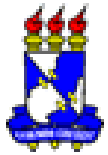
Introdução ao Estudo de Linguagens de Programação

Prof. Alberto Costa Neto
alberto@ufs.br

Linguagens de Programação



Departamento de Computação
Universidade Federal de Sergipe

Conteúdo

- Razões para estudar LP
- Domínios de programação
- Critérios de avaliação de linguagens
- Influências sobre o projeto de linguagens
- Paradigmas de LP
- Trade-offs no projeto de linguagens
- Métodos de implementação
- Ambiente de Programação

Razões para Estudar LP

- Maior capacidade de expressar idéias
  - Aprendizado de novas construções de LP
    - Mesmo que a LP não possua a capacidade aprendida, pode ser simulada (Objetos)
  - Limitações de uma linguagem
    - Tipos de estrutura de dados
    - Tipos de estrutura de controle
    - Tipos de abstrações que podem ser utilizadas
  - Formas de algoritmos também são limitadas

Razões para Estudar LP

- Maior conhecimento para escolha de LP apropriadas
  - Não conhecimento dos conceitos de LP
    - Utilização da LP familiar mesmo que não seja adequada ao novo projeto
  - Conhecimento dos recursos das LPs permite uma escolha mais consciente

Razões para Estudar LP

- Maior habilidade para aprender novas LP
  - Compreensão dos conceitos fundamentais
    - Facilita a percepção de como estes são incorporados ao projeto da LP aprendida
    - Facilita a leitura e o entendimento de manuais e do "marketing" relacionado a linguagens e compiladores

Razões para Estudar LP

- Entender melhor o significado da implementação
  - Visualização de como o computador executa várias construções da linguagem
    - Entendimento da eficiência relativa de construções alternativas a serem escolhidas
  - Capacidade de utilizar uma linguagem de forma mais inteligente, como ela foi projetada
    - String x StringBuffer em Java
  - Facilita a detecção e correção de certos *bugs*
    - Comparação de Strings com == em Java

Razões para Estudar LP

- Aumento da capacidade de projetar novas linguagens
  - Um exame crítico das linguagens de programação ajuda
    - No projeto de sistemas complexos
    - A examinar e avaliar esses produtos

Razões para Estudar LP

- Avanço global da computação
  - Em alguns casos uma linguagem não se torna popular porque aqueles com capacidade de optar ainda não estavam familiarizados com os conceitos de LPs
  - Permanência no FORTRAN em detrimento do ALGOL 60 (mais elegante, melhores estruturas de controle, recursão e bloco) na década de 60

Domínios de Programação

- Computadores são utilizados em áreas bem diferentes
  - Controle de usinas elétricas nucleares
  - Armazenamento e processamento de informações bancárias
  - Medicina
  - Computação pessoal
- Têm sido desenvolvidas LP para diversos domínios

Domínios de Programação

- Aplicações científicas
  - Estrutura de dados simples
  - Computações aritméticas com ponto flutuante
  - Estrutura de dados comuns: array e matriz
  - Estruturas de controle comuns: laços de contagem e de seleções
  - FORTRAN
  - ALGOL 60 e suas descendentes

Domínios de Programação

- Aplicações comerciais
  - COBOL
  - Produz relatórios elaborados
  - Maneira precisa de descrever e armazenar números decimais e dados caracteres
  - Capacidade de especificar operações aritméticas decimais

Domínios de Programação

- Inteligência Artificial
  - Uso de computações simbólicas em vez de numéricas
  - Programação funcional (LISP)
  - Programação lógica (Prolog)

Domínios de Programação

- Programação de sistemas
  - Sistemas operacionais e ferramentas de suporte
  - Décadas de 60 e 70 alguns fabricantes de computadores desenvolveram LP de alto nível
    - IBM – linguagem PL/S
    - Digital – BLISS
    - Burroughs – Extended ALGOL
  - Unix desenvolvido quase inteiramente em C

Domínios de Programação

- Software Web
  - Markup Languages (XHTML)
    - Não são LPs
    - JSP Standard Tab Library (JSLT)
    - Extensible Stylesheet Language Transformations (XSTL)
  - Java (Applets + Servlets + JSP...)
  - Linguagens de Scripting
    - JavaScript
    - Perl

Papel de LPs

- Tornar o processo de desenvolvimento de software (PDS) mais efetivo
- Propriedades Desejadas em um Software
  - Confiabilidade, Manutenibilidade, Eficiência
- Etapas do PDS
  - Especificação de Requisitos
  - Projeto do Software
  - Implementação
  - Validação
  - Manutenção

Critérios de Avaliação de Linguagens

- Legibilidade
- Redigibilidade
- Confiabilidade
- Custo

Critérios e Características de LP

	Critério		
	Legibilidade	*Redigibilidade*	*Confiabilidade*
Simplicidade	x	x	x
Ortogonalidade	x	x	x
Tipos de Dados	x	x	x
Sintaxe	x	x	x
Suporte a Abstração		x	x
Expressividade		x	x
Checagem de Tipos			x
Tratamento de Exceções			x
Restrição a Alias			x

Critérios de Avaliação de Linguagens

- Legibilidade
  - Facilidade com que os programas podem ser lidos e entendidos
  - Facilidade de manutenção é, em grande parte, determinada pela legibilidade dos programas

Critérios de Avaliação de Linguagens

- Legibilidade
  - Simplicidade global
    - Pequeno número de componentes básicos
    - Não possuir multiplicidade de recursos
      - Mais de uma maneira de realizar uma operação particular
    - Evitar Sobrecarga de Operador
      - Usar + para soma de vetores

Critérios de Avaliação de Linguagens

- Legibilidade
  - Ortogonalidade
    - Um conjunto relativamente pequeno de construções primitivas pode ser combinado, em um número relativamente pequeno de maneiras, para construir as estrutura de controle e de dados da linguagem
      - Vetor e Registro podem ser passados como parâmetros?
      - E podem ser retornados de funções? Pascal não suporta!
      - Vetor de vetores são permitidos?
    - O significado de um recurso ortogonal de linguagem é livre do contexto de sua ocorrência em um programa

Critérios de Avaliação de Linguagens

- Legibilidade
  - Ortogonalidade
    - Torna a linguagem mais fácil de aprender
    - Diminui as restrições de programação
    - Muita ortogonalidade também pode causar problemas
    - Simplicidade
      - Combinação de um número relativamente pequeno de construções primitivas
      - Uso limitado do conceito de ortogonalidade

Critérios de Avaliação de Linguagens

- Legibilidade
  - Instruções de controle
    - Restrição à instrução de desvio incondicional (goto)
    - Incluir instruções suficientes com o objetivo de eliminar a necessidade de utilização de goto (if, case, for, while, repeat, ...)
    - As instruções de controle devem ser bem projetadas
  - Tipos de dados e estruturas
    - Facilidade para definir tipos e estrutura de dados auxilia a legibilidade
      - Falta de um tipo boolean em C
        - » usar 0 como false e 1 como true

Critérios de Avaliação de Linguagens

- Legibilidade
  - Sintaxe
    - Identificadores (Poucas restrições na escolha)
      - ANSI BASIC (1 Legra + 1 Dígito) – Ex: B2
      - FORTRAN 77 (6 caracteres) – Ex: NUMALU
    - Palavras especiais
      - begin end x enf if + end loop
      - FORTRAN permite usá-las em nomes de variáveis

Critérios de Avaliação de Linguagens

- Legibilidade
  - Sintaxe (continuação)
    - Forma e Significado
      - * em C (*p = (*p)*q;)
        - » *Retorno do endereço apontador por p*
        - » *Retorno do conteúdo da célula apontada por p*
        - » *Operação de multiplicação*
      - this em Java
        - » Referência ao próprio objeto
        - » Chamada a outro construtor da classe

Critérios de Avaliação de Linguagens

- Redigibilidade
  - Facilidade com que uma linguagem pode ser utilizada para criar programas para o domínio de problema escolhido
  - Simplicidade e Ortogonalidade
    - Pequeno número de construções primitivas
    - Um conjunto consistente de regras para combinar as construções da linguagem

Critérios de Avaliação de Linguagens

- Redigibilidade
  - Suporte a abstração
    - Habilidade para definir e usar estruturas e operações complexas de forma que muitos detalhes sejam ignorados
      - Processos (subprogramas)
      - Dados (estruturas de dados)
        - » FORTRAN não suporta registros
  - Expressividade
    - Facilidade de expressar computação em programas pequenos (++ em C e Java)
    - for (em C e Java) permitem criar laços com contadores mais facilmente do que com o while

Critérios de Avaliação de Linguagens

- Confiabilidade
  - Checagem de tipos (estática x dinâmica)
  - Tratamento de exceções (try / catch / finally)
  - Aliasing
    - Existência de diferentes métodos de acesso a uma mesma célula de memória
    - Mais de um apontador para a mesma variável
  - Legibilidade e Redigibilidade
    - Quanto mais fácil descrever e ler, maior a confiabilidade do programa

Critérios de Avaliação de Linguagens

- Custo
  - Treinamento de programadores
  - Escrever programas na linguagem
  - Compilar programas na linguagem
  - Executar programas
  - Sistema de implementação da linguagem
  - Custo da má confiabilidade
  - Manutenção dos programas

Outros critérios

- **Eficiência** (checagem de tipos e exceções)
- **Modificabilidade** (interfaces x implementação)
- **Portabilidade**
  - Programas podem rodar em diferentes ambientes/plataformas
  - Normalmente perde em eficiência de execução
- **Facilidade de Aprendizado**
- **Generalidade**
  - Poder ser aplicada em uma ampla gama de situações
- ***Well-definedness***
  - Definição completa e precisa da linguagem

Bibliografia (livros)

- Concepts of Programming Languages (Robert W. Sebesta)
  - Capítulo 1
- Linguagens de Programação (Flávio Varejão)
  - Capítulo 1