

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ

ATIVIDADE AVALIATIVA DE INTRODUÇÃO A PROGRAMAÇÃO

DENILSON DOS SANTOS DA SILVA

TÉCNICO EM INFORMÁTICA

TIANGUÁ-CE

JUNHO/2017

Conceitos Algoritmos

Utilizando o livro lógica de programação: A construção de algoritmos e estrutura de dados.

1 - Leia o capítulo 1 e faça um resumo sucinto do que você aprendeu. (2pts)

A palavra Lógica é originária do grego “logos” significa linguagem racional, Lógica é a análise das formas e leis do pensamento, mas não se preocupa com a produção do pensamento, quer dizer, não se preocupa com o conteúdo do pensamento ou uma ideia é organizada e apresentada, possibilitando que cheguemos a uma conclusão. Um argumento pode ser composto por um ou várias premissas, as quais podem ser verdadeiras ou falsas e conduzem à conclusão, que também poderá ser verdadeira ou falsa.

EXEMPLO: 1 >> Kelly é mais velha do que Eduardo.

2 >> Eduardo é mais velho que Pedro.

3 >> Logo, Kelly é mais velha do que Pedro.

A lógica pode ser entendida como a área da filosofia que estuda o processo racional. Nossas afirmativas, decisões e atitudes muitas vezes são decorrentes de relações que estabelecemos mentalmente entre coisas e ideias (causa e efeito, adição, consequência, condições etc.) e que julgamos ter ou não coerência com nosso entorno. Talvez o papel da Filosofia seja avaliar o processo de julgamento de tais relações. Todavia, antes de entrar em contato com a lógica, importa compreender suas múltiplas facetas. Para perceber a importância da lógica em nossas vidas, não só na teoria, como na prática, já que, quando queremos pensar, falar, escrever ou agir corretamente, precisamos colocar ordem no pensamento.

EXEMPLO:

1 >> A gaveta está fechada.

A caneta está dentro da gaveta.

Precisamos primeiro abrir a gaveta para depois pegar a caneta.

2 >> Anacleto é mais velho que Marivaldo

Felisberto é mais velho que Marivaldo.

Portanto, Anacleto é mais velho que Marivaldo.

Lógica de programação permite definir a sequência lógica para desenvolvimento de um programa. Essa sequência lógica nada mais é do que uma sequência de instruções.

Instruções são as informações que indicam ao computador a ação que deve ser executada. Fazendo um analogia (Clichê) a preparação de bolo, manteiga, por açúcar, mexer, por no forno, etc. Mas não se pode executar essas instruções desordenadamente, por exemplo, não se vai por a tigela no forno, sem antes ter misturado os ingredientes do bolo. Essa sequência de instruções que seguem um ordem são denominadas de ALGORITMOS.

Um algoritmo é formalmente uma sequência finita de passos que levam a execução de uma tarefa. Podemos pensar em algoritmo como uma receita, uma sequência de instruções que dão cabo de uma meta específica. Estas tarefas não podem ser redundantes nem subjetivas na sua definição, devem ser claras e precisas.

Como exemplos de algoritmos podemos citar os algoritmos das operações básicas (adição, multiplicação, divisão e subtração) de números reais decimais. Outros exemplos seriam os manuais de aparelhos eletrônicos, como um videocassete, que explicam passo-a-passo como, por exemplo, gravar um evento.

Introdução à Lógica de Programação

Apostila voltada ao curso de Eletrônica – Disciplina: Aplicativos Informatizados Centro Paula Souza – Prof. Milton Rodrigues 4

Até mesmo as coisas mais simples, podem ser descritas por sequências lógicas. Por exemplo:

Chupar uma bala.

- 1 >> Pegar a bala
- 2 >> Retirar o papel
- 3 >> Chupar a bala
- 4 >> jogar o papel no lixo.

Os algoritmos são descritos em uma linguagem chamada pseudocódigo. Este nome é uma alusão à posterior implementação em uma linguagem de programação, ou seja, quando formos programar em uma linguagem, por exemplo Linguagem C, estaremos gerando código em Linguagem C. Por isso os algoritmos são independentes das linguagens de programação. Ao contrário de uma linguagem de programação, não existe um formalismo rígido de como deve ser escrito o algoritmo.

O algoritmo deve ser fácil de se interpretar e fácil de codificar. Ou seja, ele deve ser o intermediário entre a linguagem falada e a linguagem de programação.

Entretanto ao montar um algoritmo, precisamos primeiro dividir o problema apresentado em três fases fundamentais.

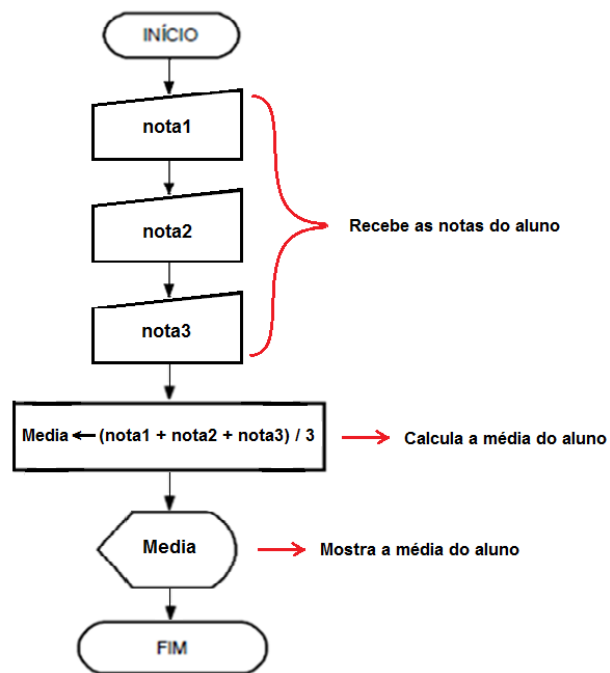
Onde temos:

ENTRADA: São os dados de entrada do algoritmo

PROCESSAMENTO: São os procedimentos utilizados para chegar ao resultado final

SAÍDA: São os dados já processados

Existem diversos símbolos em um diagrama de bloco. No decorrer do curso apresentaremos os mais utilizados. Veja no quadro abaixo alguns dos símbolos que iremos utilizar:



Dentro do símbolo sempre terá algo escrito, pois somente os símbolos não nos dizem nada.

2 - Na página 18, responda as três questões de exercícios de fixação. (1pt)

EXERCÍCIO DE FIXAÇÃO 2

RESPOSTAS

2.1 >> **U2, UYT, B52, ASDRUBAL, Rua, CEP.**

2.2 >>

NB >> NOTA DO ALUNO >> **REAL**

NA >> NOME DO ALUNO >> **CARACTERE**

NMat >> NÚMERO DA MATRÍCULA >> **INTEIRO**

SX >> SEXO >> **LÓGICO**

2.3 >> **REAL NÃO CORESPONDE A “R\$” E O “C” ESTA DECLARADO DUAS VEZES.**

3 - Na página 20, responda o exercício de fixação. (0.5pt)

A) A RESPOSTA É 9 B) A RESPOSTA É 1 C) A RESPOSTA É 34 D) A RESPOSTA É -54

E) A RESPOSTA É 67 F) A RESPOSTA É -7

4 - Na página 25, não esqueça de responder o exercício de fixação. (0.5 pt)

DADOS :

$$A = 2 ; B = 7; C = 3,5.$$

A) $B = A * C$ e (L ou V)
 $7 = 2 * 3,5$ e (F ou V)
 $7 = 2 * 3,5$ e V
 $7 = 7$ e V
Logo
V e V
V

B)
 $B > A$ ou $B = \text{pot}(A, A)$
 $7 > 2$ ou $7 = \text{pot}$
V ou $7 = 4$
V ou F
V

C)
 $L \text{ e } B \text{ div } A \geq C$ ou não $A \leq C$
 $F \text{ e } 7 \text{ div } 2 \geq 3,5$ ou não $2 \leq 3,5$
 $F \text{ e } 3 \geq 3,5$ ou não $2 \leq 3,5$
 $F \text{ e } F$ ou não V
 $F \text{ e } F$ ou F
 F ou F
F

D)
não L ou V e $\text{rad}(A + B) \geq C$
não F ou V e $\text{rad}(2 + 7) \geq 3,5$
não F ou V e $3 \geq 3,5$
não F ou V e F
V ou V e F
V ou F
V

E)

$B/A = C$ ou $B/A \neq C$

$7/2 = 3,5$ ou $7/2 \neq 3,5$

$3,5 = 3,5$ ou $3,5 \neq 3,5$

V ou F

V

F)

L ou $\text{pot}(B, A) \leq C \cdot 10 + A \cdot B$

F ou $\text{pot}(7, 2) \leq 3,5 \cdot 10 + 2 \cdot 7$

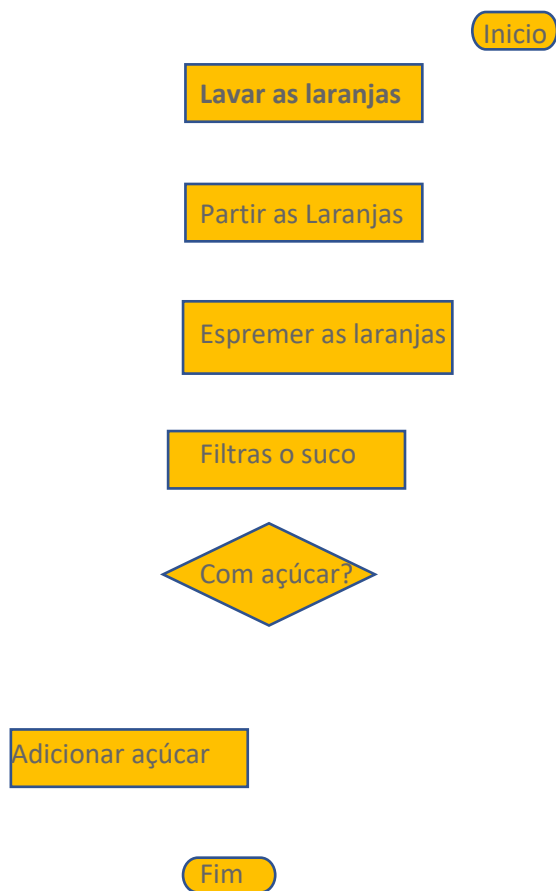
F ou $49 \leq 35 + 14$

F ou $49 \leq 49$

F ou V

V

5 - Dê exemplo do seu dia a dia em que você visualiza um algoritmo em três etapas: entrada, processamento e saída. (1pt)



Estruturas de Seleção

Utilizando o livro Introdução à programação: 500 algoritmos resolvidos

6 - Leia o capítulo 3 até a página 72 e faça um resumo sucinto do que você aprendeu. (2pts)

É uma estrutura de controle de fluxo, executando um ou vários comandos se a condição testada for verdadeira e em alguns casos, executando um ou vários comandos se for falsa.

Existem problemas que podem ter mais de um caminho a ser seguido para seleção correta, ou existem restrições em suas soluções.

O sujeito que irá executar o algoritmo, em dado momento, deve então tomar a decisão do caminho que deverá seguir para chegar à solução correta do problema.

Esta decisão, do caminho a ser seguido, tem de estar fundamentada em alguma lógica. E é função de quem escreve o algoritmo, dar condições para o sujeito que o executa, fazer a escolha correta do caminho para se chegar à solução do problema.

Um exemplo simples de problema que pode ter metodologias diferentes em sua solução, são os cálculos das raízes de equações polinomiais do 2º. grau ($ax^2 + bx + c = 0$). Se o termo $b^2 - 4ac$ for maior ou igual a zero, as raízes serão reais, caso contrário, serão complexas.

Não seria correto nós escrevermos um algoritmo que resolvesse somente a parte de raízes reais, ou, a parte complexa, pois a metodologia que vai ser usada somente será conhecida em tempo de execução do algoritmo. Logo, temos que descrever a solução por ambas as metodologias, mas deixando claro para o sujeito que vai executá-la, que ele terá de fazer uma escolha entre os caminhos possíveis.

Mas se deixássemos a critério de quem executa o algoritmo, será que ele sempre escolheria o caminho correto?

Pelo sim, pelo não, não podemos arriscar. Por isso, é que, além de dizermos que existe mais de um caminho para a solução do problema, ainda temos de prover subsídios para que o mesmo faça a escolha do caminho correto.

Assim, o simples fato de nós informarmos que a solução do problema tem duas ou mais metodologias diferentes em sua solução, não funcionam. Temos ainda, que dizer em quais situações ele utilizará a metodologia A, e em que outras ele utilizará a B.

No caso das raízes de polinômio do 2º. grau, poderíamos utilizar a seguinte descrição narrativa:

se o resultado de $b^2 - 4ac$ for maior ou igual a zero,
então utilize a metodologia de cálculo de raízes reais,
senão utilize a metodologia de cálculo de raízes complexas.

A estrutura se então permite determinar se um certo conjunto de instruções de um algoritmo deve ou não ser executada.

A estrutura se então senão nos permite fazer uma escolha entre duas alternativas mutuamente exclusivas.

7 - Faça um algoritmo que recebe dois valores do usuário e realiza a soma, então ele responde se o resultado da soma é par ou ímpar.(O algoritmo irá ajudar em uma disputa de ímpar ou par!!!). Utilize fluxogramas, passos ou código em javascript, você escolhe. (1pt)

```
var numero: inteiro
```

```
inicio
```

```
  escreva ("Digite um numero: ")
```

```
    se (numero mod 2 = 0) então
```

```
      escreva (numero, " É par")
```

```
    senão
```

```
      escreva (numero, " É ímpar")
```

```
fim
```

Estruturas de Repetição

Utilizando o livro Lógica de Programação. (Escola Alcides Maya)

8 - Leia o capítulo 5 (pág. 32), explique o funcionamento de cada um dos tipos de estruturas de repetição. (1pt)

Existem ocasiões em que é necessário efetuar a repetição de um trecho de programa um determinado número de vezes. Neste caso, poderá ser criado um looping que efetue o processamento de um determinado trecho, tantas vezes quantas forem necessárias. Os loopings também são chamados de laços de repetição ou malhas de repetição.

Supondo um programa que deva executar um determinado trecho de instruções por cinco vezes. Com o conhecimento adquirido até este momento, o leitor com toda a certeza iria escrever o mesmo trecho, repetindo-o o número de vezes necessárias. Por exemplo, imagine um programa que peça a leitura de um valor para a variável X, multiplique esse valor por 3, implicando-o a variável de resposta R, e apresente o valor obtido, repetindo esta sequência por cinco vezes

```
algoritmo "Pede_Numero"
```

```
var
```

```
X: inteiro
```

```
R: inteiro
```

```
inicio
```

```
    leia(X)
```

```
    R <- X * 3
```

```
    escreva(R)
```

```
    leia(X)
```

```
    R <- X * 3
```

```
    escreva(R)
```

```
    leia(X)
```

```
    R <- X * 3
```

```
    escreva(R)
```

```
    leia(X)
```

```
    R <- X * 3
```

```
    escreva(R)
```

```
    leia(X)
```

```
    R <- X * 3
```

```
    escreva(R)
```

```
fim
```

Para estes casos existem comando apropriados para efetuar a repetição de determinados trechos de programas o número de vezes que for necessário. A principal vantagem deste recurso é que o programa passa a ter um tamanho menor, podendo sua amplitude de processamento ser aumentada sem altera o tamanho do código de programação

Repetição do Tipo: Teste Lógico no Início do Looping

Caracteriza-se por uma estrutura que efetua um teste lógico no início de um looping, verificando se é permitido executar o trecho de instruções subordinado a esse looping. A estrutura em questão é denominada de enquanto, sendo conseguida com a utilização do conjunto de instruções enquanto...faca...fimenquanto.

A estrutura enquanto...faca...fimenquanto tem o seu funcionamento controlado por decisão. Sendo assim, poderá executar um determinado conjunto de instruções enquanto a condição verificada for Verdadeira. No momento em que esta condição se torna Falsa, o processamento

da rotina é desviado para fora do looping. Se a condição for Falsa logo de início, as instruções contidas no looping são ignoradas.

Repetição do Tipo: Teste Lógico no Fim do Looping.

Caracteriza-se por uma estrutura que efetua um teste lógico no fim de um looping. Esta estrutura é parecida com a enquanto. A estrutura em questão é denominada de repita, sendo conseguida com a utilização do conjunto de instruções repita...ate.

A estrutura repita...ate tem o seu funcionamento controlado por decisão. Porém, irá efetuar a execução de um conjunto de instruções pelo menos uma vez antes de verificar a validade da condição estabelecida. Diferente da estrutura enquanto que executa somente um conjunto de instruções, enquanto a condição é verdadeira.

Desta forma repita tem seu funcionamento em sentido contrário a enquanto, pois sempre irá processar um conjunto de instruções no mínimo uma vez até que a condição se torne Verdadeira. Para a estrutura repita um conjunto de instruções é executado enquanto a condição se mantém Falsa e até que ela seja Verdadeira.

Para exemplificar a utilização de repita, será utilizado o mesmo exemplo anterior: “Pedir a leitura de um valor para a variável X, multiplicar esse valor por 3, implicando-o à variável R, e apresentar o valor obtido, repetindo esta sequência por cinco vezes”.

Repetição do Tipo: Variável de Controle

Anteriormente, foram vistas duas formas de elaborar looping. Uma usando o conceito enquanto e a outra usando o conceito repita. Foi visto também como elaborar rotinas que efetuaram a execução de um looping um determinado número de vezes com a utilização de um contador (por meio de uma variável de controle).

Porém, existe uma possibilidade de facilitar o uso de contadores finitos sem fazer uso das duas estruturas anteriores, deixando-as para utilização de loopings em que não se conhece de antemão o número de vezes que uma determinada sequência de instruções deverá ser executada. Os loopings que possuem um número finito de execuções poderão ser processados por meio de estrutura de laços contados do tipo para, sendo conseguida com a utilização do conjunto de instruções para...de...ate...passo...faca...fimpara.

A estrutura para...de...ate...passo...faca...fimpara tem o seu funcionamento controlado por uma variável denominada contador. Sendo assim, poderá executar um determinado conjunto de instruções um determinado número de vezes.

Com relação à apresentação gráfica em um diagrama de blocos, existem várias formas adotadas por diversos profissionais no mercado, sendo assim, não existe um padrão de montagem do diagrama.

9 - Escolha dois exercícios sobre cada estrutura de repetição que o livro apresenta para você responder. (1pt)

