Algoritmos e Lógica de Programação

ALGORITMOS

Um algoritmo consiste em uma serie limitada de etapas, cujo objetivo é executar uma determinada tarefa.

É uma serie de instruções ordenadas de maneira lógica para solucionar determinado problema.

```
1  public class Tableless {
2
3      public static void main(String[] args) {
4
5          int valor1 = 1;
6          int valor2 = 2;
7
8          System.out.println((valor1 > valor2 ? "1 é maior" : "2 é maior"));
9
10         if (valor1 > valor2) {
11             System.out.println("1 é maior");
12         } else {
13             System.out.println("2 é maior");
14         }
15
16     }
17 }
```

O que é a lógica de programação?

A lógica de programação tem como objetivo mostrar métodos para solucionar problemas, além de determinar a sequência lógica para o desenvolvimento de um programa.

Método para a construção de algoritmos:

- Compreender completamente o problema a ser resolvido, destacando os pontos mais importantes e os objetos que o compõem.
- Definir os dados de entrada, ou seja, quais dados serão fornecidos e quais objetos fazem parte desse cenário problema.
- Definir o processamento, ou seja, quais cálculos serão efetuados e quais as restrições para esses cálculos. O processamento é responsável pela transformação dos dados de entrada em dados de saída.
- Definir os dados de saída, ou seja, quais dados serão gerados depois do processamento.
- Construir o algoritmo utilizando uma das formas de apresentação.
- Testar o algoritmo realizando simulações.

FORMAS DE REPRESENTAÇÃO DE ALGORITMOS

As formas de representação mais conhecidas para a representação de algoritmos são:

- Descrição narrativa;
- Fluxograma;
- Pseudocódigo.

DESCRIÇÃO NARRATIVA

A descrição narrativa consiste em analisar o enunciado do problema e escrever, utilizando uma linguagem natural (por exemplo, a língua portuguesa), os passos a serem seguidos para sua resolução.

Vantagem: não é necessário aprender nenhum conceito novo, pois uma língua natural, neste ponto, já é bem conhecida.

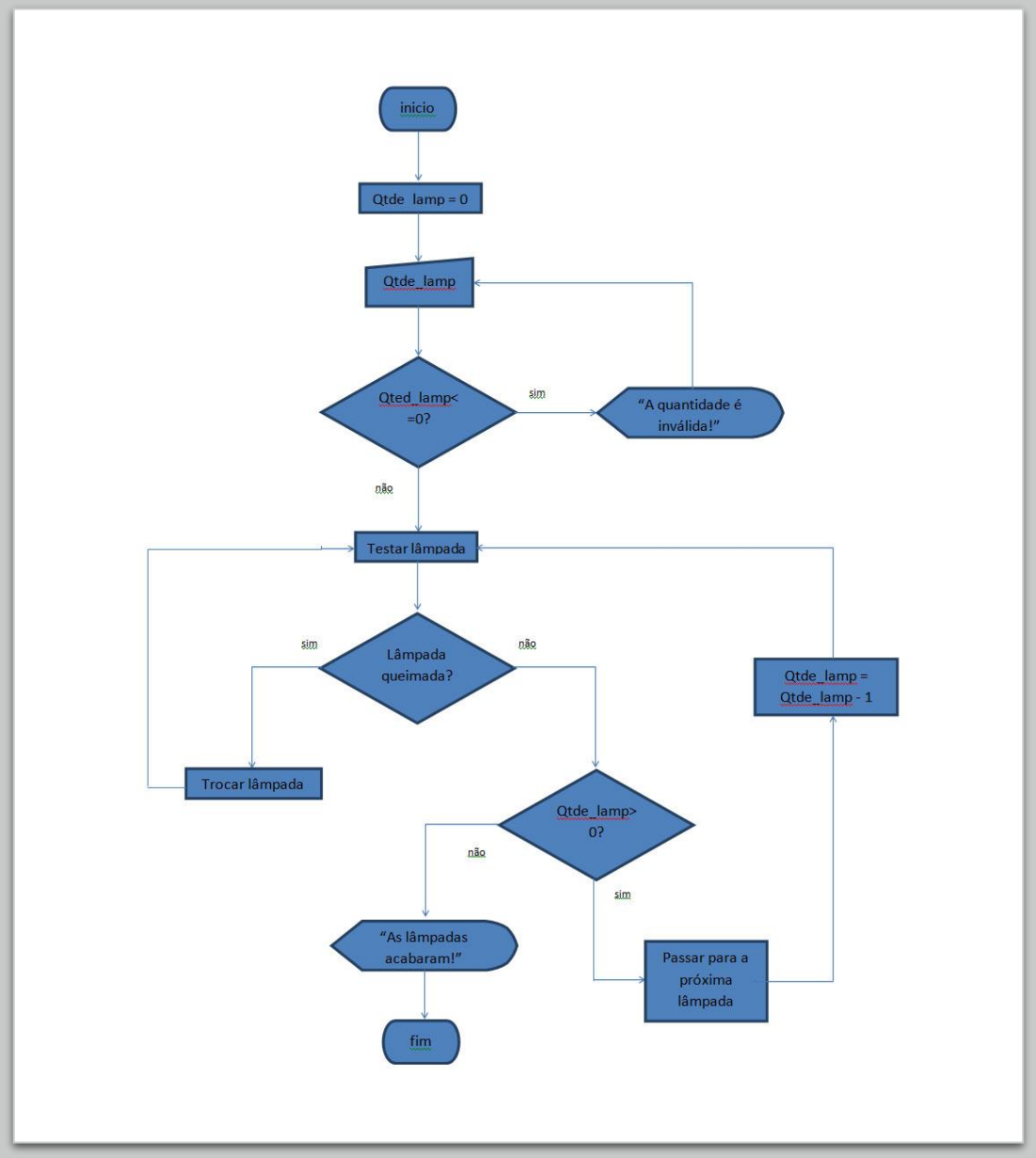
Desvantagem: a língua natural abre espaço para várias interpretações, o que posteriormente dificultará a transcrição desse algoritmo para programa.

Exemplo: **Somar três números**

Passo 1 – Receber os três números

Passo 2 – Somar os três números

Passo 3 – Mostrar o resultado obtido



FLUXOGRAMA

O fluxograma consiste em analisar o enunciado do problema e escrever, utilizando símbolos gráficos predefinidos, os passos a serem seguidos para sua resolução.

Vantagem: o entendimento de elementos gráficos é mais simples que o entendimento de textos.

Desvantagem: é necessário aprender a simbologia dos fluxogramas e, além disso, o algoritmo resultante não apresenta muitos detalhes, dificultando sua transcrição para um programa.

PSEUDOCÓDIGO

O pseudocódigo ou portugol consiste em analisar o enunciado do problema e escrever, por meio de regras predefinidas, os passos a serem seguidos para sua resolução.

Vantagem: a passagem do algoritmo para qualquer linguagem de programação é quase imediata, bastando conhecer as palavras reservadas da linguagem que será utilizada.

Desvantagem: é necessário aprender as regras do pseudocódigo.

```
1  algoritmo "exemplo1"
2  var
3     n1, n2, soma: real
4
5  inicio
6
7  escreva("digite um número ")
8  leia(n1)
9  escreva("digite outro número ")
10 leia(n2)
11 soma <- n1 + n2;
12 escreva(" primeiro número = ", n1)
13 escreva(" segundo número = ", n2)
14 escreva(" soma = ", soma)
15
16 fimalgoritmo
```

TIPOS DE DADOS

Tipo de dado primitivo e composto:

Um tipo primitivo (também conhecido por nativo ou básico) é fornecido por uma linguagem de programação como um bloco de construção básico.

Um tipo composto pode ser construído em uma linguagem de programação a partir de tipos primitivos e de outros tipos compostos, em um processo chamado composição.

Tipos de dados primitivos

Numéricos:

Os dados numéricos dividem-se em dois grupos: **inteiros e reais**.

Os **números inteiros** podem ser positivos ou negativos e não possuem parte fracionária

Ex: -23, 98, 237.

Os **números reais** podem ser positivos ou negativos e possuem parte fracionária.

Ex: 23.45, 346.89, -34.88.

Os números reais seguem a notação da língua inglesa, ou seja, a parte decimal é separada da parte inteira por um ponto, e não por uma vírgula.

Tipos de dados primitivos

Lógicos

São também chamados dados booleanos e podem assumir os valores **verdadeiro** ou **falso**.

Tipos de dados primitivos

Literais ou caracteres:

São dados formados por um único caractere ou por uma cadeia de caracteres. Esses caracteres podem ser as letras maiúsculas, as letras minúsculas, os números (não podem ser usados para cálculos) e os caracteres especiais (&, #, @, ?, +).

Ex: “aluno”, “1234”, “@ internet”, “0.34”, “1 + 2”, ‘A’, ‘3’.

Um caractere é representado entre apóstrofos e um conjunto de caracteres é representado entre aspas.

Liguagens com tipagem estática e dinâmica:

Linguagens com tipagem estática não permitem ao desenvolvedor alterar o tipo da variável depois de declarada. Geralmente a verificação de tipo é feita em tempo de compilação. Em C, C++ e Java, por exemplo, os tipos de dados são estáticos.

A tipagem dinâmica está ligado a habilidade da linguagem de programação em escolher o tipo de dado de acordo com o valor atribuído à variável em tempo de execução dinamicamente. Em Lisp, PHP e Python os dados são dinâmicos.

VARIÁVEIS, CONSTANTES

Uma variável é um espaço na memória do computador destinado a um dado que é alterado durante a execução do algoritmo. Para funcionar corretamente, as variáveis precisam ser definidas por nomes e tipos.

Constante quando o seu valor não se modifica durante a execução do algoritmo. Pode ser de qualquer tipo primitivo.

As variáveis, geralmente, são compostas por dois atributos distintos:

- Nome;
- o tipo de dados que armazenará.

Dentro de um mesmo programa não poderão ser criadas duas ou mais variáveis com o mesmo nome.

IDENTIFICADORES

Os identificadores são os nomes das variáveis, dos programas, das constantes, das rotinas, das unidades, etc.

As regras básicas para a formação dos identificadores são:

- Os caracteres permitidos são: os números, as letras maiúsculas, as letras minúsculas e o caractere sublinhado.
- O primeiro caractere deve ser sempre uma letra ou o caractere sublinhado.
- Não são permitidos espaços em branco e caracteres especiais (@, $, +, –, %, !).
- Não podemos usar as palavras reservadas nos identificadores, ou seja, palavras que pertençam à linguagem de programação.

EXPRESSÕES E OPERADORES

Uma expressão em um programa é a parte da instrução que produz um valor, normalmente através do uso de **operandos** (valores) e **operadores** (indicam a operação a ser realizada).

OPERADORES

Os operadores são meios pelos quais incrementamos, decrementamos, comparamos e avaliamos dados dentro do computador.

De acordo com o número de operandos sobre os quais os operadores atuam, os últimos podem ser classificados em:

- **Binários:** quando atuam sobre dois operandos. Esta operação é chamada diádica. Ex.: os operadores das operações aritméticas básicas (soma, subtração, multiplicação e divisão).
- **Unários:** quando atuam sobre um único operando. Esta operação é chamada monádica. Ex.: o sinal de menos (-) na frente de um número, cuja função é inverter seu sinal.

Operadores de atribuição

Um operador de atribuição serve para atribuir um valor a uma variável.

Em Algoritmo usamos o operador de atribuição:

=

A sintaxe de um comando de atribuição é:

NomedaVariável = expressão

Operadores aritméticos

Os operadores aritméticos relacionam as operações aritméticas básicas.

- Adição: (+)
- Subtração: (-)
- Multiplicação: (*)
- Divisão: (/)
- Módulo - resto da divisão: (%)

Ordem	Operação	Símbolo
1ª.	Parênteses	()
2ª.	Potenciação	**
3ª.	Multiplicação, Divisão, Resto e Divisão Inteira	*, /, mod, div
4ª.	Adição, Subtração	+, -

Operadores relacionais

Os operadores relacionais são operadores binários que devolvem os valores lógicos “verdadeiro e falso”.

Operador	Comparação
>	maior que
<	menor que
>=	maior ou igual
<=	menor ou igual
=	igual
<>	diferente

Operadores lógicos

Os operadores lógicos ou booleanos são usados para combinar expressões relacionais. Também devolvem como resultado valores lógicos verdadeiro ou falso.

Operador	Tipo	Operação	Prioridade
OU	Binário	Disjunção	3
E	Binário	Conjunção	2
NÃO	Unário	Negação	1

Operadores literais

Os operadores que atuam sobre caracteres variam muito de uma linguagem para outra.

O operador mais comum e mais usado é o operador: **+**

Por exemplo, a concatenação das strings “ALGO” e “RITMO” é representada por:

“ALGO” + ”RITMO”

O resultado de sua avaliação é: **“ALGORITMO”**

EXPRESSÕES

O conceito de expressão em termos computacionais está intimamente ligado ao conceito de expressão ou fórmula matemática, em que um conjunto de variáveis e constantes numéricas relacionam-se por meio de operadores aritméticos compondo uma fórmula que, uma vez avaliada, resulta num valor.

EXPRESSÕES ARITMÉTICAS

Expressões aritméticas são aquelas cujo resultado da avaliação é do tipo numérico, seja ele inteiro ou real. Somente o uso de operadores aritméticos, variáveis numéricas e parênteses é permitido em expressões deste tipo.

Expressões lógicas

Expressões lógicas são aquelas cujo resultado da avaliação é um valor lógico verdadeiro ou falso. Nestas expressões são usados os operadores relacionais e os operadores lógicos, podendo ainda serem combinados com expressões aritméticas.

Expressões literais

Expressões literais são aquelas cujo resultado da avaliação é um valor literal (caractere). Neste tipo de expressões só é usado o operador de literais (+).

As seguintes regras são essenciais para a correta avaliação de expressões:

1. Deve-se observar a prioridade dos operadores: **Operadores de maior prioridade** devem ser avaliados primeiro. Se houver empate com relação à precedência, então a avaliação se faz da esquerda para a direita.

2. **Os parênteses** usados em expressões têm o poder de “roubar” prioridade dos demais operadores, forçando a avaliação da subexpressão em seu interior.

3. Entre os quatro grupos de operadores existentes, a saber, aritmético, lógico, literal e relacional, há uma certa prioridade de avaliação: **os aritméticos e literais** devem ser avaliados primeiro; a seguir, são avaliadas as com **operadores relacionais** e, por último os **operadores lógicos** são avaliados.