

UNIVERSIDADE FEDERAL DA BAHIA
INSTITUTO DE MATEMÁTICA
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA: LINGUAGENS PARA APLICAÇÃO COMERCIAL

A linguagem COBOL

Salvador-Ba
Junho – 2009

Luís Bernardo Souza Braga

Mini-Seminário – A linguagem COBOL

Atividade apresentada como requisito
para avaliação no curso de Ciência da
Computação da Universidade Federal da
Bahia na disciplina Linguagens Para
Aplicação Comercial orientada pelo
professor Adonai Estrela Medrado.

Salvador-Ba
Junho – 2009

Sumário

1. INTRODUÇÃO.....	4
2. CARACTERÍSTICAS.....	4
3. ESTRUTURA BÁSICA.....	4
4. LICENÇA.....	5
5. CONCLUSÕES.....	5
REFERÊNCIAS:.....	6
APÊNDICES.....	7
APÊNDICE A – CÓDIGO DE EXEMPLO DE UM PROGRAMA COBOL QUE CADASTRA E CONSULTA AS NOTAS DE ALUNOS EM UM ARQUIVO.....	7

1. Introdução

COBOL é uma linguagem de alto nível desenvolvida pela CODASYL Committee (**Conference on Data Systems Languages**) no início do ano de 1960. Seu nome é um acrônimo para Common Business Oriented Language, percebe-se daí o principal objetivo de sua criação, prover uma linguagem voltada para a resolução de problemas de natureza comercial e administrativa.

Embora a linguagem COBOL seja antiga, ela ainda é muito usada atualmente. Em uma pesquisa promovida pelo grupo Gartner[4] em 1999, “80% das aplicações corporativas em execução no mundo naquele momento eram escritas em COBOL”. Além disso, em um estudo mais recente, em 2006, o grupo estimou que existissem “180 bilhões de linhas de código em COBOL em uso ao redor do mundo” [6].

2. Características

Algumas características marcantes de uma aplicação escrita em COBOL são:

- Possui um código fonte bem próximo do inglês;
- Tende a ser mais extenso do que um mesmo programa escrito em outra linguagem;
- Alto custo financeiro, já que a manutenção e produção do código em COBOL é caro, devido ao fato destas aplicações serem críticas para muitas empresas.
- Boa manutenabilidade, já que o código produzido é bem próximo do inglês, tornando-se assim auto documentável.

3. Estrutura básica

Um programa básico em COBOL possui uma estrutura dividida em quatro divisões que devem surgir no programa na seqüência abaixo:

1. Identification Division: Região composta por informações a respeito do autor do programa, nome do programa, data de compilação, etc;
2. Environment Division: Nesta região fica a descrição do ambiente no qual o sistema rodará, tais como: equipamentos que o programa fará uso e etc;
3. Data Division: Esta região descreve os dados que farão parte do programa;
4. Procedure Division: Nesta região é feita a implementação do algoritmo para a resolução dos problemas. É a região equivalente a função main de um programa em C.

A depender do compilador é possível a construção de um programa em COBOL com apenas a Identification Division e a Procedure Division.

4. Licença

A especificação da linguagem não pertence a qualquer particular empresa. Um comitê chamado ANSI (American National Standards Institute) é o responsável por fazer revisões nas especificações sintáticas e semânticas para as versões do COBOL. Já foram feitas algumas revisões na linguagem em 1968, 1974, 1985 e 2002. Estas revisões tiveram objetivo trazer atualizações na linguagem tais como: o paradigma de orientação a objetos (2002), paradigma procedural(1985), etc. E desta maneira, viabilizar o uso da linguagem até os dias atuais.

Quanto a portabilidade de um programa COBOL, devido ao seu extensivo uso e tempo de mercado, ele foi portado para inúmeras plataformas – para todas as versões do Windows, todas as UNIX-LIKE, para AS/400, VSE, OS/2, DOS, VMS, Unisys, DG, VM, e MVS[1].

5. Conclusões

A linguagem COBOL foi por muito tempo uma linguagem popular para a construção de aplicações comerciais. O seu uso nos dias atuais, embora não pareça, possui ainda grande relevância. Percebemos isto na quantidade de código ainda presente nesta linguagem. Código que talvez por muito tempo seja usado, já que muitos deles dificilmente sejam alterados, pois estes fazem parte do controle financeiro, administrativo ou de importantes regras de negócio de muitas empresas.

Referências:

- [1] COBOL Tutorial. <http://www.csis.ul.ie/cobol/Course/COBOLIntro.htm>. Acessado em: 30 de maio de 2009.
- [2] Li, Jiehong. Abraham, Rona. COBOL. www.csee.umbc.edu/courses/graduate/631/Fall2002/COBOL.pdf
- [3] Wikipédia. <http://pt.wikipedia.org/wiki/COBOL>. Acessado em: 1 de junho de 2009.
- [4] Alguns fatos bem interessantes (quase chocantes) sobre COBOL. <http://www.itexto.net/devkico/?p=135>. Acessado em: 31 de maio de 2009.
- [5] Cobol: The new Latin. <http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9004821>. Acessado em: 31 de maio de 2009.
- [6] Brown, Gary DeWard - COBOL: The failure that wasn't - COBOLReport.com
- [7] Ono, Mutsuo - COBOL para Microcomputadores. McGraw-Hill.

Apêndices

Apêndice A – Código de exemplo de um programa COBOL que cadastrá e consulta as notas de alunos em um arquivo.

```
IDENTIFICATION DIVISION.  
PROGRAM-ID. Aluno.  
AUTHOR. Luis Braga.  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
SPECIAL-NAMES.  
    DECIMAL-POINT IS COMMA.  
INPUT-OUTPUT SECTION.  
FILE-CONTROL.  
    SELECT SAVEALUNO  
        ASSIGN TO "SaveAluno.txt".  
DATA DIVISION.  
FILE SECTION.  
FD SAVEALUNO.  
01 REG-SAVEALUNO.  
    02 FD-NOME PIC X(30).  
    02 FD-NOTA1 PIC 99V99.  
    02 FD-NOTA2 PIC 99V99.  
    02 FD-MEDIA PIC 99V99.  
  
WORKING-STORAGE SECTION.  
  
01 OPCAO-VALOR.  
    02 OPCAO PIC 9 VALUE ZERO.  
  
01 CADASTRO-ALUNO.  
    02 CONT-CADASTRO PIC X VALUE SPACE.  
        88 CONTINUA VALUE "S" "N" "s" "n".  
01 CONSULTA-ALUNO.  
    02 CONT-CONSULTA PIC X VALUE SPACE.  
        88 CONTINUA-SAIDA VALUE "S" "N" "s" "n".  
77 SAIDA PIC X VALUE SPACE.  
  
SCREEN SECTION.  
01 SELOPC.  
    02 BLANK SCREEN.  
    02 LINE 04 COLUMN 21 VALUE  
        "CONTROLE DE NOTAS DOS ALUNOS".  
    02 LINE 06 COLUMN 27 VALUE  
        "AUTOR: Luis Braga".  
    02 LINE 09 COLUMN 21 VALUE  
        "TELÀ PRINCIPAL".  
    02 LINE 11 COLUMN 21 VALUE  
        "[1] - INCLUSAO".  
    02 LINE 12 COLUMN 21 VALUE  
        "[2] - CONSULTA".  
    02 LINE 13 COLUMN 21 VALUE  
        "[3] - FIM".  
    02 LINE 17 COLUMN 21 VALUE "OPCAO ->".  
  
01 CADASTRO.  
    02 BLANK SCREEN.
```

```

02 LINE 04 COLUMN 21 VALUE
"CONTROLE DE NOTAS DOS ALUNOS".
02 LINE 06 COLUMN 27 VALUE
"AUTOR: Luis Braga".
02 LINE 09 COLUMN 21 VALUE
"CADASTRO DOS ALUNOS".
02 LINE 11 COLUMN 21 VALUE
"NOME : ".
02 LINE 12 COLUMN 21 VALUE
"NOTA1 : ".
02 LINE 13 COLUMN 21 VALUE
"NOTA2 : ".
02 LINE 14 COLUMN 21 VALUE
"MEDIA : ".
02 LINE 17 COLUMN 21 VALUE
"DESEJA CONTINUAR(S/N) : ".

01 CONSULTA-INTRO.
02 BLANK SCREEN.
02 LINE 04 COLUMN 21 VALUE
"CONTROLE DE NOTAS DOS ALUNOS".
02 LINE 06 COLUMN 27 VALUE
"AUTOR: Luis Braga".
02 LINE 09 COLUMN 21 VALUE
"CONSULTA DOS ALUNOS".
02 LINE 11 COLUMN 21 VALUE
"NOME : ".
02 LINE 12 COLUMN 21 VALUE
"NOTA1 : ".
02 LINE 13 COLUMN 21 VALUE
"NOTA2 : ".
02 LINE 14 COLUMN 21 VALUE
"MEDIA : ".
02 LINE 17 COLUMN 21 VALUE
"DESEJA CONTINUAR(S/N) : ".

01 ENTRADA-OPCAO.
02 EN-OP LINE 17 COLUMN 31 PIC Z USING OPCAO.
01 ENTRADA-ALUNO1.
02 EN-A LINE 11 COLUMN 31 PIC X(30) USING FD-NOME.
01 ENTRADA-ALUNO2.
02 EN-B LINE 12 COLUMN 31 PIC ZZ,ZZ USING FD-NOTA1.
01 ENTRADA-ALUNO3.
02 EN-C LINE 13 COLUMN 31 PIC ZZ,ZZ USING FD-NOTA2.
01 ENTRADA-ALUNO4.
02 EN-C LINE 14 COLUMN 31 PIC ZZ,ZZ USING FD-MEDIA.

01 CONSULTANDO-CONTINUA.
02 ENC-D LINE 17 COLUMN 47 PIC X USING CONT-CONSULTA.
01 CADASTRO-CONTINUA.
02 ENC-D LINE 17 COLUMN 47 PIC X USING CONT-CADASTRO.
01 EXIBE-MEDIA.
02 EXI-A LINE 14 COLUMN 31 PIC ZZ,ZZ USING FD-MEDIA.

01 ERROOPCAO.
02 LINE 21 COLUMN 21 VALUE "OPCAO INVALIDA".

01 EXIBE-FIM.
02 LINE 21 COLUMN 21 VALUE "FIM PROGRAMA".

01 LIMPA-MENSAGEM.

```

02 LINE 21 COLUMN 21 BLANK LINE.

PROCEDURE DIVISION.

TELAINICIAL.

DISPLAY SELOPC.

LER-OPCAO.

ACCEPT ENTRADA-OPCAO.

DISPLAY LIMPA-MENSAGEM.

IF (OPCAO = 1) OR (OPCAO = 2) OR (OPCAO = 3)

GO TO SELECAO

ELSE

DISPLAY ERROOPCAO

GO TO LER-OPCAO.

SELECAO.

IF OPCAO = 1

GO TO INCLUSAO.

IF OPCAO = 2

GO TO CONSULTA.

IF OPCAO = 3

GO TO FIM.

INCLUSAO.

DISPLAY CADASTRO.

OPEN OUTPUT SAVEALUNO.

LER-CADASTRO.

ACCEPT ENTRADA-ALUNO1.

ACCEPT ENTRADA-ALUNO2.

ACCEPT ENTRADA-ALUNO3.

COMPUTE FD-MEDIA = (FD-NOTA1 + FD-NOTA2) / 2 .

DISPLAY EXIBE-MEDIA.

WRITE REG-SAVEALUNO.

LER-CONTINUA.

ACCEPT CADASTRO-CONTINUA.

IF CONTINUA NEXT SENTENCE

ELSE

GO TO LER-CONTINUA.

IF (CONT-CADASTRO = "S") OR (CONT-CADASTRO = "s")

GO TO LER-CADASTRO

ELSE

CLOSE SAVEALUNO.

GO TO TELAINICIAL.

CONSULTA.

OPEN INPUT SAVEALUNO.

LER-CONSULTA.

DISPLAY CONSULTA-INTRO.

READ SAVEALUNO INTO REG-SAVEALUNO AT END GO TO FECHAMENTO.

DISPLAY ENTRADA-ALUNO1.

DISPLAY ENTRADA-ALUNO2.

DISPLAY ENTRADA-ALUNO3.

DISPLAY ENTRADA-ALUNO4.

LER-CONTINUA-CONSULTA.

ACCEPT CONSULTANDO-CONTINUA.

IF CONTINUA-SAIDA NEXT SENTENCE

ELSE

GO TO LER-CONTINUA-CONSULTA.

IF (CONT-CONSULTA = "S") OR (CONT-CONSULTA = "s")

GO TO LER-CONSULTA

ELSE

CLOSE SAVEALUNO.

GO TO TELAINICIAL.

FECHAMENTO.

CLOSE SAVEALUNO.

GO TO TELAINICIAL.

FIM.

DISPLAY EXIBE-FIM.

ACCEPT SAIDA.

STOP RUN.