
Algoritmos e Lógica de Programação

— Estruturas Sequenciais —

Prof. Rhafael Freitas da Costa

Conteúdo

- Retomada dos principais pontos da aula anterior;
- Entrada e saída de dados;
- Operadores e expressões matemáticas;
- Precedência de operadores;
- Ao final do componente curricular será possível aplicar os conceitos de entrada, processamento e saída de dados na resolução de problemas algorítmicos em português estruturado.

Pseudocódigo

//O comando programa é obrigatório

programa {

//Dentro do programa é permitido declarar variáveis globais, constantes globais e funções em qualquer ordem

funcao inicio() {

//Declarações de variáveis locais, constantes locais, estruturas de controle e expressões

}

}

Na aula anterior

Componentes de linguagens de programação:

- **Literais:** números, valores lógicos e caracteres
- **Identificadores:** nomes de variáveis, constantes, algoritmos, funções, ...
- **Palavras reservadas:** se, entao, fim, repita, inicio, fim, ...
- **Símbolos especiais:** = (atribuição), + (soma), () (parênteses), > (maior), == (igual), ; (ponto e vírgula)
- **Comentários:** em pseudocódigo são delimitados por `/*` e `*/` ou pelos símbolos `/*` no início da linha

Na aula anterior

Tipos de dados:

- **inteiro** – somente valores numéricos inteiros
 - ex.: 3, -6, 23, ...
- **real** – somente valores numéricos fracionários
 - ex.: 76.9, -47.5, 93.2, ...
- **logico** – somente dois valores lógicos
 - ex.: verdadeiro ou falso
- **caracter** – um único caractere alfanumérico ASCII
 - ex.: 'a', '+', 'j', ...
- **cadeia** – sequência de caracteres ou string
 - ex.: "Algoritmos", "+55 41 3068 7715", "02/10/2023", "zezinho do carro do ovo", ...

Na aula anterior

- Uma **constante** representa um espaço de memória identificado e reservado para guardar um valor **fixo e imutável** durante a execução do programa;
- Uma **variável** representa um espaço de memória identificado e reservado para guardar um valor **mutável** durante a execução do programa.

Na aula anterior

- Declaração de constantes

const <tipo da constante> <NOME DA CONSTANTE> = <valor da constante>

- Declaração de variáveis

<tipo da variavel> <nome da variavel>

- Atribuição de conteúdo para variáveis

<nome da variavel> = <valor do mesmo tipo da variável>

Comando de saída

- Serve para imprimir algum tipo de informação na tela do computador;
- A lista de valores de saída pode conter:
 - cadeias de caracteres cercadas por aspas;
 - nomes de variáveis cujos conteúdos devem ser previamente atribuídos;
 - expressões lógicas, aritméticas ou relacionais cujo resultado é impresso na saída.
- Em pseudocódigo se utiliza o comando:

escreva (<conteudo_de_saida>)

Comando de saída

- Para imprimir o conteúdo de uma variável, basta colocar o seu identificador diretamente dentro do comando.

```
cadeia nome = "Mariane"  
escreva(nome)
```

- Como nome não tem aspas, o algoritmo considera nome como identificador de uma variável e o substitui pelo seu conteúdo, no caso Mariane

```
Mariane  
Programa finalizado. Tempo de execução: 28 ms
```

Comando de saída

- E se eu imprimir o nome da variável com aspas?

```
cadeia nome = "Mariane"
```

```
escreva("nome")
```

- Como "nome" tem aspas, o algoritmo o considera como uma simples sequência de caracteres e o imprime

```
nome  
Programa finalizado. Tempo de execução: 20 ms
```

Exemplo da quantidade de gasolina

```
programa {  
    funcao inicio() {  
        real preco_gasolina, valor_disponivel,  
            qtde_litros  
  
        preco_gasolina = 5.69  
        valor_disponivel = 100.0
```

→ *(segue no quadro ao lado)*

```
// realiza o cálculo e atribui o resultado  
na variável qtde_litros
```

```
qtde_litros = valor_disponivel /  
preco_gasolina
```

```
escreva("A quantidade de litros para este  
valor e: ")
```

```
escreva(qtde_litros)
```

```
}
```

```
}
```

A quantidade de litros para este valor e: 17.57469244288225
Programa finalizado. Tempo de execução: 28 ms

Exemplo calculo da área

```
programa {  
    funcao inicio() {  
        const real PI = 3.14159265 // constante  
        real area, raio // variáveis  
  
        raio = 20.0  
  
        area = PI * raio * raio  
  
        escreva("A area e:")  
        escreva(area)  
    }  
}
```

Comando de saída

- E se eu quiser imprimir uma sequência de caracteres + o conteúdo da variável?

```
cadeia nome = "Mariane"  
escreva("Nome digitado: ", nome)
```

- Em pseudocódigo, o sinal de **soma (+)** e a **vírgula (,)** tem a função de **concatenar** (unir) as duas informações.

```
Nome digitado: Mariane  
Programa finalizado. Tempo de execução: 31 ms
```

Comando de concatenação de saída

```
programa {  
    funcao inicio() {  
        cadeia nome  
        caracter letra  
  
        nome = "Maria"  
        letra = 'X'  
  
        escreva("Maria" + "Silva" + "\n")  
        escreva("Maria", "Silva", "\n")  
        escreva(nome + "Silva" + "\n")  
        escreva(nome + 'S' + "\n")  
        escreva(letra + nome + letra)  
    }  
}
```

O comando "**\n**" adiciona uma quebra de linha

```
MariaSilva  
MariaSilva  
MariaSilva  
MariaS  
XMariaX  
Programa finalizado. Tempo de execução: 34 ms
```

Comando de entrada

- Servem para **receber informações de fora do algoritmo**. Ex.: caixa de supermercado e biblioteca;
- É necessário **declarar variáveis de tipos compatíveis** com as informações que estão sendo solicitadas ao usuário;
- Em pseudocódigo, se utiliza o comando:

leia(<nome_da_variavel>)

Comando de entrada

```
programa {
```

```
    funcao inicio() {
```

```
        inteiro numero
```

```
        escreva("Digite um numero e tecle <ENTER>:")
```

```
        leia(numero)
```

```
        escreva("O numero digitado foi: ")
```

```
        escreva(numero)
```

```
    }
```

```
}
```

```
Digite um numero e tecle <ENTER>:23
```

```
O numero digitado foi: 23
```

```
Programa finalizado. Tempo de execução: 2735 ms
```


Comando de entrada

```
programa {  
  
    funcao inicio() {  
        inteiro numero // o programa solicita ao sistema operacional que reserve um espaço na memória para a  
        variável numero  
  
        escreva("Digite um numero e tecle <ENTER>:")  
        leia(numero) // o espaço na memória associado à variável numero passa a conter o valor digitado  
  
        escreva("O numero digitado foi: ")  
        escreva(numero) // o computador busca na memória o conteúdo da variável numero e imprime essa  
        informação  
    }  
}
```

Exemplo da biblioteca

```
programa {  
  
    funcao inicio() {  
        inteiro codigo_estudante  
        cadeia nome_livro  
  
        escreva("Digite o codigo do estudante e tecle <ENTER>: ")  
        leia(codigo_estudante)  
        escreva("Digite o nome do livro e tecle <ENTER>: ")  
        leia(nome_livro)  
  
        escreva("---Saida--- \n")  
        escreva("Estudante: ", codigo_estudante, "\n")  
        escreva("Nome do livro: ", nome_livro)  
    }  
}
```

```
Digite o codigo do estudante e tecle <ENTER>: 123  
Digite o nome do livro e tecle <ENTER>: Sherlock Holmes  
---Saida---  
Estudante: 123  
Nome do livro: Sherlock Holmes  
Programa finalizado. Tempo de execução: 7137 ms
```

Comando de saída

```
programa {  
  
    funcao inicio() {  
        cadeia nome, sobrenome  
  
        nome = "Homer"  
        sobrenome = "Simpson"  
  
        escreva("Ola, turma! \n")  
        escreva(nome, sobrenome, "\n")  
        escreva("Bart" + sobrenome + "\n")  
  
        nome = "Lisa"  
  
        escreva(nome + sobrenome + "\n")  
        escreva("Bart e " + nome + " e Meg " + sobrenome + " sao irmaos.")  
    }  
}
```

```
Ola, turma!  
HomerSimpson  
BartSimpson  
LisaSimpson  
Bart e Lisa e Meg Simpson sao irmaos.  
Programa finalizado. Tempo de execução: 37 ms
```

Comando de entrada

```
programa {
```

```
    funcao inicio() {
```

```
        cadeia nome, sobrenome
```

```
        inteiro ano
```

```
        real altura
```

```
        escreva("Digite o ano da serie: ")
```

```
        leia(ano)
```

```
        escreva("Digite o nome do personagem: ")
```

```
        leia(nome)
```

```
        escreva("Digite o sobrenome do personagem: ")
```

```
        leia(sobrenome)
```

→ *(segue no quadro ao lado)*

```
        escreva("Digite a altura do personagem: ")
```

```
        leia(altura)
```

```
        escreva("Ano da serie: ", ano, "\n")
```

```
        escreva("Personagem: ", nome, sobrenome, "\n")
```

```
        escreva("Altura: ", altura, "\n")
```

```
    }
```

```
}
```

```
Digite o ano da serie: 1972
```

```
Digite o nome do personagem: Sr
```

```
Digite o sobrenome do personagem: Madruga
```

```
Digite a altura do personagem: 1.72
```

```
Ano da serie: 1972
```

```
Personagem: SrMadruga
```

```
Altura: 1.72
```

```
Programa finalizado. Tempo de execução: 146795 ms
```

Exercício

Elabore um algoritmo que leia um número inteiro digitado pelo usuário e imprima seu sucessor e seu antecessor. Por exemplo, suponha que o usuário digite o número 7, o algoritmo deverá imprimir (escrever) na tela o seu antecessor (número 6) e o seu sucessor (o número 8). Apresente a solução em pseudocódigo e/ou fluxograma.

Expressões aritméticas, lógicas e relacionais

Expressões aritméticas

- São expressões cujos resultados são valores numéricos (inteiros ou reais);
- A sintaxe de uma expressão aritmética é a seguinte:

`<operando> <operador aritmético> <operando>`

Operadores aritméticos

| Tipo | Função | Exemplos |
|----------|------------------|--|
| + | Adição | $2 + 3$, $x + y$ |
| - | Subtração | $4 - 2$, $n - m$ |
| * | Multiplicação | $3 * 4$, $a * b$ |
| / | Divisão | $10 / 2$, $x1 / x2$ |
| ^ ou ** | Potenciação | 10^x , 2^3 |
| % ou mod | Resto da divisão | $11 \% 2$, $123 \% 10$ |
| \ ou div | Divisão inteira | $11 \setminus 2$, $123 \text{ div } 10$ |

Precedência geral de operadores

- As sub-expressões com maior prioridade são avaliadas primeiro e o seu valor substituído pela sub-expressão inteira;
- A seguir, a próxima sub-expressão na ordem é avaliada e assim por diante até que toda a expressão corresponda a um só valor.

| Ordem | Operador | Símbolo |
|-------|---|--------------------------|
| 1º | Parênteses | () |
| 2º | Potenciação | ^ ou ** |
| 3º | Multiplicação, divisão, resto e divisão inteira | *, /, % ou mod, \ ou div |
| 4º | Adição, subtração | +, - |

Precedência geral de operadores

- A maneira de alterar a ordem de execução das operações em uma expressão aritmética é através de **parênteses**;
- Parênteses são executados com prioridade, a partir dos mais internos para os mais externos;
- Exemplo:

$$(5+3)^2 * (5-2) + 8$$

$$8^2 * 3 + 8$$

$$64 * 3 + 8$$

$$192 + 8$$

$$200$$

Se ignorássemos a precedência de operadores, teríamos um resultado completamente diferente do esperado

Precedência geral de operadores

- A maneira de alterar a ordem de execução das operações em uma expressão aritmética é através de **parênteses**;
- Parênteses são executados com prioridade, a partir dos mais internos para os mais externos;
- Exemplo:

$$(5 + 3) ^ 2 * (5 - 2) + 8$$

$$8 ^ 2 * 3 + 8$$

$$64 * 3 + 8$$

$$192 + 8$$

$$200$$

$$5 + 3 ^ 2 * 5 - 2 + 8$$

$$5 + 9 * 5 - 2 + 8$$

$$5 + 45 - 2 + 8$$

$$50 - 2 + 8$$

$$56$$

Precedência geral de operadores

- Exemplos:

A. $7 - 2 * 3$

B. $3 + ((1 - 4) * 5)$

C. $7 + 1 - 5 * 3$

D. $7 + (1 - 5) * 3$

E. $1 - 4 * 3 / 6 - 3 ^ 2$

Precedência geral de operadores

- Exemplos:

A. $7 - 2 * 3 = 1$

B. $3 + ((1 - 4) * 5) = -12$

C. $7 + 1 - 5 * 3 = -7$

D. $7 + (1 - 5) * 3 = -5$

E. $1 - 4 * 3 / 6 - 3 ^ 2 = -10$

Escrita de expressões aritméticas geral de operadores

Independentemente do conhecimento da ordem de precedência adotada na linguagem, o **uso de parênteses é recomendado** não só porque garante a correta avaliação das expressões, mas também porque **facilita o entendimento** do que está sendo executado.

Exceções em expressões aritméticas

- Nem todas as expressões aritméticas possuem um valor definido matematicamente:
 - Divisão por 0 (zero)
 - Raiz quadrada com número negativo
- É interessante verificar os valores que farão parte das expressões aritméticas para evitar esse tipo de situação.

Exemplo do churrasco

- O objetivo é calcular a quantidade de carne a ser comprada para um churrasco
- Sabe-se que o cálculo a ser utilizado é o seguinte:

```
qtde_carne = ((consumo_medio_crianca * qtde_crianças) + (consumo_medio_adulto * qtde_adultos)) * 1.2
```


Exemplo do churrasco

```
programa {  
    funcao inicio() {
```

```
        const real MARGEM_SEGURANCA = 1.2
```

```
        real consumo_medio_crianca
```

```
        real qtde_carne
```

```
        inteiro qtde_crianças, c
```

```
        escreva("Digite o consu
```

```
        leia(consumo_medio_crianca
```

```
        escreva("Digite a quantidade de crianças: ")
```

```
        leia(qtde_crianças)
```

→ (segue no quadro ao lado)

```
        escreva("Digite o consumo medio para adultos: ")
```

```
        leia(consumo_medio_adulto)
```

```
        escreva("Digite a quantidade de adultos: ")
```

```
        Digite o consumo medio para crianças: 0.200
```

```
        Digite a quantidade de crianças: 3
```

```
        Digite o consumo medio para adultos: 0.500
```

```
        Digite a quantidade de adultos: 6
```

```
        A quantidade total de carne para o churrasco e: 4.32
```

```
        Programa finalizado. Tempo de execução: 12152 ms
```

```
        escreva("A quantidade total de carne para o  
        churrasco e: ", qtde_carne)
```

```
    }
```

```
}
```

Exemplo do churrasco

| | |
|--|---|
| <pre>programa { funcao inicio() { const real MARGEM_SEGURANCA = 1.2 real consumo_medio_crianca real qtde_carne inteiro qtde_crianças, qtde_adultos escreva("Digite o consumo medio para crianças: ") leia(consumo_medio_crianca) escreva("Digite a quantidade de crianças: ") leia(qtde_crianças) escreva("Digite o consumo medio para adultos: ") leia(qtde_adultos) escreva("Programa finalizado. Tempo de execução: 9756 ms") } }</pre> | <pre> escreva("Digite o consumo medio para adultos: ") leia(consumo_medio_adulto) escreva("Digite a quantidade de adultos: ") leia(qtde_adultos) consumo_medio_crianca * consumo_medio_adulto * MARGEM_SEGURANCA + (consumo_medio_adulto * qtde_adultos)) * MARGEM_SEGURANCA } }</pre> |
|--|---|

→ (segue no quadro ao lado)

Próxima aula

- Retomada dos principais pontos desta aula;
- Estruturas sequenciais;
- Exercícios.

Dúvidas



Obrigado

Prof. Rhafael Freitas da Costa