

Formação COBOL

Professor: Vagner Bellacosa

Disciplina: Linguagem de Programação COBOL

Formação: COBOL



1) Vagner Bellacosa seu facilitador

Analista programador desde 1989, onde comecei como auxiliar e tecnólogo em processamento de dados desbravando os primórdios da computação brasileira, um eterno aprendiz em processos Mainframe..

Desde então trabalhei em centenas de projetos, em 4 países e dezenas de instituições financeiras, ora como funcionário, consultor externo e freelancer.

A mais memorável foi o Banco REAL, uma verdadeira escola, onde aprendi muito e tive a oportunidade de participar no aliciante Projeto Y2K, o temível bug do milênio.

Particpei em projetos na CESP, Fundação CESP, Transbrasil, ABSA, Real Seguros, BPN, Skandia, DGITA, BES, CGD, BPI, Barclay, Skandia, IBM Italia, Sistemi Informativi, Unicredit, Zurich Assicurazione, Banco Safra e Banco Itaú.

Atualmente trabalho na Spread no Projeto BRB em Brasília, via remoto.

Formação: COBOL

Modulo 05.02 – Primeiros comandos

- 1) Elementos da Linguagem
- 2) Variáveis
- 3) Constantes
- 4) Regras de Pontuação
- 5) Hierarquia de Variáveis
- 6) Nível 66
- 7) Nível 77
- 8) Nível 88
- 9) Nível 88 – Complexo
- 10) Dúvidas



Formação: COBOL

Elementos da Linguagem

```

000011 *****
000012 ***          CARACTERES ACEITOS NA CODIFICACAO COBOL
000013 *****
000014
000015 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000016 SPACE
000017 : ; , .
000018 ( ) [ ] { }
000019 ' "
000020 + - * / =
000021 ? ! @ # $ % & ~
000022
000023     IMPORTANTE ALGUNS DESTES ELEMENTOS SAO EXCLUSIVO DE COMANDOS
000024     OUTROS SAO ELEMENTOS DE VISUALIZACAO GRAFICA
000025     PARA AS LETRAS MINUSCULAS USE O COMANDO TSO CAPS ON OFF
000026 *****
    
```

- 1) As 27 letras do alfabeto maiúsculas e minúscula
- 2) Espaço em branco
- 3) Dois pontos, Ponto e virgula, Virgula e Ponto
- 4) Aspas e apóstrofe
- 5) Parenteses, colchetes e chaves :
esquerdo e direito
- 6) Sinais matemáticos mais, menos, traço, asterisco, barra e igual
- 7) Sinais gráficos maior, menor, cifrão, &,
%, @, #, !, ?

COBOL

Elementos da Linguagem

```

EDIT      KC02746.WORKBOOK.COBOL(ELEMGRAF) - 01.04      Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 *****
000002 ***          EEEEE  BBBB  CCCCC  DDDD  IIIII  CCCCC  ***
000003 ***          EE      BB  BB  CC  CC  DD  DD  II  CC  CC  ***
000004 ***          EE      BB  BB  CC      DD  DD  II  CC      ***
000005 ***          EEEEE  BBBB  CC      DD  DD  II  CC      ***
000006 ***          EE      BB  BB  CC      DD  DD  II  CC      ***
000007 ***          EE      BB  BB  CC  CC  DD  DD  II  CC  CC  ***
000008 ***          EEEEE  BBBB  CCCCC  DDDD  IIIII  CCCCC  ***
000009 *****
000010 *** EBCDIC - ELEMENTOS DA LINGUAGEM TABELA DE CARACTERES ***
000011 *****
000012 ***          CARACTERES ACEITOS NA CODIFICACAO COBOL ***
000013 *****
    
```

COBOL

Elementos da Linguagem

```

000010 *** EBCDIC - ELEMENTOS DA LINGUAGEM TABELA DE CARACTERES ***
000011 ****
000012 *** CARACTERES ACEITOS NA CODIFICACAO COBOL ***
000013 ****
000014
000015 ABCDEFGHIJKLMNOPQRSTUVWXYZ
000016 SPACE
000017 : ; , .
000018 ( ) [ ] { }
000019 ' "
000020 + - * / =
000021 ? ! @ # $ % & ~
000022
000023     IMPORTANTE ALGUNS DESTES ELEMENTOS SAO EXCLUSIVO DE COMANDOS
000024     OUTROS SAO ELEMENTOS DE VISUALIZACAO GRAFICA
000025     PARA AS LETRAS MINUSCULAS USE O COMANDO TSO CAPS ON OFF
000026 ****
    
```

COBOL

Variáveis

Reforçando o conceito o COBOL aceita dois tipos de variáveis as voláteis e as fixas ou **CONSTANTES**.

As variáveis voláteis também são chamadas de literais.

Literal Numérica – recebe valores numéricos, sinal de mais, menos e ponto ou virgula tem um tamanho máximo de 18 bytes.

Literal alfanumérica – Tem um tamanho máximo de 120 caracteres e podem receber qualquer caractere.

Literal alfabética – Tem um tamanho máximo de 120 caracteres e podem receber apenas caracteres alfabéticos

COBOL

Constantes

São variáveis de valor fixo e estão enquadradas nas palavras reservadas do COBOL.

Não sofrem alterações no decorrer do processamento e podem ser movidas para variáveis, sua função é trazer um valor sabido e conhecido.

Na tabela ao lado apresento algumas utilizadas.

HIGH-VALUES

Atribui o maior valor na variável, preenchendo todos os campos com ele.

LOW-VALUES

Atribui o menor valor na variável, preenchendo todos os campos com ele.

SPACE ou SPACES

Preenche com espaços

ZERO ou ZEROES

Preenche com zeros

ALL

Transforma uma literal alfanumérica em constante figurativa.

COBOL

Regras de Pontuação

Um Statement Cobol pode terminar de duas maneiras distintas, a mais comum e através do ponto “.” alguns comandos específicos podem terminar tanto com o ponto como o comando END.

Entre comparações de variáveis deve haver um espaço.

Respeitando as áreas de codificação A e B

Entre comandos e variáveis, deve haver a separação por espaços.

Entre operando matemáticos e variáveis deve haver separação por espaços.

.

COBOL

Hierarquia das Variáveis

O COBOL tem um conceito bem interessante para a criação de variáveis, existe uma hierarquização dos campos que criam uma ordem de colocação na memória.

Podem ser campos livres ou associados há um registro, quando fomos falar sobre Arquivos, explicarei melhor o conceito de Registro.

O nível 01 é o primeiro nível podendo ser utilizado até o 55.

O restante dos níveis são reservados do COBOL, com a exceção do 77 e do 88

O nível 77 é de uso livre para variáveis sem estrutura e usadas no programa.

O nível 88 é utilizado para variáveis lógicas.

COBOL

Hierarquia das Variáveis

```

EDIT          KC02746.WORKBOOK.COBOL(VARNIVEL) - 01.07      Member VARNIVEL saved
Command ==> _____ Scroll ==> CSR
*****
000001 ***** Top of Data *****
000002 *** HIERARQUIA DE VARIÁVEIS - EXEMPLO HIPOTETICO ***
000003 *****
000004 01 ARQUIVO
000005     05 REG-CODIGO.
000006         10 REG-CODG-NUM          PIC 9(05).
000007         10 REG-CODG-VAL          PIC 9(01).
000008     05 REG-CLI-NOME              PIC X(30).
000009     05 REG-CLI-CPF.
000010         10 REG-CPF-NUM          PIC 9(09).
000011         10 REG-CPF-DIG          PIC 9(02).
000012     05 REG-DATA-NASC              PIC X(10).
000013     05 FILLER REDEFINES REG-DATA-NASC.
000014         10 REG-DTNSC-ANO        PIC 9(04).
000015         10 FILLER                PIC 9(01).
000016         10 REG-DTNSC-MES        PIC 9(02).
  
```

COBOL

Hierarquia das Variáveis

```
000017      10 FILLER                PIC 9(01).
000018      10 REG-DTNSC-DIA          PIC 9(02).
000019      05 REG-ENDERECO            PIC X(30).
000020      05 REG-NUMERO                PIC X(10).
000021      05 REG-CEP                  PIC X(08).
000022      05 FILLER REDEFINES REG-CEP.
000023      10 REG-CEP-NUM              PIC 9(05).
000024      10 FILLER                    PIC X(01).
000025      10 REG-CEP-COMPL            PIC 9(02).
000026
```

COBOL

Exemplo de nível

O COBOL tem um conceito bem interessante para a criação de variáveis, existe uma hierarquização dos campos que criam uma ordem de colocação na memória.

Podem ser campos livres ou associados há um registro, quando fomos falar sobre Arquivos, explicarei melhor o conceito de Registro.

O nível 01 é o primeiro nível podendo ser utilizado até o 55.

O restante dos níveis são reservados do COBOL, com a exceção do 66, 77 e do 88

O nível 66 é usado para renomear uma área de memória.

O nível 77 é de uso livre para variáveis sem estrutura e usadas no programa.

O nível 88 é utilizado para variáveis lógicas.

COBOL

Razão de uso do nível 66

Em algumas situações a lógica de nosso programa necessita utilizar um grupo de variáveis com uma nova nomenclatura.

Nessas situações criamos um nível 66, criamos uma nova variável que utilizará o mesmo espaço de uma variável ou grupo, após o RENAMEs.

Útil por exemplo no formato de CEPs, Data, Endereços e outros detalhes subdivididos em variáveis

```
WORKING-STORAGE SECTION.  
01 WS-A.  
    05 WS-DT1      PIC X(02) VALUE 'AA'.  
    05 WS-DT2      PIC X(02) VALUE 'BB'.  
    05 WS-DT3      PIC X(02) VALUE 'CC'.  
    05 WS-DT4      PIC X(02) VALUE 'DD'.  
    05 WS-DT5      PIC X(02) VALUE 'EE'.  
66 WS-B           RENAMES WS-DT1 THRU WS-DT3.
```

COBOL

Razão de uso do nível 77

Eles foram muito úteis em versões anteriores do COBOL, onde havia uma diferença no armazenamento e processamento dos itens de nível 77 e 01. Mas com a versão mais recente do COBOL, os campos declarados 01 e 77 estão funcionando de maneira semelhante, portanto, 77 itens não têm muito significado, embora estejam em uso.

77 O número do nível era fundamental nos dias anteriores, pois a velocidade de execução do código do objeto era um fator importante. O tempo da CPU também era um grande problema nos tempos antigos.

Eles costumavam atribuir registros de acesso mais rápido às variáveis usadas com mais frequência e os 77 números de nível são projetados para atribuí-lo a registros de acesso mais rápido. Portanto, as variáveis usadas com frequência costumavam ser declaradas com 77 números de nível.

COBOL

Exemplo de nível 77

```

EDIT      KC02746.WORKBOOK.COBOL(VARNIVEL) - 01.07      Columns 00001 00072
Command ==> _____ Scroll ==> PAGE
000027 *
000028 * NIVEL  77 & 88 NAO POSSUEM NIVEIS.
000029 *
000030
000031 * VARIAVEIS AUXILIARES  NUMERICAS
000032 77  WSS-DATA-DIA          PIC 9(08).
000033 77  WSS-LINHAS           PIC 9(02)  VALUE 24.
000034 77  WSS-COLUNAS         PIC 9(02)  VALUE 80.
000035 77  WSS-PAGINA          PIC 9(02)  VALUE 0.
000036 *
000037 * VARIAVEIS AUXILIARES  ALFANUMERICAS
000038 77  WSS-CEP              PIC X(09).
000039 *
000040 * VARIAVEIS AUXILIARES  ALFABETICAS
000041 77  WSS-SOBRENOME        PIC A(20).
  
```

A declaração deve começar na Área A

Não são permitidos subitens.

Não deve fazer parte de nenhum outro item.

Pode ser usado para todos os tipos de itens de dados.

COBOL

Exemplo de nível 88

```

VIEW          KC02746.WORKBOOK.COBOLO12) - 01.03          Columns 00001 00066
Command ==>          Scroll ==> CSR
001900 001900*****
002000 002000
002100 002100*****
002200 002200 DATA DIVISION.
002300 002300*****
002400 002400 FILE SECTION.
002500 002500*
002600 002600 WORKING-STORAGE SECTION.
002700 002700*
002800 002800 01  WSS-SEXO                                PIC X(01).
002900 002900      88 LOG-MASCULINO VALUE 'M'.
003000 003000      88 LOG-FEMININO VALUE 'F'.
003100 003100*
003200 003200*****
003300 003300 PROCEDURE DIVISION.
003400 003400*****
003500 003500
003600 003600
003700 003700      DISPLAY 'COBOLO12'.
*DSLIST
  
```

COBOL

Exemplo de nível 88

```

000062 *  VARIÁVEIS LÓGICAS CONDICIONAIS COMPLEXAS
000063 01  WSS-ESTADOS          PIC X(02).
000064      88 SUDESTE          VALUE 'SP', 'RJ', 'ES', 'MG'.
000065      88 SUL               VALUE 'PR', 'RS', 'SC'.
000066      88 NORTE              VALUE 'AC', 'AP', 'AM', 'PA', 'RO', 'RR'.
000067      88 NORDESTE          VALUE 'AL', 'BA', 'CE', 'MA', 'PB', 'PE',
000068                               'PI', 'RN', 'SE'.
000069      88 CENTRO-OESTE       VALUE 'DF', 'GO', 'TO', 'MT', 'MS'.
000070
000071 *  EXEMPLO DE USO
000072
000073      EVALUATE TRUE
000074          WHEN NORTE
000075              PERFORM 2100-TRATAR-NORTE
000076          WHEN NORDESTE
000077              PERFORM 2200-TRATAR-NORDESTE
000078          WHEN CENTRO-OESTE
    
```

COBOL

Exemplo de nível 88

```
000077      PERFORM 2300-TRATAR-CENTRO-OESTE
000078      WHEN SUDESTE
000079      PERFORM 2400-TRATAR-SUDESTE
000080      WHEN SUL
000081      PERFORM 2500-TRATAR-SUL
000082      WHEN OTHER
000083      PERFORM 9100-TRATAR-ERRO
000084      END-EVALUATE.
000085
```

COBOL

Exemplo de nível 88 - complexo

```

000086 *****
000087 *** TRATAMENTOS ESPECIAIS EM VARIAVEIS CONDICIONAIS ***
000088 *****
000089
000090 * PARAMETRO : VALUES
000091 * PERMITE INICIALIZAR UMA VARIAVEL COM UM VALOR OU RANGE DE VALORES
000092 01 WSS-NUM-PEDIDO PIC 9(02)
000093 88 WSS-RANGE-10 VALUE 1 THRU 10.
000094 88 WSS-RANGE-20 VALUE 11 THRU 20.
000095 88 WSS-RANGE-30 VALUE 21 THRU 30.
000096 *
000097 01 WSS-TIPO-CARTAO PIC X(01) VALUES SPACES.
000098 88 WSS-CARTAO-DEBITO VALUE 'D'.
000099 88 WSS-CARTAO-CREDITO VALUE 'C'.
000100 88 WSS-CARTAO-POUPANCA VALUE 'P'.
000101

```

COBOL

Exemplo de nível 88 - complexo

```
000102 * EXEMPLO DE USO
000103
000104     IF WSS-CONTA(01:04) = '1000'
000105         SET WSS-CARTAO-DEBITO      TO TRUE
000106     ELSE
000107         IF WSS-CONTA(01:04) = '2000'
000108             SET WSS-CARTAO-CREDITO  TO TRUE
000109         ELSE
000110             SET WSS-CARTAO-POUPANCA  TO TRUE
000111         END-IF
000112     END-IF.
000113
000114     IF WSS-CARTAO-DEBITO
000115         PERFORM 2500-TRATAR-CONTA-DEBITO
000116     END-IF.
000117
000118 ***** FIM DO EXEMPLO *****
```

COBOL

Duvidas????

