Green

21

1.73

Operadores comparativos

Operador	Significado
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
==	igual
!=	diferente

```
int a = 10;  
bool c1 = a < 10;  
bool c2 = a < 20;  
bool c3 = a > 10;  
bool c4 = a > 5;  
Console.WriteLine(c1);  
Console.WriteLine(c2);  
Console.WriteLine(c3);  
Console.WriteLine(c4);  
Console.WriteLine("-----");
```

```
bool c5 = a <= 10;  
bool c6 = a >= 10;  
bool c7 = a == 10;  
bool c8 = a != 10;  
Console.WriteLine(c5);  
Console.WriteLine(c6);  
Console.WriteLine(c7);  
Console.WriteLine(c8);
```

Operadores lógicos

Operador	Significado
&&	E
 	OU
!	NÃO

C1	C2	C1 E C2
F	F	F
F	V	F
V	F	F
V	V	V

C1	C2	C1 OU C2
F	F	F
F	V	V
V	F	V
V	V	V

NOTAS:

1) Precedência: ! > && > ||

2) Pode-se usar parêntesis à vontade

3) Exemplos:

2 > 3 || 4 != 5 → Resultado: true

!(2>3) && 4 != 5 → Resultado: true

praticando...

```
bool c1 = 2 > 3 || 4 != 5; // true
bool c2 = !(2 > 3) && 4 != 5; // true
Console.WriteLine(c1);
Console.WriteLine(c2);

Console.WriteLine("-----");

bool c3 = 10 < 5; // false
bool c4 = c1 || c2 && c3; // true
Console.WriteLine(c3);
Console.WriteLine(c4);
```


Perguntas ??

