

Métodos de Representação de Algoritmos

Contudo, quando falamos em algoritmo sob o enfoque computacional, como já foi mencionado, não podemos utilizar uma descrição narrativa para representar um algoritmo.

Sendo assim, veremos dois métodos para representação de algoritmos:

- fluxograma – representação gráfica;
- pseudocódigo (português estruturado) – representação textual.

Métodos de Representação de Algoritmos

Os métodos em questão impõem regras e disponibilizam um conjunto reduzido de palavras/símbolos passíveis de serem utilizados (rigidez sintática). O objetivo é obter uma consistência semântica para a eliminação da ambiguidade intrínseca à linguagem natural.

Destacaremos agora uma vantagem e uma desvantagem de cada uma das formas mencionadas, de representação de algoritmos.

Métodos de Representação de Algoritmos

Fluxograma:

Vantagem – a representação gráfica é mais concisa que a representação textual.

Desvantagem – é necessário aprender a simbologia dos fluxogramas.

Pseudocódigo:

Vantagem – sua transcrição para qualquer linguagem de programação é quase que direta.

Desvantagem – é necessário aprender as regras de construção de um pseudocódigo.

Métodos de Representação de Algoritmos

Solução do problema de trocar a resistência de um chuveiro resolvido com um algoritmo representado em pseudocódigo.

Conforme foi mencionado são impostas regras e é definido um número restrito de ações. Neste caso as ações disponíveis são: pegar, largar, abrir, fechar, retirar e colocar.

Descrição Narrativa

Adquira uma resistência nova e localize o chuveiro a ser manipulado. Em seguida abra o chuveiro retirando a resistência defeituosa, coloque a resistência nova e feche o chuveiro. Após descarte a resistência defeituosa.

Pseudocódigo

1. Pegar (resistência nova);
2. Pegar (chuveiro);
3. Abrir (chuveiro);
4. Retirar (resistência defeituosa);
5. Colocar (resistência nova);
6. Fechar (chuveiro);
7. Largar (resistência defeituosa).

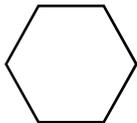
Métodos de Representação de Algoritmos

Solução do problema de trocar a resistência de um chuveiro resolvido com um algoritmo representado em fluxograma.

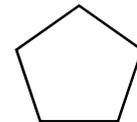
Conforme foi mencionado são impostas regras e é definido um número restrito de ações/símbolos. Neste caso, as ações disponíveis e os símbolos a elas associados são:

pegar $\langle \Rightarrow \rangle$ 

largar $\langle \Rightarrow \rangle$ 

fechar $\langle \Rightarrow \rangle$ 

abrir $\langle \Rightarrow \rangle$ 

colocar $\langle \Rightarrow \rangle$ 

retirar $\langle \Rightarrow \rangle$ 

Métodos de Representação de Algoritmos

Fluxograma

Descrição Narrativa

Adquira uma resistência nova e localize o chuveiro a ser manipulado. Em seguida abra o chuveiro retirando a resistência defeituosa, coloque a resistência nova e feche o chuveiro. Após descarte a resistência defeituosa.

Ações/Símbolos

pegar <=> 

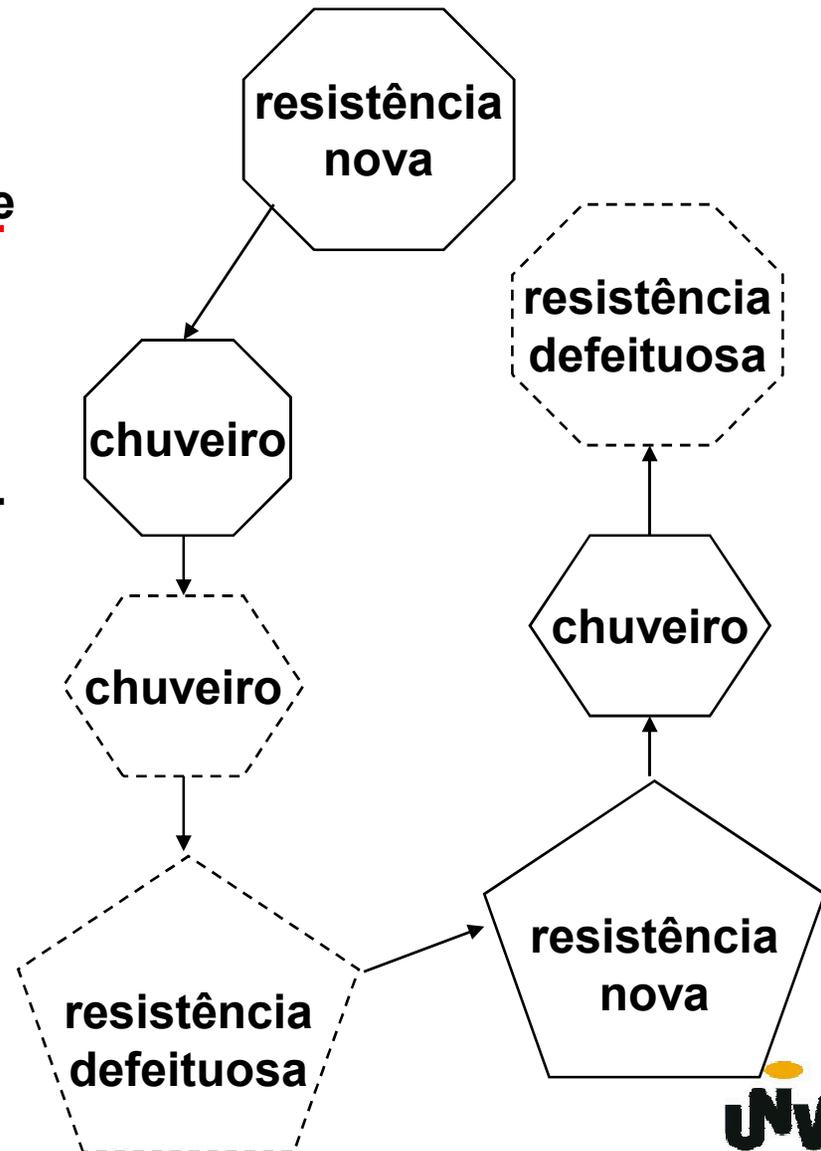
largar <=> 

fechar <=> 

abrir <=> 

colocar <=> 

retirar <=> 



Conceitos Básicos de Algoritmos

Antes de nos aprofundarmos nos métodos de representação de algoritmos, devemos ter de forma clara a compreensão de alguns conceitos como:

- Constante;
- Variável;
- Identificador;
- Palavra-reservada;
- Entrada;
- Saída;
- Operadores.

Conceitos Básicos de Algoritmos

Para uma melhor compreensão dos conceitos citados, analisaremos o problema de preparar uma omelete.

Partiremos da lista de ingredientes:

- 5 ovos
- 1/3 de uma xícara de chá de leite
- 2 colheres de sopa de cebolinha verde picada
- 3 pitadas de sal
- 1 colher de sopa de manteiga

Conceitos Básicos de Algoritmos

O procedimento de preparo, em linguagem natural, é o seguinte:

Colocar em uma tigela os ovos, o leite, a cebolinha e o sal. Com a ajuda de uma espátula bater bem os ingredientes contidos na tigela. Colocar a manteiga em uma frigideira e a derreter. Colocar o conteúdo da tigela na frigideira e fritar em fogo baixo até a omelete dourar suavemente. Quando estiver quase seca, dobrar a omelete ao meio, colocar a omelete em um prato. A omelete está pronta para servir.

Conceitos Básicos de Algoritmos

Ao analisarmos o procedimento de preparo percebemos que são necessários três recipientes: uma tigela, uma frigideira e um prato, além de uma espátula.

Podemos visualizar os recipientes como **variáveis**, pois estes são regiões do espaço onde pode-se armazenar inúmeras substâncias.

A espátula pode ser visualizada como uma **constante** devido a esta representar um objeto que participa do processo de preparo da omelete e que ao final se mantém inalterado.

Conceitos Básicos de Algoritmos

Percebemos também que são executadas algumas **operações**: colocar, bater, derreter, fritar, dobrar e servir.

Os nomes dos recipientes e os verbos correspondentes às ações podem ser visualizados como **identificadores**, uma vez que estes especificam o que será manipulado ou como será manipulado. Uma observação a ser feita é o fato de não podermos nomear um recipiente com um verbo correspondente a uma ação utilizada no processo, logo os verbos podem ser considerados como **palavras-reservadas**.

Os ingredientes constituirão as **entradas** para o processo e a omelete será a **saída**.

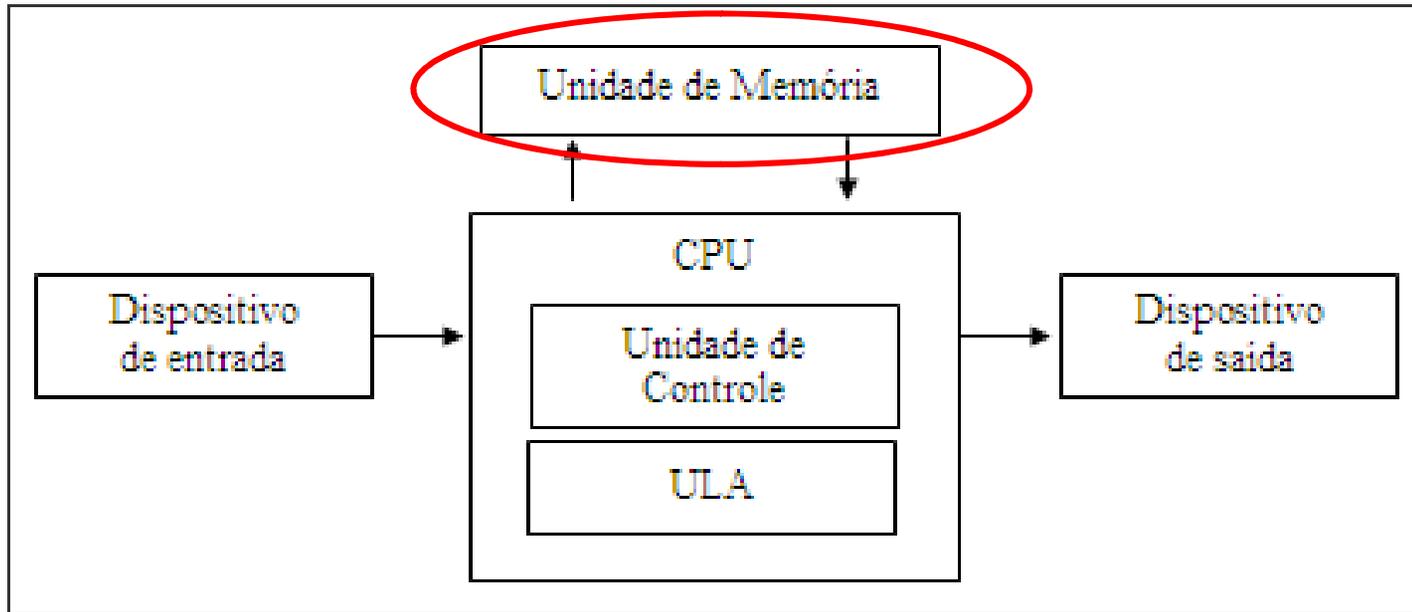
Conceitos Básicos de Algoritmos

Nosso objetivo final com o estudo de algoritmos é a aplicação computacional dos mesmos.

Desta forma, devemos contextualizar os conceitos vistos. Definiremos quais serão as entradas possíveis para os procedimentos que virão a constituir soluções de futuros problemas e especificaremos quais as formas de manipulação das mesmas.

Conceitos básicos de algoritmos

Arquitetura de John Von Neumann



Conceitos Básicos de Algoritmos

Constante →

São Valores fixos, tais como números. Estes valores não podem ser alterados pelas instruções do algoritmo, ou seja, é um espaço de memória cujo valor não deve ser alterado durante a execução de um algoritmo.

Exemplos:

inteiro (10, -23768)

real (-2.34, 0.149)

caractere (“a”, “professor”)

Conceitos Básicos de Algoritmos

Variável →

é um espaço de memória que recebeu um nome (identificador) e armazena um valor que pode ser modificado durante a execução do algoritmo.

Identificadores →

são os nomes utilizados para referenciar variáveis, funções ou vários outros objetos definidos pelo construtor do algoritmo.

- ✦ letras, dígitos e sublinhado(_);
- ✦ não podem começar com dígito;
- ✦ não podem ser iguais a uma palavra-reservada e nem iguais a um nome de uma função declarada pelo construtor do algoritmo ou disponibilizada pelo método utilizado para construção de algoritmos.

Conceitos Básicos de Algoritmos

Palavras-reservadas (palavras-chave) →

são identificadores predefinidos que possuem significados especiais para o interpretador do algoritmo.

inicio	senao	para	repita
var	logico	se	ate
faca	inteiro	enquanto	real

Conceitos Básicos de Algoritmos

TIPOS PRIMITIVOS

- Palavra-reservada: inteiro - define variáveis numéricas do tipo inteiro, ou seja, sem casas decimais
- Palavra-reservada: real - define variáveis numéricas do tipo real, ou seja, com casas decimais
- Palavra-reservada: caractere - define variáveis do tipo string, ou seja, cadeia de caracteres
- Palavra-reservada: logico - define variáveis do tipo booleano, ou seja, com valor VERDADEIRO ou FALSO

Conceitos Básicos de Algoritmos

DECLARAÇÃO DE VARIÁVEIS

- Palavra-reservada: var - utilizada para iniciar a seção de declaração de variáveis

Exemplos:

```
var a: inteiro
```

```
nome_do_aluno: caractere
```

```
senalizador: logico
```

```
Valor1, Valor2: real
```

Obs.: Sensível ao caso.

Conceitos Básicos de Algoritmos

Durante a execução de determinadas tarefas ou durante a execução de um processo para obtenção da solução de um determinado problema são necessários alguns dados e ao final deve-se retornar uma saída.

Da mesma forma, em alguns algoritmos tornam-se necessários mecanismos que possibilitem uma interface com o ambiente externo, ou seja, são necessários comandos que possibilitem a entrada e saída de dados.

Conceitos Básicos de Algoritmos

Entrada de dados

- Palavra-reservada: leia - utilizada para receber dados externos ao algoritmo e armazená-los na memória, ou melhor, em variáveis.

Exemplos:

...

```
var  a: inteiro  
     b: real
```

...

```
leia(a)
```

...

```
leia(b, a)
```

...

Conceitos Básicos de Algoritmos

Entrada de Dados

Tecnicamente podemos utilizar a seguinte definição: O comando de entrada de dados *leia* possui a sintaxe

`leia (<lista-de-variáveis>)`

Onde este recebe valores digitados pelo usuário, atribuindo-os às variáveis cujos nomes estão em <lista-de-variáveis> (é respeitada a ordem especificada nesta lista).

Conceitos Básicos de Algoritmos

Saída de dados

- Palavra-reservada: escreva - utilizada para externar (enviar para o monitor) dados gerados pelo algoritmo.

Exemplos:

...

```
var  x: inteiro
     y: caractere
```

...

```
escreval(y)
escreva (x,y)
escreva (“Inteiro: ”, x-2)
escreva(“Estudou e se dedicou tirou ”, 10)
```

...

Conceitos Básicos de Algoritmos

Saída de Dados

Tecnicamente podemos utilizar a seguinte definição: O comando de saída de dados **escreva** possui a sintaxe

`escreva (<lista-de-expressões>)`

Onde este escreve no dispositivo de saída padrão (monitor) o conteúdo de cada uma das expressões que compõem <lista-de-expressões>. As expressões dentro desta lista devem estar separadas por vírgulas; depois de serem avaliadas, seus resultados são impressos na ordem indicada.

Conceitos Básicos de Algoritmos

Saída de dados

É possível especificar o número de colunas da tela que se deseja reservar para escrever um determinado valor. Por exemplo, considerando uma variável inteira x , o comando `escreva (x:5)` escreve o valor da variável x em 5 colunas, alinhado-o à direita.

Para variáveis reais, pode-se também especificar o número de casas fracionárias que serão exibidas.

Por exemplo, considerando y como uma variável real, o comando `escreva(y:6:2)`, escreve seu valor em 6 colunas colocando 2 casas decimais.

Conceitos básicos de algoritmos

➤ Operador de Atribuição <-

Exemplo: var a: inteiro
 valor1, valor2: real

...

A <- 5

Valor1 <- 3.14

valor2 <- valor1

escreva(valor2)

...