



Curso Profissional: Programador/a de Informática
PSD – 10.º ano: UFCD 0807 - Programação em Cobol – fundamentos

FICHA DE TRABALHO N.º 2

Ano letivo 21/22

Relembra-se que:

Número do Nível	Descrição
01	Descrição do registo entrada
02 a 49	Grupo e itens elementares
66	O nível 66 é utilizado para renomear outras variáveis do programa.
77	É utilizado para declarar variáveis que não irão possuir sub-itens, este nível é utilizado da mesma forma que o nível 01.
78	É utilizado para declarar constantes, ou seja, são referências que não podem ter o seu conteúdo alterado no programa.
88	É utilizado para representações de VERDADEIRO e FALSO.

PROCEDURE DIVISION

Local onde são processados todos os dados do programa, é aqui que são escritas as instruções.

Dentro da **PROCEDURE DIVISION** podemos separar as rotinas em secções e parágrafos de forma procedimental, ou num corpo de programa só (depende da necessidade). Ou então, manipulamos os ficheiros, recebemos e processamos inputs enviados pelo utilizador.

Para todas estas operações utilizam-se os comandos da linguagem (**verbos**), que devem também respeitar regras pré-estabelecidas.

Deve haver pelo menos uma declaração nesta divisão! A última instrução para terminar a execução nesta divisão pode ser **STOP RUN** o que implicará o fim da execução do programa (independentemente de existirem outras instruções após a mesma, fechando todos os ficheiros) ou **EXIT PROGRAM** usada em subprogramas devolvendo o controlo ao programa principal.

GOBACK, é também uma instrução que determina o final lógico de um programa específico ou até mesmo um método que foi chamado. Sempre que a instrução **GOBACK** é inserida no **COBOL**, é importante tê-la no final de todas as instruções concluídas. Qualquer declaração que apareça após a execução do **GOBACK** não será executada, tem um efeito semelhante ao **EXIT PROGRAM**.

Declarações/instruções

As declarações da linguagem **COBOL** são referenciadas como verbos.

Uma declaração começa com o nome do verbo e é seguida pelo operando ou operandos nos quais o verbo atua.

Os verbos principais do COBOL – Categorizados por tipo

Aritméticos	Manipulação de Ficheiros	Controlo de Fluxo	Atribuição & Input/output	Manipulação de Tabelas	Manipulação de Strings
COMPUTE ADD SUBTRACT MULTIPLY DIVIDE	OPEN CLOSE READ WRITE DELETE REWRITE START SORT RETURN RELEASE	IF EVALUATE PERFORM GO TO CALL STOP RUN EXIT PROGRAM	MOVE SET INITIALIZE ACCEPT DISPLAY	SEARCH SEARCH ALL SET	INSPECT STRING UNSTRING

Comando/verbo ACCEPT

Existem dois formatos para o comando ACCEPT:

- É utilizado para receber informação do teclado ou outros dispositivos de entrada;
- Permite aceder à data e hora do sistema (do relógio interno do PC), usando certas variáveis do próprio sistema DATE, DAY, TIME, DAY-OF-WEEK). A atribuição tem de seguir as regras do comando MOVE sem a opção CORRESPONDING.

FORMATO 1

ACCEPT <item, variável> [FROM nome_dispositivo]

Caso não se especifique qual o dispositivo de input, será considerado, por defeito, o teclado.

FORMATO 2

ACCEPT <item, variável> FROM { DATE [YYYYMMDD]
DAY [YYYYDDD]
DAY-OF-WEEK
TIME }

DATE: Sequência da data no formato: "AAMMDD", a PICTURE implícita é 9(06).

DAY: Data no formato JULIANO "AADDD", a PICTURE implícita é 9(05).

TIME: Hora no formato "HHMMSSDD", a PICTURE implícita é 9(08).

DAY-OF-WEEK: Dia da semana em formato numérico PIC 9. Onde 1 = Segunda-feira, 2 = Terça-feira, etc.

Exemplo:

WORKING-STORAGE SECTION.

77 HOJE PIC 9(06).

...

PROCEDURE DIVISION.

ACCEPT HOJE FROM DATE.

Neste exemplo o comando ACCEPT irá ler a data do sistema e armazenará essas informações no item HOJE.

Comando/verbo DISPLAY

O comando display é utilizado para exibir informações no ecrã ou outros dispositivos de output.

SINTAXE:

DISPLAY OutputItem1\$#il [OutputItem2\$#il]...
[UPONMnemonic-Name][WITH NO ADVANCING]

- { } as chavetas indicam que uma ou mais opções podem ser seleccionadas;
 - [] os parêntesis retos significam que as cláusulas são opcionais;
 - As reticências (...) indicam que o item pode repetir-se
- Os símbolos usados na sintaxe dos identificadores têm o seguinte significado:
- \$ indica que o item é uma string (alfanumérico) ;
 - # indica um item numérico;
 - i indica que o item pode ser um identificador de uma variável;
 - l indica que o item pode ser um literal.

A cláusula WITH NO ADVANCING – assegura que o cursor não se desloca para a linha seguinte

Exemplos:

DISPLAY PrinterSetupCodes UPON PrinterPort1	MOSTRE PRINTERSETUPCODES PARA O DISPOSITIVO ASSOCIADO AO NOME PRINTERPORT1
DISPLAY "Um," WITH NO ADVANCING DISPLAY "Dois," WITH NO ADVANCING DISPLAY "Três."	MOSTRE 'Um,' sem mudar de linha MOSTRE 'Dois,' sem mudar de linha MOSTRE 'Três,' e neste caso muda de linha
DISPLAY 1 " , " 2 " , " 3 " ."	MOSTRE 1 , 2 , 3 .
WORKING-STORAGE SECTION. 77 NOME PIC X(030) VALUE "JOÃO DE SOUZA MELLO". PROCEDURE DIVISION. DISPLAY "Nome do Cliente: " NOME GOBACK.	MOSTRE "Nome do Cliente:" E O NOME
WORKING-STORAGE SECTION. 77 DATAATUAL PIC 9(008). 77 HORAATUAL PIC 9(008). PROCEDURE DIVISION. ACCEPT DATAATUAL FROM DATE ACCEPT HORAATUAL FROM TIME DISPLAY DATAATUAL DISPLAY HORAATUAL DISPLAY DATAATUAL(7:2) "/" DATAATUAL (5:2) "/" DATAATUAL(1:4) GOBACK.	RECEBA DATAATUAL DA DATA DO COMPUTADOR RECEBA HORAATUAL DA HORA DO COMPUTADOR MOSTRE O CONTEUDO DE DATAATUAL MOSTRE O CONTEÚDO DE HORAATUAL MOSTRE OS CARATERES 7 E 8 DE DATAATUAL "/" OS CARATERES 5 E 6 "/" E OS CARATERES DE 1 A 4

MODIFICAÇÃO REFERENCIAL (Reference Modification) – ver exemplo acima

Permite trabalhar com partes de campos (substring/subconjunto de dígitos), e isso é determinado informando-se a posição inicial e o número de carateres/dígitos a mover/mostrar separados por “:”.

O seu formato é: CAMPO (x : y)

Y = Número de bytes à direita, a partir da posição inicial. Se omissa, será a partir da posição inicial até o fim do campo.

X = Posição inicial.

Reference Modification pode ser utilizado apenas em campos no formato DISPLAY.

Comando/verbo MOVE

Este verbo é um dos mais usados em COBOL pois é o que transfere/move dados, ou seja, associa um determinado valor ou um item a outro ou outros itens de dados.

NOTA: Repara que ao contrário de outras linguagens a transferência de informação é efetuada da esquerda para a direita.

SÍNAXE:

MOVE Source\$#il **TO** Destination\$#i...

Combinações válidas e inválidas de aplicação do verbo MOVE

Source Item	Destination Item									
ALPHANUMERIC										
	ALPHANUMERIC	Y								
NUMERIC INTEGER		Y	Y							
				Y	Y					
NUMERIC NON-INTEGER						Y	Y			
								Y	Y	
NUMERIC EDITED										Y
ALPHANUMERIC EDITED										
ALPHABETIC										

Edited – tem a ver com o formato da picture

Exemplos:

1. Dados 2 campos, C1 e C2, sendo C1="TESTE" e C2= "FICHA"

Após a instrução: **MOVE C1 TO C2**

Obtém-se: C1 = "TESTE" e C2 = "TESTE"

2. Podem-se "mover" dados para vários campos em simultâneo

Exemplo: **MOVE VALOR TO PRECO1 PRECO2 PRECO3**

3.

01 Surname PIC X(8) VALUE "COUGHLAN".

MOVE "SMITH" TO Surname

MOVE "FITZWILLIAM" TO Surname

Surname							
C	O	U	G	H	L	A	N
S	M	I	T	H	*	*	*
F	I	T	Z	W	I	L	L

Nota: Como a string "SMITH" é muito curta os restantes bytes são preenchidos com espaços.

Os últimos 3 caracteres de "FITZWILLIAM" são truncados porque excedem o tamanho definido para a variável Surname.

4

<p>a)</p> <p>WORKING-STORAGE SECTION.</p> <p>77 NOME PIC X(030) VALUE SPACES.</p> <p>77 OUTRONOME PIC X(030) VALUE SPACES.</p> <p>PROCEDURE DIVISION.</p> <p>ACCEPT NOME</p> <p>MOVE NOME TO OUTRONOME</p> <p>GOBACK.</p>	<p>RECEBA A VARIÁVEL</p> <p>COPIE O CONTEÚDO DE NOME PARA OUTRONOME</p>
---	---

2. Quando estivermos a mover para campos numéricos:

- a) O alinhamento é feito pelo ponto decimal e as posições não usadas são preenchidas com zeros.
- b) Se o campo emissor tem mais dígitos à direita ou à esquerda da vírgula decimal, esses dígitos em excesso serão truncados à direita e/ou à esquerda no campo recetor.

Exercícios:

1. Apresente o output dos seguintes programas em COBOL:

A)

```
IDENTIFICATION DIVISION.
PROGRAM-ID. UFCD0807-F2EX1-A.
DATA DIVISION.
    WORKING-STORAGE SECTION.
        01 WS-NUM1 PIC S9(3)V9(2).
        01 WS-NUM2 PIC PPP999.
        01 WS-NUM3 PIC S9(3)V9(2) VALUE -123.45.
        01 WS-NAME PIC A(6) VALUE 'ABCDEF'.
        01 WS-ID PIC X(5) VALUE 'A121$'.
PROCEDURE DIVISION.
    DISPLAY "WS-NUM1 : "WS-NUM1.
    DISPLAY "WS-NUM2 : "WS-NUM2.
    DISPLAY "WS-NUM3 : "WS-NUM3.
    DISPLAY "WS-NAME : "WS-NAME.
    DISPLAY "WS-ID : "WS-ID.
STOP RUN.
```

B.

```
IDENTIFICATION DIVISION.
PROGRAM-ID. UFCD0807-F2EX1-B.
DATA DIVISION.
    WORKING-STORAGE SECTION.
        01 WS-NUM1 PIC 99V9 VALUE IS 3.5.
        01 WS-NAME PIC A(6) VALUE 'ABCD'.
        01 WS-ID PIC 99 VALUE ZERO.
PROCEDURE DIVISION.
    DISPLAY "WS-NUM1 : " WS-NUM1.
    DISPLAY "WS-NAME : " WS-NAME.
    DISPLAY "WS-ID : " WS-ID.
STOP RUN.
```

C.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. UFCD0807-F2EX1-C.  
DATE-WRITTEN. 12-04-2019.  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
01 DataNasc.  
    02 Ano.  
        03 seculoAnt PIC 99.  
        03 anoNasc PIC 99.  
    02 mesNasc pic 99.  
    02 diaNasc pic 99.  
PROCEDURE DIVISION.  
    MOVE 19750218 TO Datanasc  
    DISPLAY "Mês: " mesNasc  
    DISPLAY "século: " seculoAnt  
    DISPLAY "Ano: "ano  
    DISPLAY "data de nascimento: "diaNasc/"mesNasc/"anoNasc  
    DISPLAY "data de nascimento: "Datanasc  
STOP RUN.  
END PROGRAM UFCD0807-F2EX1-C.
```

D.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. UFCD0807-F2EX1-D.  
*> Programa para mostrar a declaração MOVE  
AUTHOR. TPI3.  
date-written. 17/04/2019.  
DATA DIVISION.  
    WORKING-STORAGE SECTION.  
        01 UltimoNome      PIC X(9)  VALUE "FREI".  
PROCEDURE DIVISION.  
InicioPrograma.  
    DISPLAY UltimoNome  
    MOVE "PINTO" TO UltimoNome  
    DISPLAY UltimoNome  
    MOVE "HEITORHeitor" TO UltimoNome  
    DISPLAY UltimoNome  
STOP RUN.
```

E.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. UFCD0807-F2EX1-E.  
*> Programa para mostrar a declaração MOVE  
DATA DIVISION.  
    WORKING-STORAGE SECTION.  
        01 PrecoVenda PIC 9(4)V99.  
  
PROCEDURE DIVISION.  
InicioPrograma.  
    DISPLAY PrecoVenda  
    MOVE ZEROS TO PrecoVenda  
    DISPLAY PrecoVenda  
    MOVE 25.5 TO PrecoVenda  
    DISPLAY PrecoVenda  
    MOVE 7.553 TO PrecoVenda  
    DISPLAY PrecoVenda  
    MOVE 93425.158 TO PrecoVenda  
    DISPLAY PrecoVenda  
    MOVE 128 TO PrecoVenda  
    DISPLAY PrecoVenda  
STOP RUN.
```

F)

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. UFCD0807-F2EX1-F.  
*> Programa para mostrar a declaração MOVE  
AUTHOR. Professora.  
date-written. 10/03/2018.  
DATA DIVISION.  
    WORKING-STORAGE SECTION.  
        01 NumdeEmpregado PIC 999.  
        01 Salario PIC 9999V99.  
        01 NomePais PIC X(9).  
  
PROCEDURE DIVISION.  
BEGIN.  
    MOVE 12.4 TO NumdeEmpregado  
    DISPLAY NumdeEmpregado  
    MOVE 6745 TO NumdeEmpregado  
    DISPLAY NumdeEmpregado  
    MOVE NumdeEmpregado TO Salario  
    DISPLAY Salario  
    MOVE "PERU" TO NomePais  
    DISPLAY NomePais  
    MOVE ALL "@" TO NomePais  
    DISPLAY NomePais  
STOP RUN.
```


G)

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. UFC0807-F2EX1-G.  
*> Programa com a declaração ACCEPT com funções especiais  
AUTHOR. Professora.  
date-written. 10/03/2018.  
DATA DIVISION.  
    WORKING-STORAGE SECTION.  
    77 DATAS          PIC 9(06).  
    77 DIA            PIC 9(05).  
    77 HORA           PIC 9(08).  
    77 DiaSemana      PIC 9.  
PROCEDURE DIVISION.  
BEGIN.  
    ACCEPT DATAS FROM DATE.  
    DISPLAY DATAS  
    ACCEPT DIA FROM DAY.  
    DISPLAY DIA  
    ACCEPT HORA FROM TIME.  
    DISPLAY HORA  
    ACCEPT DiaSemana FROM DAY-OF-WEEK.  
    DISPLAY DiaSemana  
STOP RUN.
```

H.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. UFC0807-F2EX1-H.  
DATA DIVISION.  
    WORKING-STORAGE SECTION.  
    01 UserName      PIC X(20).  
*>definição das variáveis para armazenar a data: Formato YYMMDD  
01 CurrentDate.  
    02 CurrentYear    PIC 99.  
    02 CurrentMonth   PIC 99.  
    02 CurrentDay     PIC 99.  
*>definição da variável para armazenar o dia da semana: Formato D (1-Monday)  
    01 DayOfWeek      PIC 9.  
  
*>definição das variáveis para armazenar o dia do ano: Formato YYDDDD  
01 DayOfYear.  
    02 FILLER         PIC 99.  
    02 YearDay        PIC 9(3).  
  
*>definição das variáveis para armazenar as horas: Formato HHMMSSss com s=S/100  
01 CurrentTime.  
    02 CurrentHour    PIC 99.  
    02 CurrentMinute  PIC 99.  
    02 FILLER         PIC 9(4).  
  
*>definição das variáveis para armazenar a data: Formato YYYYMMDD  
01 Y2KDate.  
    02 Y2KYear        PIC 9(4).  
    02 Y2KMonth       PIC 99.  
    02 y2KDay         PIC 99.
```

*>definição das variáveis para armazenar o dia do ano: Formato YYYYDDD

01 Y2KDayOfYear.

02 Y2KYear2 PIC 9(4).

02 Y2KDay2 PIC 999.

PROCEDURE DIVISION.

DISPLAY "Insere o teu nome: " WITH NO ADVANCING

ACCEPT UserName

DISPLAY "*****"

ACCEPT CurrentDate FROM DATE

ACCEPT DayOfYear FROM DAY

ACCEPT DayOfWeek FROM DAY-OF-WEEK

ACCEPT CurrentTime FROM TIME

ACCEPT Y2KDate FROM DATE YYYYMMDD

ACCEPT Y2KDayOfYear FROM DAY YYYYDDD

DISPLAY " O Nome é "UserName

DISPLAY " a data é "CurrentDay" – "CurrentMonth" – "CurrentYear

DISPLAY " Hoje é " DayOfWeek

DISPLAY " Hoje é o dia "DayOfYear" do ano."

DISPLAY " São "CurrentHour" : "CurrentMinute

DISPLAY " Y2KData é "Y2kDay SPACE Y2kMonth SPACE Y2kYear

DISPLAY " Y2kDia2 do ano é "Y2KYear2 " do ano de "Y2KYear

2- Cria um item elementar com o nome ValorTaxa que aceite um valor entre 0 e 99999.99.

3- Cria um item elementar alfanumérico chamado NomeVideo, com 35 carateres. Define que NomeVideo é inicializado com espaços quando o programa inicia.

4. Um item chamado Minimo é definido com PIC 9V99. Mostra o que acontece ao dado após a execução do programa:

MOVE 123.5 TO Minimo

	Minimo
Antes	
Depois	

5. Considera o grupo de itens chamado Cliente:

01 Cliente.

02 Cliente Id PIC 9(5) VALUE ZEROS.

02 ClienteNome.

03 Iniciais PIC XX VALUE SPACES.

03 Sobrenome PIC X(4) VALUE SPACES.

02 Genero
02 Pagamento

PIC X
PIC 9(5)V99

VALUE SPACES.
VALUE ZEROS.

- a) Em baixo está disponível o diagrama do Cliente. Complete o diagrama mostrando como os dados subordinados são distribuídos pelos 19 caracteres correspondentes ao armazenamento de Cliente.

Cliente																		
Cliente Id					ClienteNome						Genero	Pagamento						
					Iniciais		Sobrenome											

- b) Na primeira linha do diagrama, mostre como os dados de cada item de Cliente são inicializados com a cláusula VALUE.

- c) Para cada declaração do programa apresentado a seguir, mostre o que acontece aos dados de Cliente. Utilize uma linha para cada declaração.

PROCEDURE DIVISION.

BEGIN.

MOVE "45145MCRyanF23445.67" TO Cliente

MOVE "Male" TO Genero

MOVE "GSPower" TO ClienteNome

MOVE "Fitzroy" TO Sobrenome

MOVE 34 TO Pagamento

STOP RUN.

6. Tendo em conta o exemplo da página 2 (sobre datas) e o exercício 1.H, desta ficha, define as alterações necessárias para se conseguir o seguinte output, para a variável HOJE (supondo que a data apresentada corresponde à do relógio do pc, obviamente).

2022/04/04