UNIVERSIDADE DO ALGARVE

INSTITUTO SUPERIOR DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA

ALGORITMIA E TÉCNICAS DE PROGRAMAÇÃO

Carlos Marinho

1. INTRODUÇÃO

Para a resolução de qualquer problema real há que estabelecer uma metodologia ou estratégia que permita, a partir das especificações desse problema, obter os resultados desejados.

Para um problema concreto podem-se encontrar, normalmente, várias estratégias de resolução.

A decisão de adopção de uma ou de outra estratégia será determinada por um conjunto de restrições donde se podem salientar:

a) Exigências quanto a tempos de resposta;

b) Recursos disponíveis:

bl) Qualidade do processador ou processadores;

b2) Capacidades de memória (principal e permanente);

b3) Periféricos ou interfaces de entrada e saída;

c) Utilização de *software* já existente.

Para aplicar um computador na resolução de uma determinada tarefa, não basta dispormos de uma linguagem de programação. Temos de saber definir quais as acções a desenvolver para executar essa tarefa, pois caso contrário não somos capazes de escrever o texto em linguagem de alto nível que irá dar origem ao programa em linguagem máquina pretendido.

1.1 Desenvolvimento de uma aplicação

No desenvolvimento de uma aplicação, podemos considerar as seguintes fases:

1.1.1 ANÁLISE

Na análise, deve-se entender o problema a resolver em termos da informação disponível, da informação que se pretende obter, quais os pressupostos, que casos particulares existem e qual o domínio dos dados. Esta fase é de grande importância, pois é a partir das informações recolhidas durante a análise que se especificam as funções que o programa deve cumprir.

1.1.2. CONCEPÇÃO

Nesta fase importa delinear um esquema lógico a implementar no computador, que deve respeitar as especificações funcionais determinadas na Análise. Esta fase é de grande importância para as seguintes, porque um bom esquema lógico permite a construção de programas eficientes, fáceis de corrigir e de alterar.

1.1.3 IMPLEMENTAÇÃO

Esta fase é constituída pela implementação do esquema lógico concebido na fase anterior, normalmente através da utilização de uma linguagem de programação. E aqui que se procede à escrita do texto do programa.

1.1.4 TESTE

Nesta fase o programa é testado para se verificar se respeita as especificações resultantes da Análise. No caso de existirem desvios a estas especificações, a sua origem pode localizar-se na concepção do modelo lógico ou na implementação.

1.2 ALGORITMO

O algoritmo é uma sequência finita, não ambígua, de passos ou operações bem definidas que levam à resolução do problema.

Para um problema concreto podem-se desenvolver várias estratégias de resolução. Têm sido estudadas e usadas diversas formas para ultrapassar a dificuldade de mudança brusca entre o enunciado do problema e a escrita da resolução codificada numa linguagem de programação, apresenta-se de seguida duas: Pseudo-código e Fluxograma.

1.2.2 PSEUDO-CÓDIGO

Utilização do "Português Estruturado", como linguagem intermédia de descrição de algoritmos, ou Pseudo-código.

1.2.3 FLUXOGRAMA

Utilização de simbologia pré-definida para a descrição dos algoritmos.

1.3 TIPOS DE DADOS

Conceptualmente podemos encarar um algoritmo como uma máquina (abstracta) de manipulação de dados.

Em qualquer problema, o objectivo é partir dum determinado conjunto de informação (dados que modelizam uma fracção do mundo real), processa-lo e obter outro conjunto de informação.

A descrição dum algoritmo deve traduzir a ênfase dos objectivos que manipula, e no caso de muitas linguagens de programação a noção de tipo de dados ocorre ligada à noção de objecto.

A especificação dum tipo de dados deve definir o conjunto de valores do tipo bem como as operações que se podem executar sobre ele.

De uma forma geral qualquer linguagem de descrição de algoritmos oferece alguns tipos de dados básicos, ou seja predefinidos, também chamados primitivos, bem como os mecanismos para criar outros.

São exemplos de tipos de dados primitivos:

a) Inteiros, num conjunto cujos limites inferiores e superiores depende de matemática e das limitações postas pela linguagem de programação. As operações que se podem usar para a sua manipulação são: a adição, subtracção, multiplicação, divisão inteira,…

b) Real, num conjunto de números racionais cujos limites tem a haver com a parte inteira por um lado e o numero de dígitos usados na descrição da parte fraccionária.

As operações sobre números reais são todas as designadas aritméticas bem como outras (potenciação,...).

c) Booleano, um conjunto de dois valores, verdadeiro e falso sobre o qual se pode efectuar operações lógicas (e, ou, não).

d) Caracter, o conjunto dos símbolos (A.. z e O.. 9, e outros símbolos especiais).

1.4 VARIÁVEIS E CONSTANTES

Já dissemos que um algoritmo manipula objectos, mas interessa manipula-los tendo em consideração não um valor permanente mas sim o conjunto de valores associado ao tipo de dados. O objecto pode então variar de valor e representar em determinado instante um valor em particular, designando-se assim por variável. Objectos há cujo valor se mantém ao longo de todo a execução do algoritmo, designando-se por constante.

2- MECANISMOS DE CONTROLO

Na construção de algoritmos, os seguintes mecanismos de controlo definem a forma de executar as operações de manipulação de dados:

- SEQUÊNCIA
- SELECÇÃO
  - SE
  - CASO
- REPETIÇÃO
  - ENQUANTO / FAZER
  - FAZER / ENQUANTO
  - DESDE / ENQUANTO / PASSO
- O PROCEDIMENTO OU FUNÇÃO

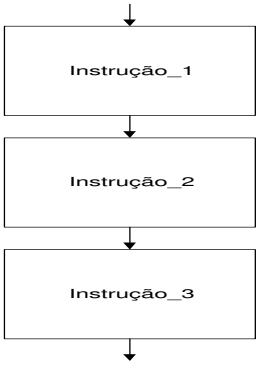
2.1 SEQUÊNCIA

As instruções indicadoras das acções a executar surgem no texto de uma forma sequencial e as operações que elas representam são executadas sequencialmente.

DESCRIÇÃO ALGORÍTMICA

.

Instrução_1
Instrução_2
Instrução_3



2.2 SELECÇÃO (SE)

DESCRIÇÃO ALGORÍTMICA .

Se <condição>

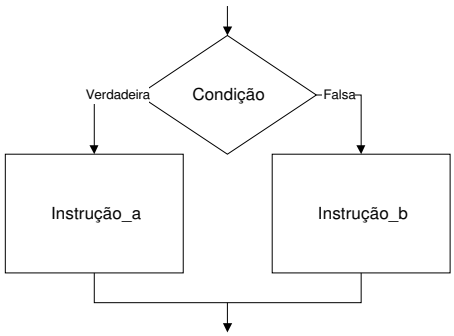
Então

Instrução_a

Senão

Instrução_b

Fim_Se



2.3 SELECÇÃO (CASO)

DESCRIÇÃO ALGORÍTMICA .

Caso <Expressão>

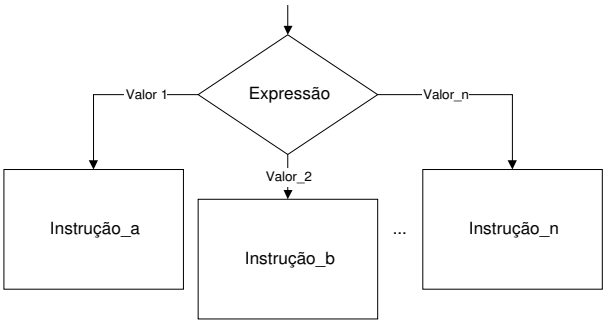
Valor_l: Instrução_a

Valor_2: Instrução_b

…

valor_n: Instrução_n

Fim_Caso



2.4 - REPETIÇÃO (ENQUANTO / FAZER)

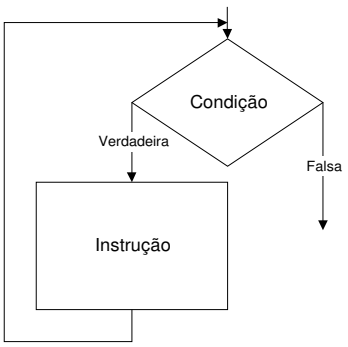
DESCRIÇÃO ALGORÍTMICA .

Enquanto <Condição>

Fazer

Instrução

Fim_Fazer



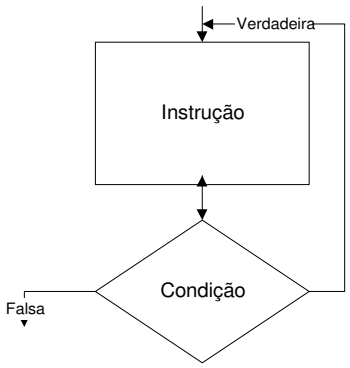
2.5 - REPETIÇÃO (FAZER / ENQUANTO)

DESCRIÇÃO ALGORÍTMICA .

Fazer

Instrução

Enquanto <Condição>



Nota: o teste da condição só ë feito depois da execução da instrução.

2.6 - REPETIÇÃO (DESDE / ENQUANTO / PASSO)

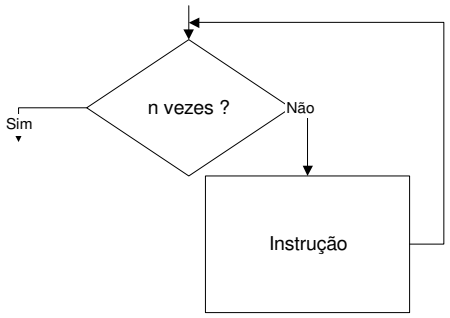
DESCRIÇÃO ALGORÍTMICA .

Desde <Condição de partida / Condição de execução / passo>

Fazer

Instrução

Fim fazer



2.7 - O PROCEDIMENTO OU FUNÇÃO

O Procedimento é um sub-algoritmo que actua sobre um conjunto de dados de entrada e fornece os resultados. Os dados de entrada e de saída (PARAMETROS) constituem o interface com o módulo assim implementado.

DESCRIÇÃO ALGORÍTMICA .

Procedimento Exemplo (Entrada: a,b; Saída: m,n)

corpo do procedimento

Fim_Procedimento

O desenvolvimento MODULAR vai permitir a construção de algoritmos em que é possível termos uma visão global do problema, sem nos preocuparmos com questões de detalhe. Os detalhes de cada módulo surgem numa análise mais individualizada, através de refinamentos sucessivos