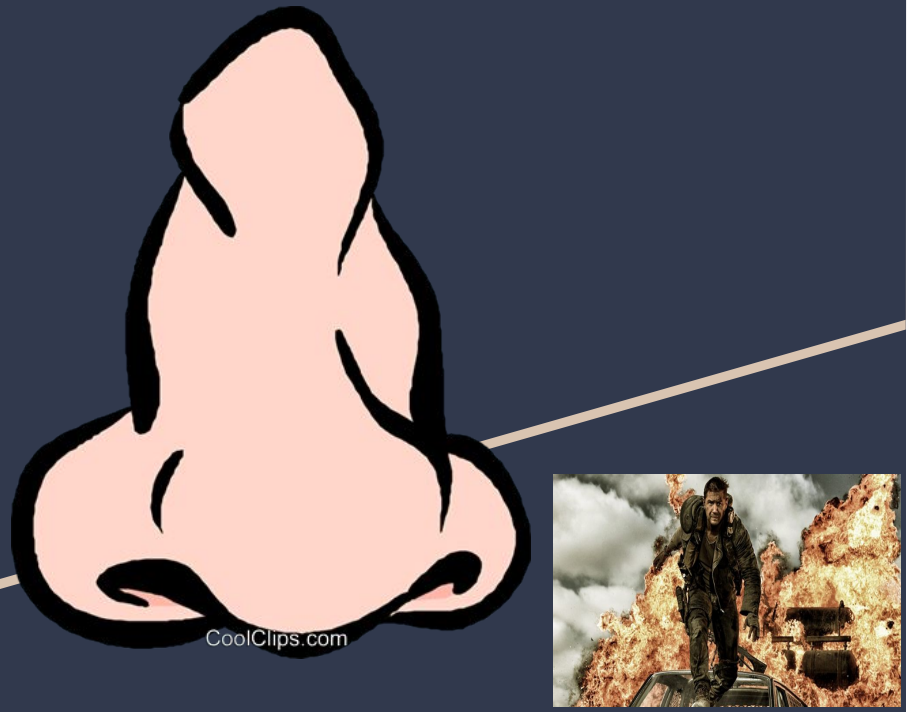
COBOL

Jhemeson Silva Mota
Lucas Alves Bastos

Nascimento

- COmmon Business Oriented Language
- Criada por um comitê de investigadores

Inspirações



- 1959
- FLOW-MATIC
  - boa parte de suas especificações foram inspiradas na linguagem de programação
  - Considerada a mãe do COBOL
- IBM COMTRAN
  - Também utilizada no processo

Paradigma - COBOL RAIZ

- Imperativo
  - Com ações, enunciados, variáveis, comandos
- Estruturado
  - Três estruturas
    - Sequência
    - Decisão
    - Iteração
  - “Sucedida” pela orientação a objetos

Paradigma - COBOL NUTELLA

- COBOL 2002
  - Orientado a Objetos
    - Tem estruturas de classes etc.

Legibilidade

- Linguagem com poucos componentes básicos
  - Poucos Comandos
  - Comandos em inglês simples

- Não têm diversas formas de se executar uma mesma ação
  - Devido à pouca quantidade de comandos, a quantidade de meios para se executar uma determinada ação é, consequentemente, menor.

Simplicidade Global

Google

- O COBOL é uma linguagem com poucos componentes básicos, e estes são, em sua absoluta maioria, ordens imperativas representadas por uma palavra em inglês simples. Além disso, devido à pouca quantidade de comandos, a quantidade de meios para se executar uma determinada ação é, consequentemente, menor.

Ortogonalidade

COBOL

≠

- COmmon Business Oriented Language não usa estruturas aninhadas, portanto, não é uma linguagem de programação ortogonal. Por sinal, esta é uma das grandes diferenças entre o COBOL e o ADA.

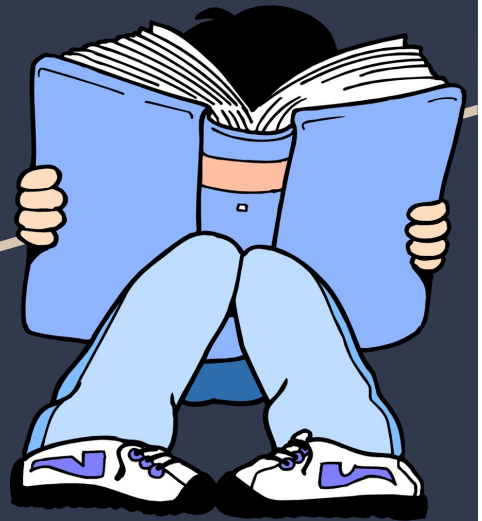
Instruções de Controle

- O COBOL é procedural e utiliza estruturação em blocos. Além disso, é sequencial, portanto, legível de cima a baixo. O COBOL utiliza GOTO, portanto, tal fator pode fazer com que o usuário precise saltar ou reler certas partes de código durante sua leitura.

Tipos e Estruturas de Dados

- No COBOL há apenas três tipos de dados, sendo eles:
  - numérico
  - alfanumérico
  - constante figurativa.
- Não há vetor em COBOL, porém, existe o “WS-TABLE” que pode servir para simular um vetor.

Sintaxe



- Cada Programa tem 4 Divisões:
  - Identification Division
  - Environment Division
  - Data Division
  - Procedure Division

Capacidade de Escrita

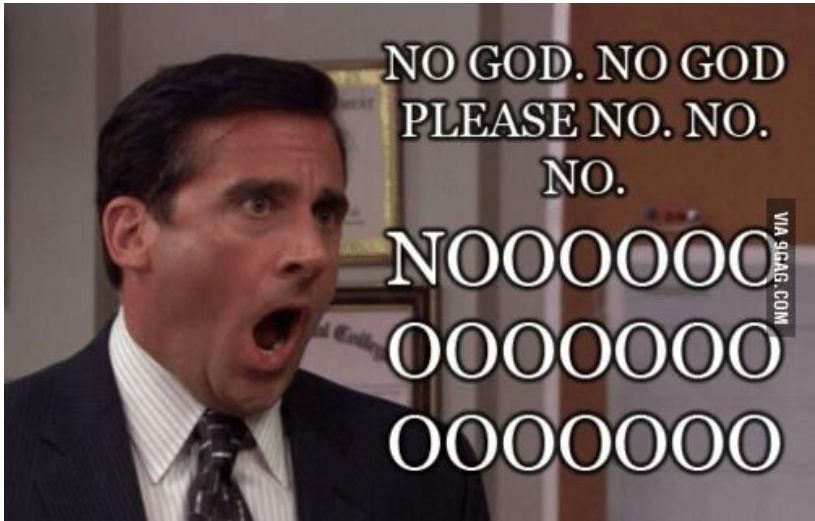
- Mesmo depois de ler a documentação e entender a estrutura, pode ser difícil escrever o código pois são muitos detalhes a serem lembrados.

Simplicidade e Ortogonalidade

- A **leitura** da linguagem não é complexa, e é possível saber o que o programa está fazendo sem ler uma documentação, ou ter um conhecimento prévio da linguagem, tendo apenas um nível razoável de inglês e lógica de programação.

Suporte e Abstração

- Não da suporte à abstração



Expressividade

- Considerando que expressividade é um conjunto relativamente conveniente de maneiras de especificar operadores, COBOL não é expressivo pois a forma de escrever é rígida.

Confiabilidade

Verificação de Tipos

- O COBOL faz a verificação dos tipos das variáveis envolvidas nas atribuições de valores em tempo de compilação.

Manipulação de Exceções

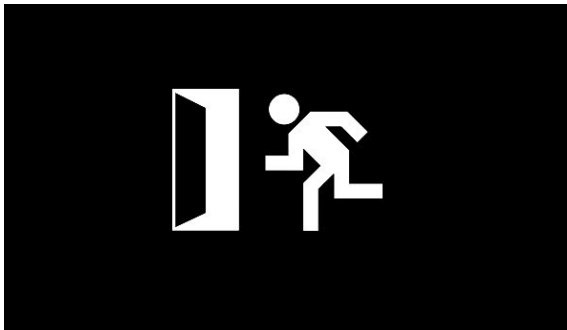
- Em COBOL não existem comandos para tratamento de exceção, porém, algumas vezes, podemos simular o tratamento através de um comando condicional.

Apelidos

- Aliases não são utilizados em COBOL

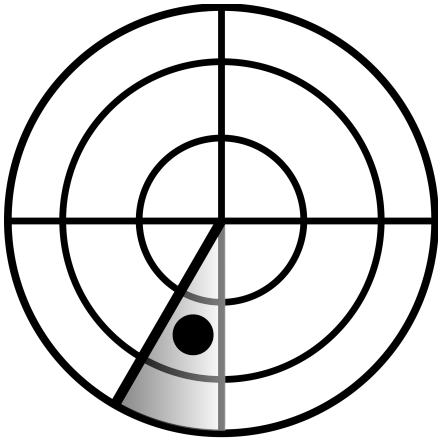
Portabilidade

- Roda em praticamente todo os sistemas operacionais
- Possui a capacidade de importar bibliotecas de outras linguagens.



Detecção de Erros

- Há dois tipos de erro de tempo de execução;
  - Exceções
    - As exceções são erros fatais que não são relatados e, portanto, não podem ser capturados.



Detecção de Erros

- Erros de E / S.
  - Os erros de E / S são fatais ou recuperáveis. Os erros recuperáveis são relatados pelo sistema operacional; Isso permite que você os aprisione e tome medidas para se recuperar, se possível.

Tipos de Dados

- Numéricos
- Alfanuméricos
- Constantes Figurativas

Declaração de Variáveis

- Padrão
  - 01 NOME PIC VALUE ‘0’
- ‘01’ define o nível
  - Se >= 2 e <= 49 == item elementar
- “PIC” descreve o tipo do dado
  - Se **PIC XXX** ou **PIC X(3)**
    - Declara um alfanumérico de 3 posições
  - Se **PIC 999** ou **PIC 9(3)**
    - Declara um numérico de 3 posições

Receita Federal

Declaração de Variáveis

- Se **PIC 9(n)**
  - Declara um numérico com posições dinâmicas (<18)
- Se **PIC 999V99**
  - Declara um numérico com ponto depois da terceira casa

Declaração de Variáveis

```
01 WS-NUM1 PIC 99V9 VALUE IS 3.5.
01 WS-NAME PIC A(6) VALUE 'ABCD'.
01 WS-ID PIC 99 VALUE ZERO.
```

Praticando...

Definição de variáveis e operações básicas

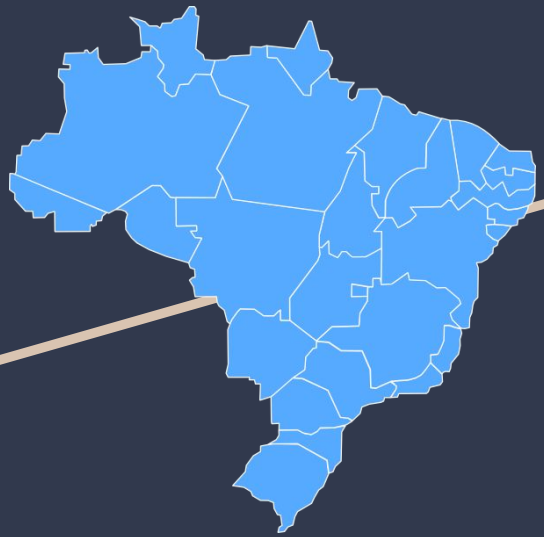
Constantes Figurativas

- Zero, Zeros ou Zeroes: será entendido pelo compilador como um valor zerado. Seu formato será escolhido de acordo com a necessidade.
- Space ou Spaces: espaço em branco
- High-Value ou High-Values: campo possui todos os bits ligados. Nada é maior que um HIGH-VALUE

Constantes Figurativas

- Low-Value ou Low-Values: é Menor valor possível para um campo. Nenhum campo tem valor menor que o Low-Value.
- Quote ou Quotes: usado quando se precisa imprimir o apóstrofo. Quote substitui o mesmo (na delimitação de strings) por outro caractere delimitador

Mapeamento Finito



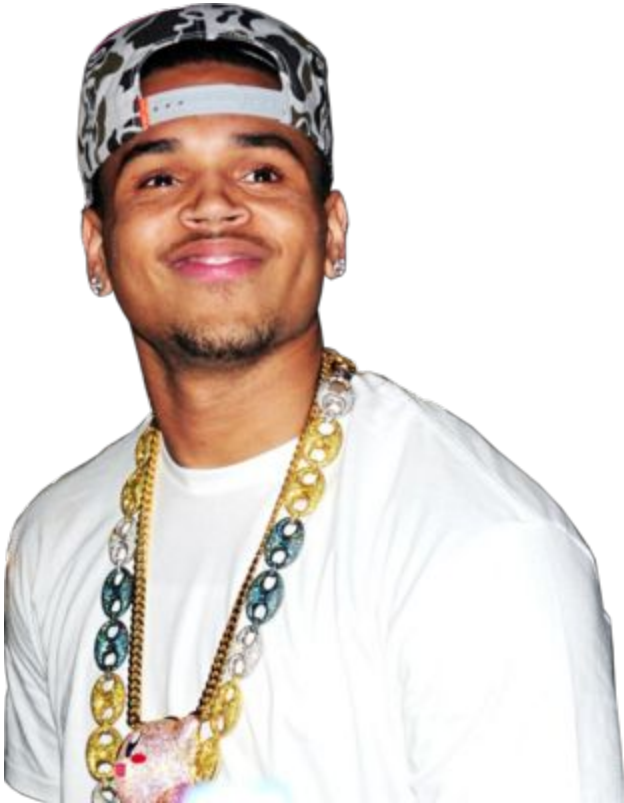
- Em COBOL todos os elementos tem sua quantidade limite definida (até o ‘n’)
- OCCURS indica a repetição do nome do dado
- WS-NOMEARRAY OCCURS 10 TIMES
  - Temos um “array” com o nome “NOMEARRAY” de 10 posições

Produto Cartesiano

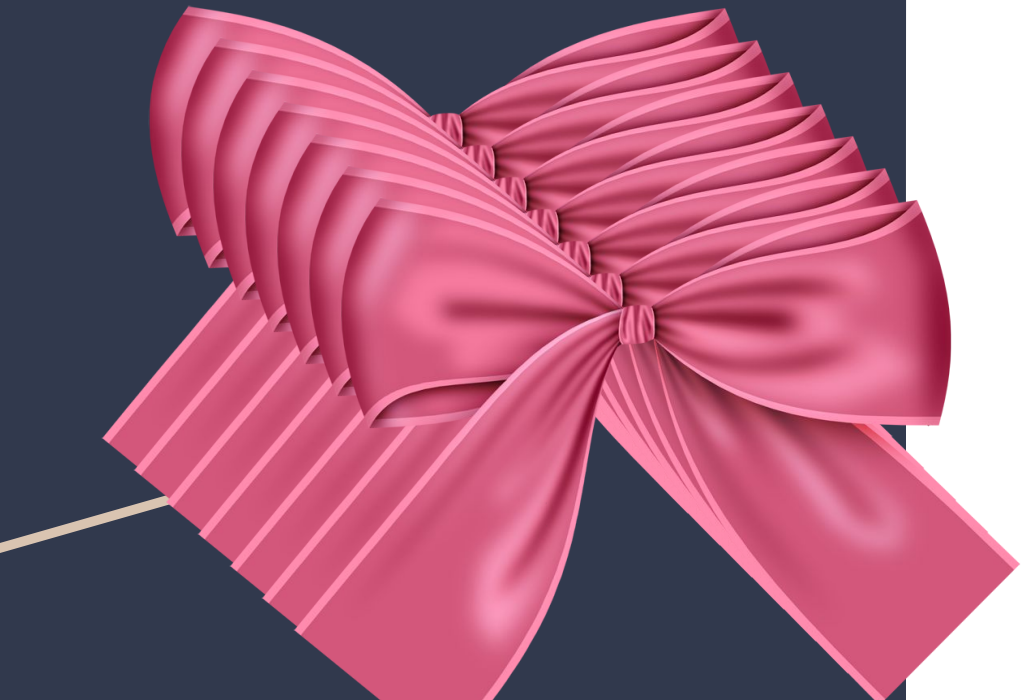
```
01 ws-data.
   03 ws-dia                pic 9(002) value 11.
   03 filler                pic x(001) value "/".
   03 ws-mes                pic 9(002) value 02.
   03 filler                pic x(001) value "/".
   03 ws-ano                pic 9(004) value 2015.
```

Estruturas Condicionais

IF E ELSE.



Laço de Repetição



- perform until:

```
move 5 to n
move 0 to i
move 1 to fact
perform until i greater than n
    move i to ist
    move fact to factst
    display ist "! = " factst
    add 1 to i
    multiply i by fact
```

- go to:

```
GOTO-A.
    DISPLAY "GO TO A"
    GO TO GOTO-B.
GOTO-B.
    DISPLAY "GO TO B"
    GO TO GOTO-C.
GOTO-C.
    DISPLAY "GO TO C"
    GO TO FIM.
FIM.
    DISPLAY "FIM DO GO TO"
STOP RUN.
```

Praticando...

Fibonacci e Fatorial

COBOL

Jhemeson Silva Mota
Lucas Alves Bastos