

# EXPRESSÕES ARITMÉTICAS – PARTE 1

---

## 5.1 Operadores aritméticos

Os operadores aritméticos definem as operações aritméticas que podem ser realizadas sobre os números inteiros e reais. Para os inteiros, as operações aritméticas são a adição, subtração, multiplicação e resto. Para os números reais, as operações aritméticas são a adição, subtração, multiplicação e divisão. Nesta aula, restringiremos nossa atenção aos números inteiros apenas. Na linguagem Portugol, os operadores aritméticos correspondentes às operações definidas sobre os inteiros são

- + (adição)
- – (subtração ou menos unário)
- \* (multiplicação)
- \ (divisão *inteira*)
- % (resto – aplicado apenas aos valores inteiros).

Com os operadores acima, podemos escrever *expressões aritméticas* que envolvem constantes e variáveis inteiras. Por exemplo, suponha que  $a$ ,  $b$  e  $c$  sejam variáveis do tipo inteiro. Então, temos que

$$a + b + c, \quad a - b * c \% 2, \quad e \quad - 5 + 3 * 8 \backslash 2$$

são todas expressões aritméticas *válidas* na linguagem Portugol.

Nos exemplos acima, tivemos o cuidado de usar operandos do *mesmo tipo*. A razão para tal é que, por definição, cada operador aritmético atua sobre valores de um mesmo tipo e o resultado da operação deve sempre ser um valor do mesmo tipo dos operandos. Logo, não faz sentido escrevermos algo como  $a + 2$  quando  $a$  é uma variável ou constante do tipo real, pois existe uma ambigüidade em relação ao resultado da operação. No entanto, como veremos mais adiante, podemos definir *regras semânticas* associadas aos operadores que nos permitem interpretar, de forma única, o resultado da operação aritmética correspondente. Tais regras nos permitirão escrever expressões aritméticas envolvendo variáveis e constantes dos tipos inteiro e real.

## 5.2 Precedência de operadores

Qual é o valor da expressão aritmética

$$5 * 3 \% 2 \quad ?$$

Podemos dizer que o valor da expressão é igual a 1, se avaliarmos  $5 * 3$  primeiro e, depois,  $15 \% 2$ , ou podemos dizer que é igual a 5 se avaliarmos  $3 \% 2$  primeiro e, depois,  $5 * 1$ . As duas respostas são igualmente válidas. No entanto, como não podemos permitir ambigüidades em algoritmos, devemos definir **regras de precedência** de operadores, que são regras para definir a ordem em que os operadores aritméticos que vimos são aplicados em uma expressão aritmética qualquer.

Na linguagem Portugol, os operadores possuem **prioridades** associadas com eles. A operação associada a um operador com prioridade  $p$  é sempre executada *antes* da operação associada a um operador com prioridade  $q$  sempre que  $p > q$ . Quando  $p = q$ , a operação correspondente ao operador mais à esquerda é executado. O operador de maior prioridade é o **menos unário**,  $-$ . Em seguida, temos os operadores  $*$ ,  $\backslash$  e  $\%$ . Finalmente, com a prioridade mais baixa, temos os operadores  $+$  e  $-$ , onde  $+$  e  $-$  são os operadores de adição e subtração, respectivamente. A Tabela 5.1 resume essas prioridades.

Operador	Símbolo	Prioridade
menos unário	$-$	mais alta
multiplicação, divisão inteira e resto	$*$ , $\backslash$ e $\%$	$\uparrow$
adição e subtração	$+$ , $-$	mais baixa

Tabela 5.1: Operadores aritméticos sobre os inteiros e suas prioridades.

Por exemplo, em

$$a + b + c,$$

a operação  $a + b$  é realizada e, em seguida, o resultado dela é adicionado ao valor de  $c$ , pois os operadores possuem a mesma prioridade e, portanto, as operações são realizadas da esquerda para a direita.

Na expressão aritmética

$$a - b * c \% 2,$$

a operação  $b * c$  é efetuada primeiro e, em seguida, o resto da divisão de  $b * c$  por 2 é calculado. Finalmente, o resto é subtraído de  $a$ . Note que a multiplicação foi efetuada antes da divisão, pois os operadores  $*$  e  $\%$  possuem a mesma prioridade, mas  $*$  está mais à esquerda.

Uma boa forma de se familiarizar com os operadores aritméticos e as regras de precedência é escrevendo algoritmos para escrever o resultado de expressões aritméticas. O Algoritmo 5.1 calcula e escreve, usando a instrução `escreval`, o resultado de expressões envolvendo números inteiros. A instrução `escreval` faz o mesmo que a instrução `escreva`, mas gera um “salto de linha” após a escrita. Um algoritmo mais interessante, o Algoritmo 5.2, recebe, como entrada, três inteiros quaisquer e calcula e escreve o resultado de algumas expressões aritméticas envolvendo os inteiros lidos.

### 5.3 Alteração de prioridades

Algumas vezes é desejável alterar a ordem (imposta pelas regras de precedência) segundo a qual as operações são realizadas em uma expressão aritmética. Para tal, fazemos uso de *parênteses*. Por hipótese, todo operador possui prioridade mais baixa do que a dos parênteses. Isto garante que os operandos correspondentes ao valor das expressões entre parênteses sejam calculados *antes* de serem usados pelos demais operadores. É importante destacar que os parênteses devem ocorrer em *pares* (um aberto e um fechado) nas expressões e podem ocorrer “aninhados” em diversos níveis.

Algoritmo 5.1: Algoritmo para calcular algumas expressões aritméticas envolvendo números inteiros.

```

1  algoritmo "Calculo de expressoes aritmeticas"
2
3  inicio
4      escreval( "O resultado da expressao 5 * 3 % 2 e: ", 5 * 3 % 2 )
5
6      escreval( "O resultado da expressao -5 * 3 % 2 \ 8 e: ", -5 * 3 % 2 \ 8 )
7
8      escreval( "O resultado da expressao -5 - 3 - 6 * 3 e: ", -5 - 3 - 6 * 3 )
9
10 fimalgoritmo

```

Por exemplo, na expressão

$$(a - b) * (c \% 2),$$

a operação  $a - b$  é realizada primeiro. Em seguida, a operação  $c \% 2$  é realizada e, por último, a multiplicação dos resultados das duas operações anteriores entre os parênteses é realizada. Mas, como sabemos disso? A idéia é imaginar as expressões entre parênteses como operandos a serem “descobertos”. Com isso em mente, a expressão acima pode ser imaginada como tendo a forma

$$op_1 * op_2,$$

onde  $op_1$  e  $op_2$  são as expressões  $(a - b)$  e  $(c \% 2)$ . Então, o que temos é uma simples multiplicação de dois valores,  $op_1$  e  $op_2$ . No entanto, para que esta multiplicação seja realizada, precisamos dos valores  $op_1$  e  $op_2$ . Para tal, **assumimos** que o valor,  $op_1$ , à esquerda do operador de multiplicação,  $*$ , será obtido antes do valor,  $op_2$ , à direita dele. Para calcular  $op_1$ , avaliamos a expressão

$$a - b,$$

que se encontra dentro do parênteses. Neste momento, descobrimos que a subtração  $a - b$  é a primeira operação aritmética *realizada*. Uma vez que o valor  $op_1$  tenha sido descoberto, avaliamos

$$c \% 2,$$

que é a expressão correspondente ao valor  $op_2$ . Neste momento, descobrimos que a operação de resto de divisão,  $c \% 2$ , é a segunda operação aritmética *realizada*. Neste momento, dispomos dos valores  $op_1$  e  $op_2$  e, portanto, podemos realizar a multiplicação  $op_1 * op_2$ , que passa a ser

a terceira operação realizada. Logo, os operadores são aplicados na ordem  $-$ ,  $\%$  e  $*$ . Usamos a notação

$$(a -_1 b) *_3 (c \%_2 2)$$

para indicar este fato. Isto é, os operadores possuem índices que indicam a ordem em que são aplicados.

Algoritmo 5.2: Algoritmo para calcular algumas expressões aritméticas com variáveis inteiras.

```

1  algoritmo "Expressoes aritmeticas envolvendo variaveis e constantes"
2  var
3      a , b , c : inteiro
4  inicio
5      escreva( "Entre com o valor da variavel a: " )
6      leia( a )
7
8      escreva( "Entre com o valor da variavel b: " )
9      leia( b )
10
11     escreva( "Entre com o valor da variavel c: " )
12     leia( c )
13
14     escreval( "O resultado da expressao a * b % c e: ", a * b % c )
15
16     escreval( "O resultado da expressao -a * b % c * 8 e: ", -a * b % c * 8 )
17
18     escreval( "O resultado da expressao -a - b - c * 3 e: ", -a - b - c * 3 )
19
20  fimalgoritmo

```

A expressão

$$((2 + 3) - (1 + 2)) * 3 - (3 + (3 - 2))$$

é bem mais complexa do que a anterior, mas podemos determinar a ordem em que os operadores são aplicados da mesma forma que antes. O primeiro passo é substituir as expressões dentro dos parênteses por operandos a serem descobertos. Isto é feito para os parênteses mais externos:

$$op_1 * 3 - op_2.$$

Agora, vemos que se os valores entre parênteses fossem conhecidos, haveria apenas duas operações a serem realizadas: uma multiplicação e uma adição. A multiplicação possui prioridade sobre a adição e ela precisa do valor  $op_1$  para ser realizada. Então, considere a expressão correspondente a  $op_1$ :

$$(2 + 3) - (1 + 2).$$

Esta expressão contém outras expressões dentro de parênteses e, assim como antes, ela pode ser vista como

$$op_3 - op_4.$$

onde  $op_3$  e  $op_4$  correspondem às expressões  $2 + 3$  e  $1 + 2$ , respectivamente. Para realizarmos a operação de subtração acima, precisamos dos valores  $op_3$  e  $op_4$ . Por estar à esquerda do

operador, o valor  $op_3$  é descoberto primeiro. Isto implica que a primeira operação realizada é a adição

$$2 + 3$$

e a próxima é a adição

$$1 + 2.$$

Em seguida, temos a subtração  $op_3 - op_4$ :

$$(2 + 3) - (1 + 2).$$

Depois que a subtração acima for realizada, o valor  $op_1$  se torna conhecido e, conseqüentemente, a multiplicação  $op_1 * 3$  pode ser realizada, tornando-se a quarta operação realizada. O resultado desta operação é o primeiro operando disponível da subtração em  $op_1 * 3 - op_2$ . Mas, esta subtração não pode ser efetuada antes do valor  $op_2$  ser conhecido, ou seja, antes da expressão

$$3 + (3 - 2)$$

ser avaliada. Assim como fizemos antes, podemos imaginar a expressão acima tendo a forma

$$3 + op_5,$$

onde  $op_5$  é o valor da expressão,  $3 - 2$ , entre parênteses. A adição na expressão acima precisa do valor  $op_5$  para ser realizada. Isto significa que a subtração  $3 - 2$  é a quinta operação realizada. Depois dela, a adição  $3 + op_5$  é realizada, tornando-se a sexta operação realizada. Logo em seguida, o valor  $op_2$  se torna conhecido, o que possibilita a realização da sétima e última operação, que é a subtração em  $op_1 * 3 - op_2$ . Usando a notação de subscrito, temos a seguinte ordem:

$$((2 +_1 3) -_3 (1 +_2 2)) *_4 3 -_7 (3 +_6 (3 -_5 2)).$$

## 5.4 A instrução de atribuição

Quando utilizamos expressões aritméticas em nossos algoritmos, necessitaremos, muito freqüentemente, armazenar o valor da expressão em uma variável. Como vimos na Aula ??, a atribuição de um valor a uma variável pode ser realizada através da instrução de leitura `leia` ou do operador de atribuição `<-`. A instrução `leia` é usada *apenas* quando o valor a ser atribuído à variável é fornecido como entrada para o algoritmo. Como o valor que queremos atribuir à variável é resultante da avaliação de uma expressão aritmética, devemos usar o operador de atribuição.

Por exemplo, suponha que o resultado da expressão

$$5 * \%2$$

deva ser atribuído a uma variável inteira de nome *resultado*. Então, a atribuição pode ser realizada da seguinte forma:

$$\text{resultado} <- 5 * \%2$$

Para um exemplo mais concreto do uso do operador de atribuição, considere o Algoritmo 5.3, que lê dois números inteiros, calcula o quadrado da soma dos dois números lidos, atribui o resultado a uma variável inteira, usando o operador `<-`, e escreve o valor da variável.

## 5.5 Exercícios resolvidos

1. Considere a expressão polinomial

$$5x^3 + 7x^2 - 3x - 1,$$

onde  $x$  é uma variável. Escreva a expressão acima usando a linguagem Portugal.

solução:

$$5 * x * x * x + 7 * x * x - 3 * x - 1.$$

2. Avalie a seguinte expressão aritmética de acordo com as regras de precedência da linguagem Portugal:

$$2 - 3 * 5$$

solução:

$$2 - 3 * 5 = 2 - 15 = -13.$$

Algoritmo 5.3: Algoritmo para calcular o quadrado da soma de dois inteiros.

```

1  algoritmo "Quadrado da soma de 2 inteiros"
2  var
3    a , b , quadrado : inteiro
4  inicio
5    escreva( "Entre com o primeiro inteiro: " )
6    leia( a )
7
8    escreva( "Entre com o segundo inteiro: " )
9    leia( b )
10
11   quadrado <- ( a + b ) * ( a + b )
12
13   escreva ( "O quadrado da soma dos inteiros lidos e: " , quadrado )
14  fimalgoritmo

```

## 5.6 Exercícios propostos

1. Escreva a expressão abaixo usando a linguagem Portugal:

$$x_0 + v \cdot t$$

2. Avalie a seguinte expressão aritmética de acordo com as regras de precedência da linguagem Portugal:

$$(2 - 3) * 5$$

3. Suponha que a linha 11 do Algoritmo 5.3 seja substituída por

```
quadrado <- a + b * a + b
```

Você acha que o algoritmo continuará correto? Justifique sua resposta.

4. Suponha que a linha 11 do Algoritmo 5.3 seja substituída pelas duas seguintes linhas:

```
quadrado <- a + b
```

```
quadrado <- quadrado * quadrado
```

Você acha que o algoritmo continuará correto? Justifique sua resposta.

5. Escreva um algoritmo, usando a linguagem Portugol, para ler dois números inteiros, calcular o cubo da soma desses dois números e escrever o resultado calculado como saída.
6. Implemente o algoritmo anterior usando a ferramenta VISUALG.