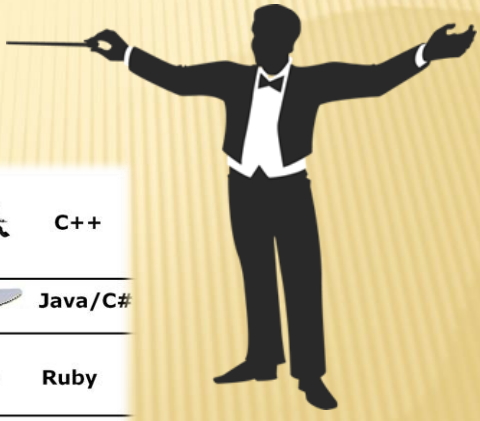
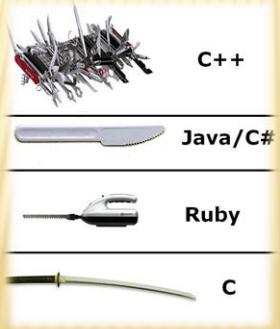
SUMÁRIO - AULA 18

- Avaliação de Linguagens de Programação





MOTIVAÇÃO

- Por que avaliar?

  - Medir algo visando ter parâmetros para comparação...

MOTIVAÇÃO

- Por que avaliar?
  - Hilux
  - S-10
  - Frontier
- Quais os critérios?
  - segurança
  - custo
  - conforto
  - consumo
  - beleza
  - revenda
  - etc.

MOTIVAÇÃO

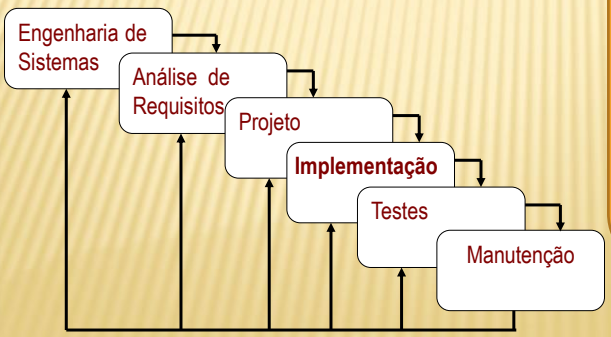
- Quais os critérios?
  - passageiros, espaço, poluição, custo, etc.

MOTIVAÇÃO

- **Crise de Software** (final dos anos 60):
  - Demanda muito superior à capacidade de desenvolvimento;
  - Qualidade insuficiente dos produtos;
  - Estimativas de custo e tempo raramente cumpridas nos projetos.

MOTIVAÇÃO

- Por volta de 1970 criou-se o conceito de ciclo de vida do software.
  - Winston W. Royce
- Modelo Clássico (cascata):



Implementação

Tradução das representações do projeto para uma linguagem artificial resultando em instruções executáveis pelo computador.

MOTIVAÇÃO

- Avaliar as características das diversas LPs observando-se os impactos sobre o processo de desenvolvimento de software.

CRITÉRIOS PARA AVALIAR LPS

- Precisamos de critérios:
  - **Legibilidade** (*Readability*)
  - **Escritabilidade** (*Writability*)
  - **Confiabilidade** (*Reliability*)
  - **Custos**

CRITÉRIOS PARA AVALIAR LPS

- **Legibilidade** (*Readability*)
  - Quão fácil **ler** e **entender** um programa
    - Que tal Assembly?

NEM
tUdO QuE é
legível é
nEcESSArIaMeNTe
FÁCIL DE LER!

- **Escritabilidade** (*Writability*)
  - Quão fácil usar uma linguagem para **criar** programas.
    - Que tal o Lisp/Prolog?

CRITÉRIOS PARA AVALIAR LPS

- **Confiabilidade** (*Reliability*)
  - Conformidade com as especificações de acordo as condições impostas.
    - Faz o que diz fazer?

- **Custos**
  - Principais elementos na avaliação de qualquer linguagem de programação.

CRITÉRIOS PARA AVALIAR LPS

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade e Escritabilidade
  - Uma linguagem que não suporta **maneiras naturais** (ex: língua escrita) de expressar os **algoritmos** usará, necessariamente, abordagens **não naturais** e assim, a legibilidade será **reduzida**.
    - A **legibilidade** afeta a **confiabilidade** tanto na **escrita** quanto na **manutenção**.

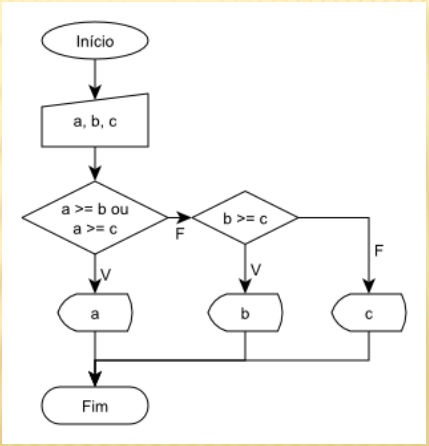
```
//Binary microcode:
//J command_I command_vector command
//2 last bits for loop command

//Line 0:
000_000_11__000_001_11__00
//Line 1:
000_010_00__000_000_00__00
//Line 2:
101_001_00__010_001_00__01
//Line 3:
000_001_01__000_000_00__01
//Line 4:
000_001_01__000_000_00__00
//Line 5:
001_100_01__001_000_10__11
//Line 6:
000_000_00__101_000_00__11
//Line 7: halt program
111_111_11__111_111_11__11
```

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade e Escritabilidade
  - Quanto mais fácil de escrever um programa:
    - maior a probabilidade dele estar certo
    - maior a facilidade de corrigi-lo e de (re-) escrevê-lo

AVALIE A LEGIBILIDADE E ESCRITABILIDADE

- O que faz o seguinte algoritmo?



CRITÉRIOS PARA AVALIAR LPS

- Legibilidade
  - Simplicidade Global
  - Ortogonalidade
  - Declarações de Controle
  - Instruções de Controle
  - Tipos e Estruturas de Dados
  - Considerações sobre a Sintaxe

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade
  - Simplicidade Global

    - Linguagens com um grande número de componentes básicos são mais difíceis de aprender do que aquelas com um número reduzido.
      - Ex: Racket...inúmeras funções.
      - Exige mais esforço no aprendizado...

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade
  - Simplicidade Global
    - Multiplicidade de Recursos
      - mais de uma maneira para realizar uma operação.

cont = cont + 1;
cont += 1;
cont++;
++cont;

do
while
for
foreach

```
boolean a;

// operador ternário
a = 10 > 5 ? true : false;

// if tradicional
if (10 > 5) {
    a = true;
} else {
    a = false;
}
```

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade
  - Ortogonalidade
    - Um conjunto de construções pode ser combinado para formar um outro conjunto.
    - Possibilidade de combinar entre si, sem restrições, os componentes básicos da LP.
    - Influenciada pelo modelo de LP:
      - Modelo de Objetos: **objeto**
      - Modelo funcional: **funções**

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade
  - Declarações de Controle
    - Um programa que pode ser lido (quase) linearmente do início ao fim é muito mais fácil de entender do que um programa repleto de desvios (go to).
    - Estruturas de controle (if-then-else, while, for) permitem uma leitura mais compreensível

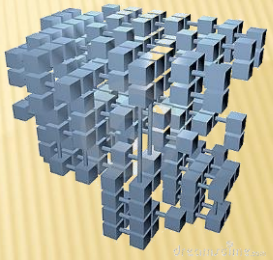
CRITÉRIOS PARA AVALIAR LPS

- Legibilidade
  - Instruções de Controle

```
loop1:
     if (incr >= 20) goto out;
loop2:
     if (sum > 100) goto next;
          sum += incr;
          goto loop2;
next:
     incr++;
     goto loop1;
out:
```

```
while (incr < 20) {
     while (sum <= 100) {
      sum += incr;
     }
     incr++;
}
```

CRITÉRIOS PARA AVALIAR LPS



- Legibilidade
  - Tipos e Estruturas de Dados
    - Facilidades para definir tipos e estruturas de dados
      - Exemplo: suponha uma linguagem que não define um tipo de dado **booleano**. Um tipo numérico poderia ser usado:
        - timeout = 1 (?! significado não claro)
        - timeout = true (significado claro)

CRITÉRIOS PARA AVALIAR LPS

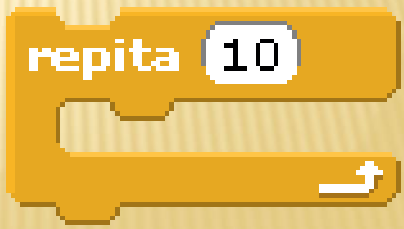
- Legibilidade
  - Considerações sobre a Sintaxe
    - A forma dos elementos tem um efeito significativo sobre a linguagem.
    - Formas identificadoras
      - Identificadores curtos prejudicam a legibilidade. Ex:
        - FORTRAN (6 caracteres)
        - BASIC ANSI (1 caractere ou 1 caractere seguido de 1 número)
        - Java (não tem limite)

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade
  - Considerações sobre a Sintaxe
    - Palavras especiais/reservadas
      - Exemplo: ***while***, ***for*** e ***begin-end***
        - Palavras **especiais** de uma linguagem podem ser usadas como **nomes de variáveis**?

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade
  - Considerações sobre a Sintaxe
    - Forma e significado
      - Projetar instruções de forma que sua aparência indique sua finalidade




```
int[] matriz = new int[2];
matriz[0] = 10;
matriz[1] = 20;
```

CRITÉRIOS PARA AVALIAR LPS

- Legibilidade

```
for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
```

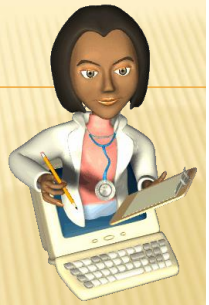
  - Considerações sobre a Sintaxe

    - **Boa** sintaxe é aquela que faz tudo **automaticamente** se **nada for informado**, mas permite que **tudo** seja **customizado**, da forma mais flexível e econômica possível. Ex;
      - laços for com incremento 2, 3, 4...

CRITÉRIOS PARA AVALIAR LPS

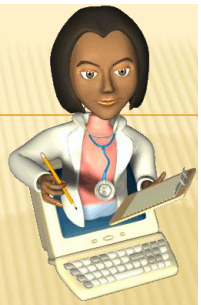
- Escritabilidade ou Capacidade de Escrita
  - Indica o quão facilmente uma linguagem pode ser usada em um determinado domínio de problema.
    - Linguagens Específicas de Domínio
      - (*domain-specific language* - DSL)

DSLs



- Vantagens
  - Expressividade.
  - Linguagem comum aos especialistas do domínio.
  - Oculta detalhes de implementação.
  - Ênfase no significado.

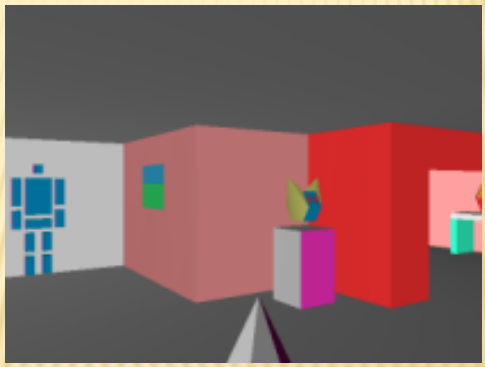
DSLs



- Exemplos
  - SQL – *databases*
  - HTML – *web layout*
  - XML – *data encoding*
  - UML – *visual modeling*
  - SysML – *systems modeling*
  - VHDL – *hardware design*
  - R - *for statistics*
    - https://www.r-project.org

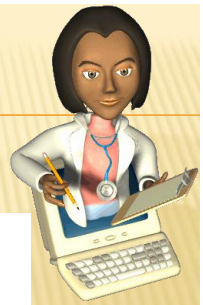
DSLs

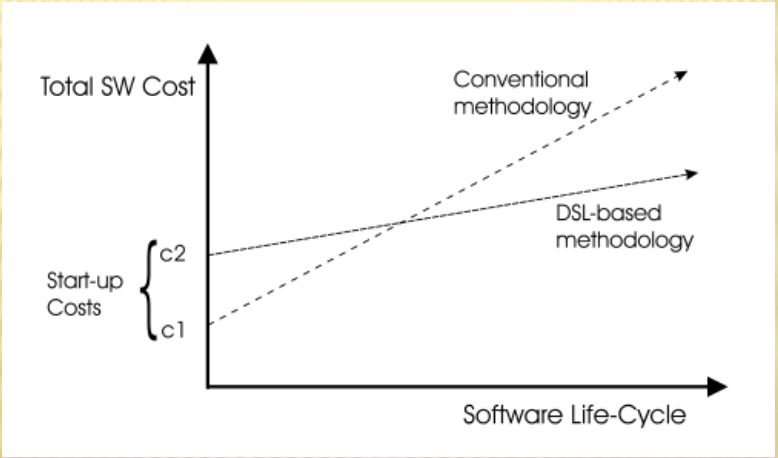
- *Virtual Reality Modeling Language (VRML)*



DSLs

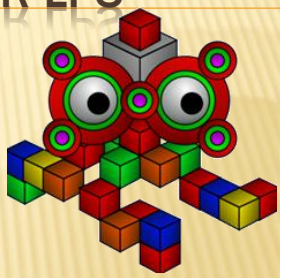
- Vantagens





Paul Hudak (1998) *Modular Domain Specific Languages and Tools*

CRITÉRIOS PARA AVALIAR LPS



- Escritabilidade
  - Simplicidade x Ortogonalidade
    - Poucos construtores, primitivas, e um diminuto conjunto de regras para combiná-los.
    - Se uma linguagem tiver um grande número de construções, programadores podem não se familiarizar com todas.
      - **Ortogonalidade demais pode prejudicar a escritabilidade!**

CRITÉRIOS PARA AVALIAR LPS

- Escritabilidade
  - Suporte para Abstração
    - Capacidade de definir e de usar estruturas ou operações complexas de maneira que permita ignorar muitos dos detalhes subjacentes. Ex:
      - abrir um **arquivo** via classe **File** em Java.

- Definir e usar estruturas complexas sem que os aspectos internos de implementação dessas estruturas sejam vistos pelo programador.

CRITÉRIOS PARA AVALIAR LPS

- Escritabilidade
  - Expressividade
    - Um conjunto relativamente conveniente de maneiras de especificar operadores. Exemplo:
      - count++ é mais conveniente que count = count + 1
      - a inclusão do ***for*** em muitas linguagens modernas

```
1  for (int i=0; i < 10; i++)
2  {
3      Console.WriteLine("Sei contar até " + i);
4  }
```

```
01.  ;...
02.  MOV AX, 0
03.  for:
04.        CMP AX, 10
05.        JE endfor
06.        ; Faça algo aqui
07.        ADD AX, 1
08.        JMP for
09.  endfor
10.  ;...
```

CRITÉRIOS PARA AVALIAR LPS

- Escritabilidade
  - Checagem de Tipos

    - Verifica se os valores atribuídos aos tipos estão de acordo com os definidos.
    - Linguagens **fortemente tipadas** são aquelas em que a declaração do tipo é obrigatória.
      - Todas as variáveis têm um tipo específico que tem que ser explicitado. Ex: Java, C++, Fortran e Cobol.

CRITÉRIOS PARA AVALIAR LPS

- Escritabilidade
  - Checagem de Tipos
    - Linguagens **fracamente tipadas** ou **dinamicamente** tipadas são aquelas em que **durante a execução** do programa podem **alterar o tipo de dados** contido em uma variável.
    - O programador **não** precisa fazer **conversões** de tipos. Ex: PHP, Javascript e Python.

CRITÉRIOS PARA AVALIAR LPS

- Escritabilidade
  - Checagem de Tipos
    - Linguagens **não tipadas**, são aquelas em que existe **apenas um tipo genérico** ou mesmo **nenhum** tipo de dados.
    - Entre estas está o Assembly.

CRITÉRIOS PARA AVALIAR LPS

- Escritabilidade
  - Manipulação de Exceções
    - Capacidade de interceptar erros em tempo de execução e por em prática medidas corretivas.
      - tomar as medidas adequadas e continuar (ou não) a execução

CRITÉRIOS PARA AVALIAR LPS

- Confiabilidade

  - Um programa é dito confiável se ele executa de acordo com suas especificações sob quaisquer circunstâncias. Ex:
    - Windows, Mac, Linux
    - Angry Birds sem rede

CRITÉRIOS PARA AVALIAR LPS

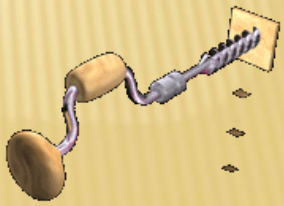
- **Custo,** o qual engloba:
  - Custo de **treinamento**
  - Custo de **escrever** programas na linguagem
  - Custo de **compilar** programas
  - Custo de **executar** programas
  - Custo do sistema de **implementação**
    - existência de editores/compiladores *free*
  - Custo da **confiabilidade** pobre
  - Custo de **manutenção**
  - ... e outras características.

CRITÉRIOS PARA AVALIAR LPS

- **Custo de criação, teste e uso de programas**
  - O **esforço** gasto para resolver um **problema** através da **implementação** de uma aplicação deve ser **minimizado**.
    - De que forma?

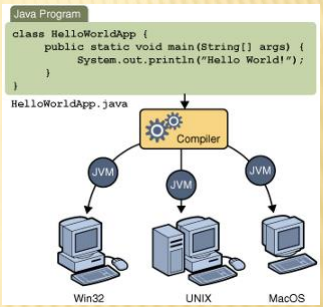
CRITÉRIOS PARA AVALIAR LPS

- Custo de criação, teste e uso de programas
  - Que ferramentas facilitam estas tarefas?
    - (1) Ambientes **gráficos** para desenvolvimento;
    - (2) Uso de **componentes** ao invés de implementação;
    - (3) Ferramentas de ***debug***;
    - (4) **Automação** de testes;
    - (5) Gerenciadores de **versões**;
    - (6)...



OUTROS CRITÉRIOS PARA AVALIAR LPS

- Portabilidade



  - Quão facilmente um programa pode ser movido de uma implementação para outra linguagem.

OUTROS CRITÉRIOS PARA AVALIAR LPS

- Generalidade
  - Seu uso em uma gama de aplicações.
    - A linguagem serve **para que**?
  - A precisão e a completeza da definição oficial da linguagem.
    - Está clara a **finalidade** da linguagem?
      - Prolog

CONSIDERAÇÕES

- Podem haver outros critérios:
  - portabilidade, generalidade, definição, etc.
- Quais outros você vê?

CONSIDERAÇÕES

- Critérios são subjetivos:
  - legibilidade, escritabilidade, etc.
- Critérios têm valores diferentes, dependendo da perspectiva:
  - implementadores, usuários, projetistas

RESUMO:

- **Legibilidade**
  - Quão facilmente um programa pode ser lido e entendido
- **Escritabilidade**
  - Quão facilmente uma linguagem pode ser usada para criar programas
- **Confiabilidade**
  - Conformidade com as especificações sob todas as condições
- **Custo**
  - O custo final de uma linguagem é uma função de suas características

$ É UM DOS MAIS SALIENTADOS

- Quais outros **custos** associados ao desenvolvimento de software que você consegue **visualizar**?
  - Como diminuir tais custos?

FIM DE PAPO!