

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Infraestrutura tecnológica
Programação e estruturas de dados

Módulo 2 – parte 2

Prof. Maurício B. Almeida
Escola de Ciência da Informação UFMG
<http://mba.eci.ufmg.br>
mba@eci.ufmg.br

Adaptado do curso de Informática Instrumental, DCC-UFMG

1

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Apresentação

Assunto abordado:

- Conceitos básicos de programação
- Estruturas de dados no computador

2

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Tópicos abordados

- Introdução à programação
- Programação Pascal
- Exemplos de programas básicos (diagramas, códigos)
 - Programa para desenhar retângulo simples
 - Programa para desenhar um retângulo de tamanho variável
 - Programa para desenhar quadrado ou retângulo
 - Cálculo do valor de multa em biblioteca
- Conceitos básicos de estruturas de dados

3

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Introdução à programação

O computador é uma ferramenta de auxílio a resolução de problemas, porém ele nada sabe fazer, apenas executa as ordens com bastante eficiência e rapidez.

As pessoas têm que pensar, resolver o problema e ensinar ao computador os passos necessários para que chegar a resposta desejada.

Ao conjunto destes passos, ou instruções, chamamos de **programa**: um conjunto de comandos em seqüência lógica, que permite o processamento de dados. É por seu intermédio que o computador é capaz de entender a tarefa a ser executada, ou seja, age como interface entre homem e máquina.

4

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Um programa de computador pode ser escrito em muitas **linguagens de programação** existentes, como por exemplo Delphi, Visual Basic, Pascal, Clipper, etc.

Uma linguagem de programação é um conjunto de instruções, símbolos e operadores, rigidamente controlados, sujeitos a regras.

A linguagem de programação funciona como intermediário entre o homem e a máquina. Um programa escrito numa linguagem de programação, em geral parecida com a linguagem humana, tem que ser **compilado**, ou traduzido, para ser executado pelo computador. A compilação transforma o **programa fonte** em **programa objeto**, que é a linguagem que o computador realmente entende.

5

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Os conceitos básicos aprendidos em uma linguagem de programação podem ser utilizados na maioria das linguagens, apenas a sintaxe é diferente.

Já existem inúmeros programas e pacotes no mercado com utilidades diversas. Então, por que aprender conceitos de programação?

- Pode ser que não exista no mercado um programa com as características que você deseja;
- Contratar alguém para implementar um programa pode não ser fácil ou ter um preço acessível;
- Possuindo noções básicas de programação pode-se utilizar de forma melhor os pacotes prontos.

6

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Programação Pascal

Programa para desenhar retângulo simples

Precisa-se desenvolver um programa para desenhar um retângulo com o lado maior de tamanho 100 e o lado menor de tamanho 50.

Para isso o primeiro passo é fazer um **algoritmo** com os passos a serem seguidos para se desenhar tal retângulo.

Trata-se de uma espécie de receita de bolo com todos os passos que devem ser seguidos para se obter o resultado que se deseja.

7

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Limpar Tela
Utilizar Lápis
Desenhar um traço de tamanho 50.
Virar 90° para direita
Desenhar um traço de tamanho 100.
Virar 90° para direita
Desenhar um traço de tamanho 50.
Virar 90° para direita
Desenhar um traço de tamanho 100.
Virar 90° para direita

Diagrama de Nassi com o algoritmo para se desenhar um retângulo com a especificação pedida.

8

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

```

program retangulo;
begin
  ul;
  clrscr;
  pfronte(50);
  pdireita(90);
  pfronte(100);
  pdireita(90);
  pfronte(50);
  pdireita(90);
  pfronte(100);
  pdireita(90);
  dr;
end.
    
```

O diagrama de Nassi representa a execução sequencial dos comandos, sendo que um comando só é executado após a execução do comando na linha anterior.

Na figura ao lado é apresentado um programa que implementa este algoritmo na linguagem de programação Pascal.

9

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

O programa anterior utiliza comandos Pascal para produzir um retângulo.

Na figura seguinte pode-se visualizar a figura produzida no ambiente de programação de Pascal. O ambiente Pascal usa o desenho de um robô que executa desenhos na tela.

Algumas áreas de interesse deste ambiente estão destacadas na figura.

10

MSWLogo Screen

Sair Rodar Compilar Abrir Novo

Barra de Operações

Área de desenho

Desenho produzido

Área de histórico

Área de comandos

Commander

[Arquivo C:\Temp\2001206610\retangulo.pas aberto]

Halt Trace
Pause Status
Step Reset

Execute

11

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

O programa para desenhar um retângulo exibido é simples. Basta desenhar um traço de tamanho 50 para cima (utilizando o comando **pfronte(50)**), virar o robô 90° à direita (através do comando **pdireita(90)**), desenhar um traço de tamanho 100 (através do comando **pfronte(100)**) e assim sucessivamente, até que o desenho esteja completo.

O comando **clrscr** utilizado limpa a área de comandos do Pascal. O comando **ul** (Use Lápis), faz com que o robô deixe um rastro ao se mover.

Caso se queira mover a posição do robô sem que ele desenhe um traço você deve utilizar o comando **un** (Use Nada).

12

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

O último comando utilizado no programa foi o **dr** (Desapareça Robô). Este comando faz com que o robô não seja exibido na tela. Caso se queira visualizar a figura novamente, utilize o comando **ar** (Apareça Robô).

A linguagem de programação é um "idioma intermediário" entre um programador e o computador.

Um programa em uma linguagem de programação deve ser "traduzido" para a linguagem da máquina (apenas 0's e 1's), para que o computador possa executá-lo.

Esta "tradução" no Pascal é realizada através do botão **compilar** da barra de operações. Para que o programa seja efetivamente executado e produza a figura desejada, você deve utilizar o botão **rodar** da barra de operações.

13

ARQUIVOLOGIA
Introdução a Informática

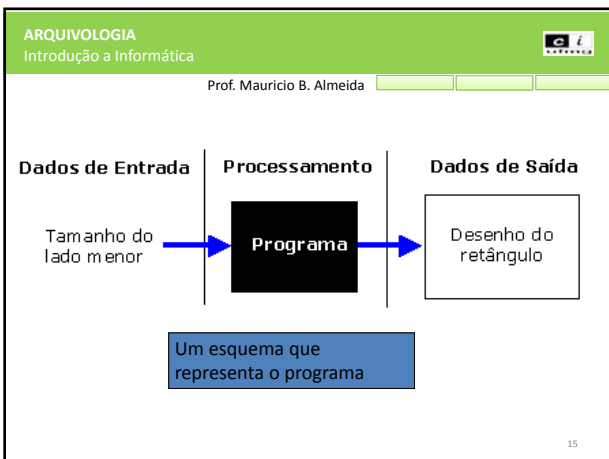
Prof. Mauricio B. Almeida

Programa para desenhar um retângulo de tamanho variável

Agora deseja-se que o tamanho dos lados do retângulo seja variável, mas que o lado maior seja sempre o dobro do tamanho do menor.

Suponha que o programa que executa a tarefa é uma caixa preta, que recebe como entrada o tamanho do lado menor e produz como saída o desenho de um retângulo.

14



15

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Limpar Tela
Utilizar lápis
Ler o tamanho do lado menor
Calcular tamanho do lado maior (2 x menor)
Desenhar um traço do tamanho do lado menor
Virar 90° para direita
Desenhar um traço do tamanho do lado maior
Virar 90° para direita
Desenhar um traço do tamanho do lado menor
Virar 90° para direita
Desenhar um traço do tamanho do lado maior
Virar 90° para direita

O diagrama de Nassi com um algoritmo para resolver o problema.

16

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Neste diagrama pode-se identificar os passos utilizados para se desenhar o retângulo.

O diagrama é semelhante ao do problema anterior, exceto pelo comando de leitura do tamanho do lado menor, que é um comando para entrada dos dados.

Ao executar um comando de entrada de dado, representado no diagrama de Nassi pela frase **Ler o tamanho do lado menor**, o computador espera que o usuário digite o dado pedido para então continuar a execução dos outros comandos do programa.

Na figura é apresentado o programa Pascal que implementa o algoritmo. A partir do tamanho do lado menor digitado pelo usuário, o programa desenha um retângulo.

17

Editor - C:\Temp\2001206610\retangulo.pas

```

program retangulo;
var ladoMenor: integer;
    ladoMaior: integer;
begin
  ul;
  clrscr;
  writeln('Entre com o tamanho do lado menor do retângulo: ');
  readln(ladoMenor);
  ladoMaior := ladoMenor * 2;

  pfronte(ladoMenor);
  pdireita(90);
  pfronte(ladoMaior);
  pdireita(90);
  pfronte(ladoMenor);
  pdireita(90);
  pfronte(ladoMaior);
  pdireita(90);
  dr;
end.
  
```

Cabeçalho (Declarações de Variáveis)

Corpo do Programa

18

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Todo programa pode ser dividido em duas seções: o **cabeçalho** e o **corpo do programa**. No cabeçalho encontram-se as *declarações de variáveis* e no corpo do programa encontram-se os *comandos* propriamente ditos.

Variáveis são utilizadas para se armazenar valores durante a execução do programa. Em Pascal as variáveis devem ser declaradas antes de serem utilizadas.

A declaração da variável consiste na especificação de seu tipo. No caso do programa acima, o tipo das variáveis é *integer*, o que significa que estas só poderão armazenar valores numéricos inteiros. Caso se precise armazenar valores como **3,14**, ou **1,5**, pode-se declarar uma variável do tipo *real*. Frases devem ser armazenadas em variáveis do tipo *String*.

19

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Os comandos são executados seqüencialmente, a não ser que haja um desvio de fluxo que são utilizados para "dizer" ao computador o que ele deve fazer.

Por exemplo, o comando *clrscr* é utilizado para que os desenhos existentes na tela do computador sejam apagados.

Para obter do usuário o tamanho do lado menor, este programa utiliza os comandos de entrada e saída *readln* e *writeln*.

O comando *writeln* faz com que a mensagem entre parênteses seja exibida na área de histórico do ambiente Pascal. É uma forma do programa interagir com o usuário, requisitando dados ou fornecendo resultados.

20

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

O comando *readln* é utilizado para que o usuário possa fornecer valores para o programa. Neste caso o programa precisa do tamanho do lado menor para poder desenhar o retângulo. O valor digitado pelo usuário é armazenado na variável entre parênteses.

Programa para calcular a área do retângulo

Suponha agora que, além de desenhar um retângulo com o tamanho do lado variável, é preciso calcular sua área e seu perímetro.

Um diagrama de Nassi contendo um algoritmo para resolver este problema encontra-se na figura abaixo:

21

Limpar Tela
Utilizar lápis
Ler o tamanho do lado menor
Calcular tamanho do lado maior (2 x menor)
Desenhar um traço do tamanho do lado menor
Virar 90° para direita
Desenhar um traço do tamanho do lado maior
Virar 90° para direita
Desenhar um traço do tamanho do lado menor
Virar 90° para direita
Desenhar um traço do tamanho do lado maior
Virar 90° para direita
Calcular área
Calcular perímetro
Imprimir área
Imprimir perímetro

O algoritmo é o mesmo mostrado anteriormente para desenhar o retângulo. Apenas os passos de calcular a área e o perímetro e a impressão destes dados é acrescentada.

22

```

program retangulo;
var ladoMenor: integer;
    ladoMaior: integer;
    area: integer;
    perimetro: integer;
begin
  clrscr;
  writeln('Entre com o tamanho do lado menor do retângulo: ');
  readln(ladoMenor);
  ladoMaior := ladoMenor * 2;

  pfrente(ladoMenor);
  pdireita(90);
  pfrente(ladoMaior);
  pdireita(90);
  pfrente(ladoMenor);
  pdireita(90);
  pfrente(ladoMaior);
  pdireita(90);

  area := ladoMaior * ladoMenor;
  perimetro := 2 * (ladoMaior + ladoMenor);

  writeln('A área do retângulo é: ', area);
  writeln('O perímetro do retângulo é: ', perimetro);
  dr;
end.
    
```

Uma implementação em Pascal do algoritmo para calcular a área

23

ARQUIVOLOGIA
Introdução a Informática

Prof. Maurício B. Almeida

Neste programa pode-se notar que para a impressão dos resultados é utilizado o comando *writeln*.

Este comando imprime na área de histórico do Pascal o texto e o valor da variável que se encontram entre os parênteses. Por exemplo, se o usuário entrar com o tamanho do lado menor 10 (ladoMenor = 10), o tamanho do lado maior será 20 e valor armazenado na variável *área* será 200 (=20 x 10).

Desta forma, o comando *writeln('A área do retângulo é: ', area);* fará com que seja impressa a seguinte frase: **A área do retângulo é: 200.**

24

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Ao final da execução do programa, um retângulo será exibido na área de desenho e na área de históricos serão impressos os seguintes resultados:

25

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Programa para desenhar quadrado ou retângulo

Nesse exemplo, de acordo com uma informação do usuário, deve-se decidir entre desenhar um quadrado ou um retângulo.

Na sequência:

- Descreve-se o problema;
- Apresenta-se o diagrama de Nassi;
- Apresenta-se o código do programa.

26

Problema

Você precisa implementar um programa que desenhe um quadrado ou um retângulo de forma eficiente, ou seja, você não deve replicar todo o código e sim reaproveitá-lo o máximo possível. A informação sobre o que deve ser desenhado deve vir do usuário, bem como o tamanho do lado menor do retângulo como visto no exercício anterior

Dados de Entrada

- Tamanho do lado do quadrilátero (caso seja um retângulo é o tamanho do lado menor, o lado maior tem o dobro deste comprimento).
- Indicação se deve ser desenhado um quadrado ou um retângulo (1 = quadrado e 2 = retângulo).

Processamento

- Ler os dados de entrada
- Calcular o tamanho do lado maior
 - Se for um quadrado, o lado maior é igual ao lado menor.
 - Se for um retângulo, o lado maior é o dobro do lado menor.
- Desenhar o quadrilátero.

Dados de Saída

- Desenho do quadrilátero pedido.

27

28

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Este diagrama possui uma estrutura de decisão (em azul na figura). Se o tipo da figura, indicado pelo usuário, for um quadrado, o lado maior deve ser igual ao lado menor. Caso contrário, o lado maior deve ser o dobro do lado menor. A decisão é representada pelo triângulo exibido na figura. Caso o tipo seja quadrado, os comandos da esquerda são executados caso contrário, os comandos da direita são executados.

Pode-se perceber neste diagrama é que uma parte do código se repete. As linhas grifadas delimitadas pelo círculo em vermelho são uma repetição anteriores. Para evitar a replicação do código pode-se utilizar um comando de repetição. Desta forma, o diagrama de Nassi ficaria:

29

30

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Neste diagrama estão destacados os comandos dentro da repetição.

A variável auxiliar *i* é utilizada para controlar o fim da repetição. Primeiramente esta variável é inicializada com 1. Enquanto *i* for menor ou igual a dois, os comandos dentro da repetição são executados.

São exatamente aqueles comandos que estavam repetidos no diagrama de Nassi anterior. O comando de incrementar a variável auxiliar *i*, ou seja, ($i := i + 1$), é muito importante. Se este comando não fosse utilizado, o valor de *i* nunca seria maior que 2 e o programa continuaria executando os comandos indefinidamente, é o que chamamos de *loop* infinito.

31

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Desta forma, os comandos dentro para desenhar meio quadrilátero são repetidos duas vezes (um para $i = 1$ e outra para $i = 2$), completando a figura.

Na figura abaixo você pode visualizar o código do programa Pascal que implementa o algoritmo mostrado acima.

32

```

program quadrilatero;
var ladoMenor: integer;
    ladoMaior: integer;
    tipo: integer;
    i: integer;
    area: integer;
    perimetro: integer;
begin
  writeln('Entrar com o tamanho do lado menor: ');
  readln(ladoMenor);
  if tipo = 1 then
  begin
    ladoMaior := ladoMenor;
  end
  else
  begin
    ladoMaior := ladoMenor * 2;
  end;
  i := 1;
  while i <= 2 do
  begin
    writeln('Área do retângulo é: ', area);
    writeln('O perímetro do retângulo é: ', perimetro);
    i := i + 1;
  end;
  area := ladoMaior * ladoMenor;
  perimetro := 2 * (ladoMaior + ladoMenor);
  writeln('Área do retângulo é: ', area);
  writeln('O perímetro do retângulo é: ', perimetro);
  dr;
end.
  
```

33

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

No Pascal a *estrutura de decisão* é representada pelo comando *if*, como podemos ver no quadro abaixo.

```

if [condicao] then
begin
  [bloco de comandos a serem executados caso a condição seja verdadeira]
end
else
begin
  [bloco de comandos a serem executados caso a condição seja falsa]
end
  
```

34

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

A *condição* é qualquer expressão lógica que tenha como resultado verdadeiro ou falso.

Caso esta condição seja verdadeira, os comandos após o *then* serão executados. Caso seja falsa, os comandos após o *else* serão executados.

Delimita-se esses dois blocos de comando com um *begin* e um *end*.

35

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

A *estrutura de repetição* é implementada no Pascal através do comando *while*. O significado deste comando pode ser vista no quadro abaixo:

```

while [condição] do
begin
  [grupo de comandos a serem executados enquanto a condição for verdadeira]
end;
  
```

Assim, condição é uma expressão que deve ter como resultado verdadeiro ou falso.

O grupo de comandos será executado enquanto a condição for verdadeira.

36

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Cálculo do valor de multa em biblioteca

Deve-se calcular o valor da multa para um usuário de uma biblioteca, de acordo com o número de livros em atraso. Deve ser calculado também o valor total das multas e a média de dias de atraso.

Na sequência:

- Descreve-se o problema;
- Apresenta-se o diagrama de Nassi;
- Apresenta-se o código do programa.

37

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Problema

Você deve fazer um programa para calcular o valor da multa para os livros de um determinado usuário, o valor total da soma de todas as multas e a média dos dias de atraso. Você deve ler o código de cada livro, o número de dias de atraso e se o livro está reservado ou não, até que seja digitado um código de livro negativo (FLAG) para encerrar a leitura dos dados. Se o livro não estiver reservado a multa é de R\$ 1,00 por dia, caso contrário é de R\$ 2,00 por dia. Caso o número de dias de atraso seja superior a 15 dias, é adicionada uma multa extra de 10% sobre o valor total da multa.

Dados de Entrada

Para cada livro do usuário deve ser informado:

- Código do livro
- Número de dias de atraso
- Informação se o livro está reservado ou não

38

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Processamento

- Ler repetidamente os dados de entrada
- Calcular o valor da multa para um livro sabendo que:
 - Se o livro não estiver reservado, a multa é de R\$ 1,00 por dia.
 - Se o livro estiver reservado, a multa é de R\$ 2,00 por dia.
 - Se o número de dias em atraso for maior que 15, o valor total da multa deve ser acrescido em 10%.
- Calcular o valor total das multas para todos os livros.
- Calcular a média dos dias de atraso.
- Exibir os dados de saída.

Dados de Saída

- Valor da multa para cada livro.
- Valor total das multas.
- Média dos dias de atraso.

39

```

totalMultas ← 0.0
numLivros ← 0.0
totalDias ← 0.0
Ler codLivro
Enquanto codLivro > 0
    Ler numDias de atraso
    Ler estaReservado
    estaReservado = 1?
    Verdadeiro      Falso
    multa ← numDias * 2 | multa ← numDias * 1
    numDias > 15?
    Verdadeiro      Falso
    multa ← multa * 1.1 | -
    totalMultas ← totalMultas + multa
    totalDias ← totalDias + numDias
    numLivros ← numLivros + 1
    Escrever codLivro
    Escrever multa
    Ler codLivro
Escrever totalMultas
numLivros > 0?
Verdadeiro      Falso
mediaDias ← totalDias / numLivros
Escrever mediaDias
    
```

40

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Para resolver este problema deve-se ler os dados para cada livro enquanto não for digitado código de livro menor que zero.

Isso foi feito pois se assumiu que o FLAG para encerrar a leitura dos dados é a utilização de um código de livro negativo. O FLAG é utilizado para que o usuário possa entrar com dados de um número indeterminado de livros. Quando ele quiser encerrar a leitura dos dados, basta utilizar o FLAG.

Para que se possa utilizar o FLAG para interromper a leitura dos dados, antes de iniciar a repetição deve-se ler o código do livro. Se o código for menor que zero, nada é feito. Como nenhum livro foi lido, não faz sentido calcular a média de dias de atraso, até porque este cálculo resultaria em uma divisão por 0, o que é um erro. Desta forma, o cálculo da média só é realizado caso o número de livros seja maior que zero.

41

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Para cada código de livro, devem ser lidos o número de dias de atraso e um número indicando se o livro está reservado ou não.

Para efeito de implementação assume-se que se o usuário entrar com o número 1, indica que o número está reservado, qualquer outro valor indica que o livro não está reservado. O valor da multa é o número de dias multiplicado pelo valor da multa diária. Caso o livro esteja reservado, o valor da multa diária é R\$ 2,00, caso contrário é R\$ 1,00.

Se o número de dias de atraso for maior que 15, o valor da multa deve ser acrescido de 10%. Isto é feito no diagrama de Nassi através do comando **multa ← multa * 1.1**.

42

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

O valor final da multa é então somado ao total das multas, o número de livros é incrementado e o número de dias de atraso é somado ao total de dias de atraso. Estes totais são preenchidos inicialmente com 0 antes de entrar na repetição, para que a soma seja correta.

Após a realização dos cálculos e a impressão da multa para o livro, os dados de um novo livro devem ser lidos. Lê-se então o código do livro. Se não for o FLAG continua-se a ler os dados. Caso o código lido seja um FLAG (menor ou igual a zero), a leitura é interrompida.

Após o fim da leitura de dados, o total das multas deve ser impresso na tela. A média do número de dias de atraso por livro deve ser calculada e impressa na tela, caso o número de livros lidos seja maior que zero.

43

Fragmentos do código do programa da biblioteca

```

program biblioteca;
var codLivro: integer;
    numDias: integer;
    estaReservado: integer;
    multa: real;
    totalMultas: real;
    numLivros: integer;
    totalDias: integer;
    mediaDias: real;
begin
    writeln('Entre com o código do livro: ');
    readln(codLivro);

    totalMultas := 0.0;
    numLivros := 0;
    totalDias := 0;

    while codLivro > 0 do
    
```

44

```

while codLivro > 0 do
begin
    writeln('Entre com o número de dias de atraso: ');
    readln(numDias);
    writeln('O livro está reservado? (1 para sim e 2 para nao)');
    readln(estaReservado);

    writeln('Esta reservado: ', estaReservado);
    if (estaReservado = 1) then
    begin
        multa := numDias * 2;
    end
    else
    begin
        multa := numDias * 1;
    end;

    if numDias > 15 then
    begin
        multa := multa * 1.1;
    end;

    totalMultas := totalMultas + multa;
    numLivros := numLivros + 1;
    totalDias := totalDias + numDias;

    write('Livro: ', codLivro);
    writeln(' Valor da multa: R$', multa);
    writeln('Entre com o código do livro: ');
    readln(codLivro);
end;

writeln('Valor total das multas: ', totalMultas);
if numLivros > 0 then
begin
    mediaDias := totalDias / numLivros;
    writeln('Média dos dias de atraso: ', mediaDias);
end;
end;
    
```

45

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

No código Pascal é necessário declarar todas as variáveis antes de utilizar. O número de dias, o código do livro e a variável *estaReservado* foram declaradas como **integer**, ou seja, só aceitam valores inteiros.

O valor da multa, no entanto, foi declarado como **real**. Isto ocorre pois, se o número de dias for maior que 15, somamos 10% ao valor da multa e isto pode resultar em um valor fracionário.

Uma outra variável definida como real é a média do número de dias de atraso. Isto ocorre pois a média é obtida através de uma divisão, e portanto, pode ser fracionária. Caso se deseje que a média seja um valor inteiro pode-se utilizar o operador **div** em invés da barra (/).

46

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Neste programa o comando **write** imprime o conteúdo entre parênteses na tela mas não cria uma nova linha, ao contrário do comando **writeln**.

Desta forma, o próximo texto ficará na mesma linha. Veja na coluna da direita da figura abaixo o resultado dos comandos da coluna da esquerda caso o valor de **codLivro** for 123 e o valor da **multa** for 4.

Código	Resultado Produzido
write('Livro: ', codLivro); writeln(' Valor da multa: ', multa);	Livro: 123 Valor da multa: R\$ 4

47

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Estruturas de dados

“Um algoritmo é uma sequencia de ações executáveis para a obtenção de uma solução para um determinado tipo de problema.” (Ziviani, 1999)

Estruturas de dados e algoritmos estão ligados. Não se pode estudar estruturas de dados sem considerar os algoritmos associados a elas, assim como a escolha dos algoritmos em geral depende da representação e da estrutura dos dados.

Para resolver um problema é necessário escolher uma abstração da realidade através de uma definição de um conjunto de dados que representa a situação real.

A escolha da representação de dados é determinada pela operações a serem realizadas sobre eles.

48

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Estruturas de dados básicas

Uma das formas mais simples de interligar os elementos de um conjunto é através de uma **lista**.

Trata-se de uma estrutura onde as operações inserir, retirar e localizar são definidas. São adequadas para aplicações onde não se pode prever a demanda por memória, permitindo a manipulação de quantidades imprevisíveis de dados, de formato também imprevisível.

Uma lista linear é uma sequência de zero ou mais itens
 x_1, x_2, \dots, x_n

Para definir um **tipo abstrato de dados lista** é necessário definir um conjunto de operações sobre objetos do tipo Lista.

49

49

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Um conjunto de operações necessárias a maioria das aplicações é:

- Criar uma lista linear vazia;
- Inserir um novo item imediatamente após o i-ésimo item;
- Localizar o i-ésimo item;
- Combinar duas ou mais listas lineares em uma única lista;
- Partir uma lista linear em duas ou mais listas;
- Fazer uma cópia da lista
- Ordenar a lista em ordem ascendente ou descendente;

50

50

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Uma **pilha** é uma lista linear em que todas as inserções, retiradas e acessos são feitas em apenas um extremo da lista.

Os itens de uma pilha estão colocados um sobre o outro, como o item inserido mais recentemente no topo e o menos recente no fundo. O último item inserido é o primeiro item que pode ser retirados na lista.

Estas propriedades tornam as pilhas ideais para processamento de estruturas aninhadas de profundidade imprevisível, situação em que é necessário garantir que as subestruturas mais internas sejam processadas antes das estruturas que as contenham.

51

51

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Estruturas aninhadas ocorrem frequentemente na computação, por exemplo na situação em que é necessário caminhar por um conjunto de dados e guardar uma lista de coisas a fazer posteriormente.

O **tipo abstrato pilha** possui operações usuais:

- Fazer a pilha vazia;
- Retornar o valor verdadeiro se a pilha está vazia e falso senão;
- Inserir itens no topo da pilha;
- Retirar um item do topo da pilha
- Retorna o número de itens da pilha

52

52

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Uma **fila** é uma lista linear em que todas as inserções são realizadas em um extremo da lista e todas as retiradas e acessos são realizados no outro extremo da lista.

Filas são utilizadas quando se deseja processar itens de acordo com a ordem "primeiro que chega, primeiro atendido".

Sistemas operacionais utilizam filas para regular a ordem na qual as tarefas devem receber processamento e recursos devem ser alocados a processos.

O conjunto de operações sobre a filas é similar ao das pilhas, contendo operações para: fazer a fila vazia, inserir item no final da fila, retornar um item ao início da fila, retornar valores verdadeiros se a fila está vazia e falso senão.

53

53

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Exemplos de Estruturas de dados

Listas

Uma lista é uma quantidade de itens de dados do mesmo tipo organizados de forma contígua na memória;

Exemplo – lista simples

Suponha que se quer controlar os jogadores de futebol que aparecem em um treino, através de uma lista. Pode ser preciso:

- Inserir um jogador na lista quando ele chega ao campo;
- Verificar se um jogador está presente;
- Excluir um jogador da lista quando ele vai embora;

54

54

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Exemplo – Lista Ordenada

Uma lista ordenada é uma lista em que os itens de dados estão organizados na ordem ascendente, ou seja, o menor valor está no índice zero e cada célula tem um valor maior que a célula seguinte;

Porque utilizar a lista ordenada? => velocidade na busca!!

Duas possibilidades de busca:

- 1) Busca linear: opera na mesma forma que em uma lista simples, mas quando o valor maior é encontrado a busca para;
- 2) Busca binária: utiliza o princípio do jogo de "advinhe o número" e dessa forma, é bem mais veloz;

55

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Classificação de dados

Classificação de bolhas

Ao se criar uma coleção de dados muitas vezes é necessário classificá-la, de forma a facilitar a busca;

Pode ser preciso ordenar nomes em ordem alfabética, alunos por notas, clientes pelo CEP, etc.

Exemplo


Imagine que o time de futebol deva ser organizado por ordem ascendente de altura;

Para fazer isso será preciso comparar dois itens adjacentes e, se necessário, trocá-los de lugar;

56

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida



Pessoas do time de futebol, em ordem aleatória

57

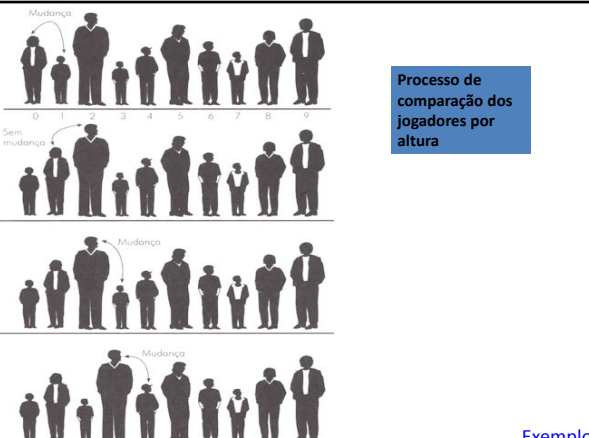
ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

O algoritmo "em português"

- Compare dois jogadores;
- Se o da esquerda for mais alto, troque-os de lugar;
- Mova-se uma posição para a direita e repita a operação;
- Continue até a ponta da direita: após essa primeira passagem realizaram-se N-1 comparações onde N=número de jogadores;
- Volte e comece outra passagem a partir da ponta da esquerda, comparando os jogadores adjacentes;
- Ao alcançar o primeiro jogador já classificado, comece de novo do final esquerdo da fila;
- Repita o processo até que todos os jogadores estejam em ordem

58



Processo de comparação dos jogadores por altura

[Exemplo](#)

59

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Classificação de inserção

A classificação anterior (por bolhas) é fácil de compreender mas notadamente lenta;

A classificação por inserção é mais rápida pois o número de comparações é menor

Baseia-se na marcação de um item da lista de forma que sejam feitas duas subclassificações;

[Exemplo](#)

60

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Pilhas

Uma pilha permite acesso apenas a um item de dados: o último item inserido;

Trata-se de uma abordagem do tipo “faça a coisa de cima primeiro”, por exemplo: ao receber correspondência as pessoas organizam uma pilha na mesa e a processam de cima para baixo, ou seja, abrem a primeira carta, pagam uma conta, jogam-na fora, armazenam... depois a segunda, e assim por diante.

As operações principais são inserir e remover itens na pilha;

[Exemplo](#)

61

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Filas

Uma fila é parecida com uma pilha, exceto pelo fato que se insere itens em uma ponta da fila e o remove pelo outro;

Filas são usadas para processar situações diversas em programação;

Retratam situações do mundo real como: pessoas esperando na fila de um banco, aviões esperando para decolar, pacotes de dados aguardando transmissão pela internet;

62

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Pessoas chegam ao final da fila

Pessoas saem da fila no início

Exemplo de uma fila no mundo real

[Exemplo](#)

63

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Recursão

Recursão é uma técnica de programação em que uma função chama a si mesma;

Apresenta limitações de eficiência mas facilita a solução do problema do ponto de vista conceitual;

Utilizada em conjunto com a abordagem “dividir para conquistar”, a qual envolve em geral uma função com contém duas chamadas recursivas a si mesmo;

Torre de Hanói - Exemplo

Transferir os discos da coluna A para a coluna C, sendo que apenas um disco pode ser movido por vez e nenhum disco pode ser colocado sobre um disco menor que ele mesmo;

64

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Árvores Binárias

Árvores binárias são estruturas que combinam vantagens de outras estruturas: permitem inserção, exclusão e busca mais rápidas;

Uma árvore consiste de nós conectados por arestas; trata-se de um instância de uma estrutura mais geral chamada grafo;

O caminho pelas arestas é denominado percurso;

O nó superior é chamado *raiz*; qualquer nó, exceto o raiz, tem exatamente uma borda dirigida para cima até outro nó, o qual é denominado *pai*; o nó inferior é denominado *filho*;

Nós que não tem filhos são chamados de *folhas*;

65

ARQUIVOLOGIA
Introdução a Informática

Prof. Mauricio B. Almeida

Uma árvore pode conter *subárvores*, com o seu próprio nó raiz;

Um nó é *visitado* quando o controle de um programa o alcança, em geral com a finalidade de realizar algumas operações com por exemplo, verificar seu valor;

O *nível* de um nó se refere a quantas gerações o nó está da raiz;

Em uma árvore binária só existem dois filhos para cada nós, um chamado *nó esquerdo* e outro *nó direito*.

[Exemplo](#)

[Aplicação do conceito de árvores](#)

66