Resumo Cobol - P1

Ppt aula2

Divisão das margens

- A (colunas 8 a 11).

- B (coluna 12 em diante).

Margens

A - Nomes de divisões, parágrafos e seções.

B - Todo o resto.

Coluna 7 – Quando marca que a próxima linha refere-se a:

\* => comentário

- => continua linha anterior

/ => salto de página

1 - Entrar no Visual Cobol

2 – File / New / Cobol Project

3 – Nome do Projeto / Click Finish

4 – Botão direito no Projeto / New / Cobol Program

5 – Nome do Programa / Click Finish

6 – Digitar o Programa na área de edição

7 – Click em Pay (run program)

**IDENTIFICATION DIVISION. - COLUNA 8**

**PROGRAM-ID. PGM00001. - COLUNA 8**

**AUTHOR.     FATEC. - COLUNA 8**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* - COLUNA 07**

**\*  C O M E N T A R I O S - COLUNA 07**

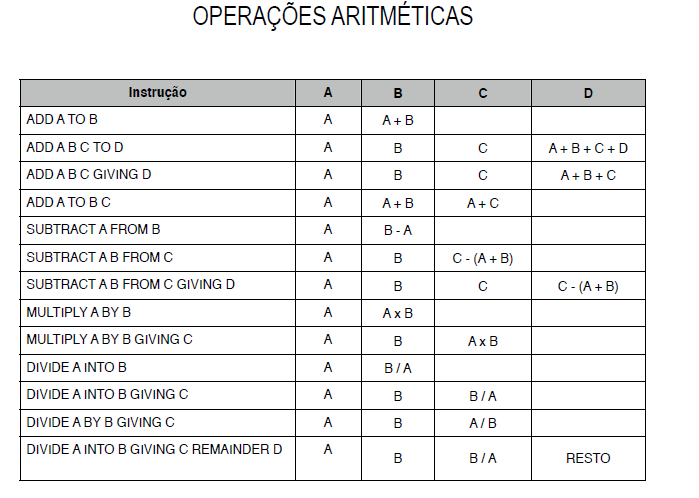
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* - COLUNA 07**

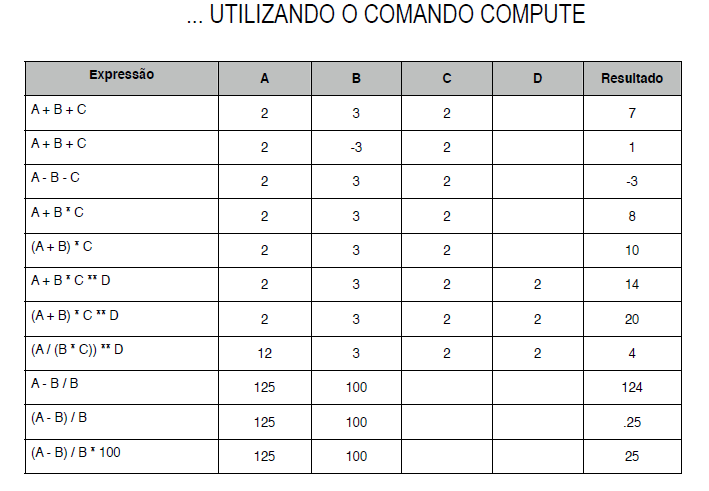
**PROCEDURE DIVISION. - COLUNA 8**

**DISPLAY ‘FATEC FATEC FATEC’. - COLUNA 12**

**STOP RUN. - COLUNA 12**

* **Todas as variáveis são descritas considerando:**
  1. **Tamanho = número de caracteres alocados**
  2. **Classe = tipo de variável**
  3. **Numérica = 9**
  4. **Alfabética = A**
  5. **Alfanumérica = X**
* **Os nomes de dados (ou variáveis):**
  1. **podem ter de 1 a 30 caracteres**
  2. **podem conter números, letras e hífen (-)**
  3. **não devem começar/terminar com hífen**
  4. **não podem ser uma palavra reservada do Cobol**





Divisão Básica de um programa Cobol

- Identification Division

- Environment Division

- Data Division

- Procedure Division

Identification Division

Sua função é identificar o programa, o nome de seu autor, a data em que foi escrito e compilado.

IDENTIFICATION DIVISION.

PROGRAM-ID. nome-do-programa.

AUTHOR. nome.

INSTALLATION. local. (op.)

DATE-WRITTEN. data-criação.. (op.)

DATE-COMPILED. data-compilação. (op.)

* Environment Division
* Executa duas funções:
* 1. Identifica o computador para compilação e execução na Configuration Section
* 2. Relaciona os arquivos usados no programa com os dispositivos de entrada e saída

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. [NOME EQUIP].

OBJECT-COMPUTER. [NOME EQUIP].

SPECIAL-NAMES.

DECIMAL POINT IS COMMA.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT ARQUIVO ASSIGN TO DISK

ORGANIZATION IS [INDEXED/SEQUENTIAL/RELATIVE]

ACCESS MODE IS [RANDOM/SEQUENTIAL/DYNAMIC]

RECORD KEY IS CAMPO-X

ALTERNATE RECORD KEY IS CAMPO-Y

FILE STATUS IS FS-W.

SELECT ARQUIVO ASSIGN TO DISK

SORT STATUS IS CAMPO-Z

SELECT RELATORIO ASSIGN TO PRINTER.

* **CONFIGURATION SECTION que trata das características do computador fonte e do computador objeto, sendo a clausula**
* **SPECIAL-NAMES que relata as implementações que o sistema usará nos programas fontes**
* **INPUT-OUTPUT SECTION que trata das informações necessárias para controlar a transmissão e a manipulação de dados entre o computador e o programa. Esta é dividida em dois parágrafos;**
* **FILE-CONTROL que dá o nome dos arquivos do programa e os associa aos meios externos**
* **I-O CONTROL que define técnicas de controle especial a serem usadas pelo programa**
* **SELECT Cláusula de seleção dos dispositivos de entrada e saída utilizados no programa, e relacionados com equipamentos externos;**
* **DISK arquivos manipulados em disco**
* **KEYBOARD console de entrada**
* **DISPLAY console de saída**
* **PRINTER especifica a impressora principal do sistema**
* **ORGANIZATION especifica a estrutura lógica do arquivo. Esta organização é estabelecida quando o arquivo é criado e não pode ser alterada.**
* **ACCESS MODE acompanha a forma pela qual o arquivo foi organizado, caso o arquivo seja indexado convém sempre que o acesso esteja definido no modo DYNAMIC (dinâmico).**
* **RECORD KEY nome de um campo do registro que vai ser defini como chave de acesso primária do arquivo; uma vez definido o campo chave não será conveniente altera-lo.**
* **ALTERNATE RECORD KEY nome de um campo do arquivo diferente do RECORD KEY, que funcionará como uma segunda chave ou chave alternativa, podendo definir até 80 chaves dentro de um único arquivo.**
* **FILE STAUTS nome de uma variável definida pelo programador DATA DIVISION, onde o sistema retorna o código na operação com arquivo. Esta variável esta subdividida em dois campos distintos STATUS KEY 1 e STATUS KEY 2**

IDENTIFICATION DIVISION.

PROGRAM-ID. SVP003.

\*

ENVIRONMENT DIVISION.

\*

INPUT-OUTPUT SECTION.

\*

FILE-CONTROL.

\*

SELECT ARQALU

ASSIGN TO DISK

ORGANIZATION INDEXED

ACCESS DYNAMIC

RECORD KEY MATRICULA

FILE STATUS AS-W.

SELECT RELDADOS

ASSIGN TO PRINTER.

***Data Division***

Descreve todos os dados que aparecem no programa. Está dividida em seções:

1 - FILE SECTION onde é defina a estrutura dos arquivos de dados

2 - WORKING-STORAGE SECTION onde é definida todas as variáveis de controle e informação

3 - REPORT SECTION específica arquivos de impressão, definição de relatórios.

4 - SCREEN SECTION definição de telas e seus recursos visuais.

DATA DIVISION.

FILE SECTION.

FD ARQALU

LABEL RECORD IS STANDARD

VALUE OF FILE-ID IS “ALUNOS.DAT”.

01 REGALU.

02 MATRICULA.

05 TURMA PIC 9(03).

05 CURSO PIC 9(03).

05 NUMERO PIC 9(05).

02 NOME PIC X(30).

02 RG PIC X(13).

FD RELDADOS

LABEL RECORD IS OMITTED.

* ***NÚMERO DE NÍVEL***
* Os número de nível são utilizados para demonstrar hierarquicamente a estrutura de um registro ou de áreas auxiliares.
* O número de nível para registro deve ser “01”, os campos devem ser classificados com um número de nível entre “02” e “49”, dependendo da posição hierárquica em ralação ao registro

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Código**  **(03)** | **Nome**  **(30)** | **Endereço** | | | **Pedido** | | |
| **Produto** | | **Preço**  **(07)** |
| **Rua (40)** | **Bairro(20)** | **Cidade(30)** | **Quant.(02)** | **Ref.(10)** |

01 REG-VENDAS.

02 CODIGO PIC 9(03).

02 NOME PIC X(30).

02 ENDEREÇO.

03 RUA PIC X(40).

03 BAIRRO PIC X(20).

03 CIDADE PIC X(30).

02 PEDIDO.

03 PRODUTO.

04 QUANTIDADE PIC 9(02).

04 REFERENCIA PIC 9(10).

03 PRECO PIC 9(05)V99.

* Depois de abrir o OpenCobol,
* Carregue o programa e no menu editar entre em Preferences,
* Selecione a caixa Run um externamente
* e no working diretory, selecione a pasta Bin do OpenCobol.

Ppt aula3

***WORKING-STORAGE SECTION***

Usada para declarar dados (variáveis) e constantes. Permite definir o layout de um relatório de impressão.

WORKING-STORAGE SECTION.

01 CONTADOR PIC 9(01).

02 DADOS.

05 RUA PIC X(30).

05 NUMERO PIC 9(03).

01 OPCAO PIC X(03).

***VALUE***

Indica o valor inicial de um dado. Deve ser consistente com PICTURE.

01 CONTADOR PIC 9(1) VALUE ZEROS.

01 OPCAO PIC X(3) VALUE "NAO".

01 OPCAO PIC X(3) VALUE SPACES.

***Sinal***

Deve-se inserir a letra S indicando que o dado descrito possui sinal. Quando um dado não possui S, ele é considerado positivo.

05 SALDO PIC S9(4)V99.

Se o valor de SALDO é -310040 teremos -3100,40 ;

se o valor for 5490 teremos 54,90.

***Símbolos de edição***

Os campos numéricos não podem conter outra coisa além de números, sinal e ponto decimal implícito. No entanto, para visualizar telas e relatórios é necessários cifrões, vírgulas, ausência de zeros à esquerda, etc. Para isto temos os símbolos (ou caracteres) de edição:

|  |  |
| --- | --- |
| **Símbolo** | **Significado** |
| . ou , | Ponto decimal explícito |
| Z | Suprime zeros |
| 0 | Apresenta zeros |
| \* | Na presença de zeros, preenche com asterisco |
| CR | Símbolo crédito - aparece quando valor positivo |
| DB | Símbolo débito - aparece quando valor negativo |
| + | Sinal positivo |
| - | Sinal negativo |
| $ | Cifrão |
| B | Representa um espaço em branco |
| / | Representa uma barra (/) no valor editado |

***SCREEN SECTION***

Onde definimos as telas usadas no programa. Na PROCEDURE DIVISION, as telas são acessadas pelos comandos DISPLAY e ACCEPT.

Exemplo:

SCREEN SECTION.

01 TELA-ENTRADA

02 BLANK SCREEN

02 LINE 01 COLUMN 20 VALUE "NOSSO PROGRAMA" REVERSE-VIDEO

02 LINE 02 COLUMN 20 VALUE "DADOS DO CLIENTE" HIGHLIGHT

02 LINE 04 COLUMN 25 VALUE "NOME: " UNDERLINE

02 LINE 06 COLUMN 25 VALUE "IDADE: " UNDERLINE

02 LINE 10 COLUMN 15 BLANK LINE

***Exemplos***

DATA DIVISION.

WORKING-STORAGE SECTION.

01 BASE PIC 9(04) VALUE ZEROS.

01 ALTURA PIC 9(04) VALUE ZEROS.

01 AREA PIC 9(04) VALUE ZEROS.

SCREEN SECTION.

01 TELA.

02 BLANK SCREEN.

02 LINE 04 COLUMN 21 VALUE "CÁLCULO DA ÁREA DE UM TRIÂNGULO".

02 LINE 10 COLUMN 21 VALUE "Insira o valor da altura: ".

02 LINE 12 COLUMN 21 VALUE "Insira o valor da base: ".

02 LINE 14 COLUMN 21 VALUE "Resultado do cálculo da área: ".

***PROCEDURE DIVISION***

Os parágrafos identificam a divisão das rotinas dentro da Procedure:

devem iniciar na margem A (colunas 8 a 11);

devem ter no máximo 120 caracteres sem aspas.

PROCEDURE DIVISION.

INICIO.

DISPLAY TELA.

DADOS.

ACCEPT (10,51) ALTURA.

ACCEPT (12,51) BASE.

CALCULO.

COMPUTE AREA = ( ALTURA \* BASE ) / 2.

FIM.

DISPLAY (14,51) AREA

DISPLAY (24,31) "FIM DE PROGRAMA"

STOP RUN.

***Comandos Básicos***

**DISPLAY** apresenta um dado ou tela

DISPLAY ERASE AT (01,01).

DISPLAY "DIGITE O NOME: “ AT (05,12).

DISPLAY NOME-CLIENTE AT (10,20).

DISPLAY TELA-ENTRADA.

**ACCEPT**  permite receber o conteúdo de uma variável

ACCEPT CODIGO AT (25,13).

**MOVE** serve para transferir um dado de um local para outro na memória

MOVE ZEROS TO TURMA.

MOVE CALCULO TO RESULTADO.

**STOP RUN** é utilizado para encerrar a execução do programa e devolver o controle dos recursos alocados do Sistema Operacional

STOP RUN.

**ADD** permite que um ou mais valores sejam somados a um ou mais campos

ADD NOTA-A NOTA-B TO NOTA-FINAL.

ADD 1 TO CONTADOR.

**SUBTRACT**  subtrai um ou mais itens de dados numéricos de um item especificado

SUBTRACT A B FROM C (C = C – (A + B))

SUBTRACT A B 2 FROM C (C = C – (A + B + 2))

SUBTRACT A B FROM 200 GIVING C

(C = 200 – (A + B))

**MULTIPLY** multiplica dois itens de dados numéricos

MULTIPLY A BY B (B = B \* A)

MULTIPLY A BY 5 GIVING B (B = A . 5)

**DIVIDE** divide dois valores numéricos

DIVIDE A BY B (B = B : A)

DIVIDE A BY B GIVING C (C = A : B)

**COMPUTE** calcula o valor de uma expressão aritmética

COMPUTE F = ( A + B ) / 2

**IF .. ELSE**  comando de seleção

IF WS-IDADE > 25

ROTINA-1

ELSE

ROTINA-2.

**Exercício:**

* Calcular a área de uma circunferência.

**Procedimentos:**

* Elaborar a tela formatada
* Aceitar o valor do raio
* Exibir a área calculada
* Exibir mensagem de "fim de programa"

IDENTIFICATION DIVISION.

PROGRAM-ID. Calculo-Area-Circunferencia.

ENVIRONMENT DIVISION.

Special-names.

Decimal-point is comma.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 DADOS.

02 W-Raio PIC 9(03)v99.

02 W-Area PIC 9(05)V99.

01 MENSAGEMS-DE-TELA.

02 MENSA1 PIC X(50) VALUE "DIGITE O Raio".

02 MENSA2 PIC X(30) VALUE “FIM DO PROGRAMA".

02 MENSA3 PIC X(30) VALUE SPACE.

01 DATA-DO-SISTEMA.

02 ANO PIC 9(02) VALUE ZEROS.

02 MES PIC 9(02) VALUE ZEROS.

02 DIA PIC 9(02) VALUE ZEROS.

SCREEN SECTION.

01 TELA01.

02 LINE 02 COLUMN 05 PIC 9(02)/ USING DIA.

02 LINE 02 COLUMN 08 PIC 9(02)/ USING MES.

02 LINE 02 COLUMN 11 PIC 9(02) USING ANO.

02 LINE 02 COLUMN 28 VALUE

"Cálcula da Área de um Circulo".

02 LINE 08 COLUMN 15 VALUE "Raio:".

02 LINE 10 COLUMN 15 VALUE "Área:".

PROCEDURE DIVISION.

Inicio.

ACCEPT DATA-DO-SISTEMA FROM DATE.

DISPLAY ERASE AT 0101.

DISPLAY TELA01 AT 0101.

MOVE ZEROS TO DADOS.

Entrada.

DISPLAY MENSA1 AT 2030.

ACCEPT W-Raio AT 0821.

DISPLAY MENSA3 AT 1830.

Calcula.

compute w-Area = 3,1416\*(w-Raio\*\*2).

Display w-Area AT 1021.

Finaliza.

DISPLAY MENSA2 AT 1830.

Stop " ".

Stop Run.

* Salva o Programa com o Nome de CalcArea.cbl
* Executar e observar os valores apresentados
* Colocar as variáveis EDITADAS
* Na Working-Storage Section inserir

01 DADOS-E.

02 W-Raio-E PIC ZZ9,99.

02 W-Area-E PIC ZZ.ZZ9,99.

* No parágrafo Entrada.

Entrada.

DISPLAY MENSA1 AT 2030.

ACCEPT w-Raio-E AT 0821.

Move W-Raio-E to W-Raio.

DISPLAY MENSA3 AT 1830.

* No parágrafo Calcula.

Calcula.

Compute w-Area = 3,1416\*(W-Raio\*\*2).

Move W-Area to W-Area-E.

Display w-Area-E AT 1021.

* Salvar e rodar o programa com as alterações
* Observar os valores apresentados
* Stop “ “.
* Stop Run.
* OU
* Call “C$SLEEP” using 50
* End Program.
* Rodar como Without Debuging no ambiente MicroFocus

Ppt aula 4

Exercício 2

Calcular a média aritmética das notas bimestrais de um aluno

Procedimentos:

Mostrar a tela formatada

Aceitar o nome do aluno e as notas

Mostrar a média aritmética

Se média inferior a 6,0

Entrar com a nota do Exame

Mostrar nova média aritmética

Exibir Aprovado ou Reprovado

Exibir "Fim de Programa"

Exercício 3

Elaborar um programa para calcular o custo de mercadoria de uma distribuidora. Os dados NÃO serão armazenado em arquivo.

Procedimentos:

mostrar a tela formatada

aceitar o nome do produto

aceitar a quantidade e preço da mercadoria

calcular o valor total da mercadoria

calcular o preço de venda, aumentando 30 % no valor total

condicionar o "fim de programa“

tamanho dos campos:

mercadoria = X(20) quantidade = 9(4)

preço unitário = 9(4)V99 preço total = 9(6)V99

preço de venda = 9(7)V99

DISTRIBUIDORA DE PRODUTOS LTDA

Nome da mercadoria: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Quantidade: \_\_\_

Preço unitário: $\_\_\_\_\_,\_\_

Preço total: $\_\_\_\_\_\_\_\_\_,\_\_

Preço de venda: $\_\_\_\_\_\_\_\_\_\_\_\_,\_\_

Continua (S/N) ? <\_>

Ppt aula 5

**Instrução EVALUATE.**

Implementa a estrutura CASE no Cobol. Elimina o uso de IF aninhados (encadeados).

Estrutura:

EVALUATE variável

WHEN caso-1

instruções

WHEN caso-2

instruções

...

WHEN OTHER

instruções

END-EVALUATE.

**Exemplos:**

ACCEPT OPCAO AT 1240.

EVALUATE OPCAO

WHEN 1

PERFORM INCLUSAO

WHEN 2

PERFORM EDICAO

WHEN OTHER

DISPLAY “OPCAO INCORRETA"

END-EVALUATE.

Uso de intervalos de valores e loop com TRUE

EVALUATE TRUE

WHEN MES = 4 OR 6 OR 9 OR 11

MOVE 30 TO DIAS

WHEN MES = 2

MOVE 28 TO DIAS

...

WHEN OTHER

DISPLAY "MÊS INCORRETO" AT 2040

END-EVALUATE.

Uso de intervalos de valores

EVALUATE NOTAS

WHEN 80 THRU 100

MOVE "A" TO NOTAS

WHEN 60 THRU 79

MOVE "B" TO NOTAS

...

WHEN OTHER

DISPLAY "NOTA INCORRETA" AT 2030

END-EVALUATE.

Exercício

Calcular a equação do segundo grau:

Ax²+Bx+C=0

Entrar com o valor de A

Restrição o valor de A <> 0

Entrar com os valores de B e C

Calcular o D (Delta)

D<0 => não existem raízes

D=0 => só existe um valor de X

D>0 => calcular as duas raízes X1 e X2

Condicionar o fim do programa

Ppt aula 6

* Conceito de Parágrafos para uso em programação estruturada
* A programação estruturada visa;
  + Redução de erros de programação
  + Redução nos custos de manutenção dos programas
  + Aumento da produtividade do programador
  + Codificação de programas fáceis de compreender
* ***Parágrafo e Perform***
* A Procedure Division, pode e deve ser dividida em parágrafos que devem começar obrigatoriamente na coluna 8 da área A, e os demais comandos ou verbos à partir da coluna 12 da área B, indo no máximo até a coluna 72 da área B.
* PROCEDURE DIVISION.
* INICIO.
* ACCEPT DATA-DO-SISTEMA FROM DATE.
* PERFORM PROCESSO UNTIL WS-CONTINUA = "N".
* PERFORM SAIDA.
* STOP RUN.
* PROCESSO.
* PERFORM TELA.
* MOVE ZEROS TO WS-FL.
* PERFORM ENTRA-DADOS UNTIL WS-FL = 1.
* PERFORM CALCULA UNTIL WS-CONTADOR > WS-NUMERO.
* PERFORM RESULTADO.
* TELA.
* DISPLAY ERASE AT 0101.
* DISPLAY TELA01 AT 0101.
* MOVE 2 TO WS-CONTADOR.
* MOVE ZEROS TO WS-NUMERO.
* MOVE 1 TO WS-RESULTADO.

ENTRA-DADOS.

ACCEPT WS-NOME AT 0832 WITH PROMPT AUTO.

ACCEPT WS-NUMERO AT 1232 WITH PROMPT AUTO.

IF WS-NUMERO > 10

DISPLAY MENSA1 AT 2310

ELSE

DISPLAY MENSA3 AT 2310

MOVE 1 TO WS-FL.

CALCULA.

COMPUTE WS-RESULTADO = WS-RESULTADO \* WS-CONTADOR.

ADD 1 TO WS-CONTADOR.

RESULTADO.

DISPLAY WS-RESULTADO AT 1432.

ACCEPT WS-CONTINUA AT 1628 WITH PROMPT AUTO.

SAIDA.

DISPLAY ERASE AT 0101.

DISPLAY MENSA2.

* Exercício
  + Transformar o programa de Equação do Segundo Grau em um programa estruturado

Ppt aula 7

Formatação do Nome do Arquivo

Até 30 caracteres

SEM caractere especial

SEM espaços no meio

ao menos uma letra

não começar nem terminar com hífen

Classificação dos arquivos:

Sequencial

Indexado

Relativo

Arquivo sequencial = registros são armazenados conforme a ordem em que foram gravados.

**ENVIRONMENT DIVISION.**

...

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT ARQ-CLIENTES ASSIGN TO DISK

ORGANIZATION LINE SEQUENTIAL

ACCESS MODE SEQUENTIAL

FILE STATUS ARQ-OK.

**INPUT-OUTPUT SECTION.**

Toda vez que o programa manipula um arquivo, esta seção é obrigatória.

**FILE-CONTROL.**

Serão descritas as características do(s) arquivo(s) que serão manipulados.

**SELECT ARQ-CLIENTES ASSIGN TO DISK**

O dispositivo de I/O de dados atribuído ao arquivo de nome ARQ-CLIENTES é o disco.

**ORGANIZATION LINE SEQUENTIAL**

Declaração do tipo da organização dos registros no arquivo.

**ACCESS MODE SEQUENTIAL**

Declara que o acesso aos dados do arquivo será sequencial.

* FILE STATUS ARQ-OK.
* A variável ARQ-OK recebe um código de verificação para as operações de I/O.
* Esta variável tem que ser definida na Working Storage com PIC X(02).
* Valores para arquivos sequenciais:
  + 00 = sem erros de I/O
  + 10 = encontrou EOF
  + 30 = erro permanente

**DATA DIVISION.**

FILE SECTION.

FD ARQ-CLIENTES LABEL RECORD STANDARD

RECORD CONTAINS 42 CHARACTERES

DATA RECORD IS REG-ARQCLI

VALUE OF FILE-ID IS "ARQCLI.DAT".

01 REG-ARQCLI.

02 CODIGO PIC 9(4).

02 NOME PIC X(30).

02 SALARIO PIC 9(5)V99.

FD ARQ-CLIENTES LABEL RECORD STANDARD

Descrição obrigatória. LABEL só é usado para discos/fitas. Para impressora, a descrição será LABEL RECORDS OMITTED.

RECORD CONTAINS 42 CHARACTERES

Opcional. Especifica o tamanho do registro.

DATA RECORD IS REG-ARQCLI

Opcional. Especifica que o nome do registro será REG-ARQCLI, cuja descrição de nível 01 vem a seguir.

VALUE OF FILE-ID IS "ARQCLI.DAT".

Obrigatória. Indica que o nome do arquivo especificado no disco é ARQCLI.DAT. Relaciona a descrição ARQ-CLIENTES com ARQCLI.DAT.

01 REG-ARQCLI.

Especifica o nome do registro e descrição dos campos.

Instrução OPEN

OPEN [ Tipo de Abertura] nome-arquivo.

Tipo de Abertura:

INPUT - abre o arquivo e já lê o primeiro registro. Somente para leitura.

OUTPUT - abre o arquivo e posiciona no primeiro registro. O arquivo é preparado para saída de dados. Caso já possua registros, estes serão perdidos.

I-O - apenas para arquivos em disco. Permite alteração de registros.

EXTEND - posiciona o arquivo para escrita no primeiro registro livre.

Instrução READ

READ nome-arquivo [ AT END instrução ].

Lê o registro do arquivo aberto com OPEN INPUT ou I-O.

Ao final do arquivo, as instruções declaradas após AT END serão executadas. O ponto deve ser colocado apenas na última instrução.

Enquanto não se atinge o EOF, a instrução seguinte ao READ é executada.

Exemplo:

READ ARQ-CLIENTES AT END DISPLAY MENSAGEM AT (21,30) STOP RUN.

IF CODIGO = 2010

COMPUTE ....

Instrução WRITE

WRITE nome-registro.

Grava o registro no arquivo aberto com OPEN OUTPUT, I-O ou EXTEND.

O nome do registro é aquele definido no nível 01 da FD da Data Division.

Instrução REWRITE

REWRITE nome-registro.

Substitui os dados de um registro de um arquivo aberto com I-O.

Antes do REWRITE, é preciso ler o registro com READ.

Crie um arquivo sequencial para registrar os produtos de uma empresa e seus respectivos custos (Inclusão)

Procedimentos:

mostrar a tela formatada

exibir a data do sistema

receber os campos

gravar o registro

Consistência:

custo unitário diferente de zero

custo total = qtidade x custo unitário

condicionar o fim do programa

Registros:

Código 9(4) / Nome X(30) / Qtidade 9(4)

Custo unitário 9(5)V99 / Custo total 9(6)V99

CONTROLE DE ESTOQUE \_\_/\_\_/\_\_

Código do produto:\_\_\_\_

Nome do Produto: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Quantidade: \_\_\_

Custo unitário: $\_\_\_\_\_,\_\_

Custo total: $\_\_\_\_\_\_\_\_\_,\_\_

Continua (S/N) ? [\_]

Mensagem: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ppt aula 8

**1. ENVIRONMENT DIVISION.**

...

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT ARQ-CLIENTES ASSIGN TO DISK

ORGANIZATION INDEXED

ACCESS MODE DYNAMIC

RECORD KEY CODIGO

FILE STATUS ARQ-OK.

ORGANIZATION INDEXED

Antes do acesso a cada registro, o programa consulta uma tabela de índice (IDX) associada ao arquivo.

ACCESS MODE DYNAMIC

Acesso ao arquivo pode ser tanto sequencial como randômico.

RECORD KEY CODIGO

Indica o campo que é a chave de pesquisa, p.ex., CODIGO.

Cada registro deve possuir um valor de CODIGO único.

Uso de outras chaves de pesquisa = ALTERNATE RECORD KEY IS NOME WITH DUPLICATES colocada após RECORD KEY.

FILE STATUS ARQ-OK.

Funciona como no arquivo sequencial, variável deve ser definida na Working

Valores para arquivos indexados:

00 = sem erros de I/O

10 = encontrou EOF

22 = chave duplicada

23 = registro não encontrado

**DATA DIVISION. (da mesma forma que arquivo sequencial)**

FILE SECTION.

FD ARQ-CLIENTES LABEL RECORD STANDARD

VALUE OF FILE-ID IS "ARQCLI.DAT".

01 REG-ARQCLI.

02 CODIGO PIC 9(4).

02 NOME PIC X(30).

02 SALARIO PIC 9(5)V99.

**Instrução READ**

READ nome-arquivo INVALID KEY instruções

[NOT INVALID KEY instruções]

END-READ.

Lê um arquivo pesquisando o registro segundo a chave RECORD KEY.

INVALID KEY = indica o que fazer quando o registro não foi encontrado.

NOT INVALID KEY = o que fazer quando encontrar o registro (opcional).

Se ocorrer sucesso no READ, a linha seguinte do programa é executada.

**Exemplo:**

ACCEPT CODIGO AT 1240.

READ ARQCLI INVALID KEY PERFORM ESTUDA-ERRO

END-READ.

DISPLAY CODIGO NOME AT 2035 ....

**Instrução WRITE**

WRITE nome-registro INVALID KEY instruções

END-WRITE.

Grava o registro no arquivo (OUTPUT ou I-O).

O valor da chave deve ser único.

Exemplo:

WRITE REG-ARQCLI INVALID KEY ESTUDA-ERRO.

**Instrução REWRITE**

REWRITE nome-registro INVALID KEY instruções

END-REWRITE.

Substitui os dados de um registro.

O registro é identificado pela chave. Não é preciso READ.

Exemplo:

ACCEPT CODIGO AT 1240.

ACCEPT NOME AT 1440.

REWRITE REG-ARQCLI INVALID KEY ESTUDA-ERRO.

**Instrução DELETE**

DELETE nome-arquivo INVALID KEY instruções

END-DELETE.

Remove o registro localizado com o último READ executado.

Exemplo:

ACCEPT CODIGO AT 1240.

READ ARQCLI INVALID KEY PERFORM ESTUDA-ERRO.

DELETE ARQCLI INVALID KEY PERFORM ESTUDA-ERRO.

Faça um programa que receba os dados de um produto e gravar em um arquivo indexado.

Procedimentos:

Exibir a tela apresentando a data do sistema

Receber os dados

Os campos devem ser apresentados na tela no formato editado.

A cada inclusão, deve-se perguntar se deseja salvar os dados.

Fim de programa condicionado.

Consistência:

Código diferente de zero e único para os produtos

Nome diferente de espaço

Qtdade maior que 10

Custo diferente de zero

Custo total = qtdade x custo unitário

Campos:

Codigo 9(4) Nome X(30) Qtdade = 9(4)

Custo unitário = 9(5)V99 Custo total = 9(6)V99