Fundamentos de Programação 1

Slides N. 2 – E / Prof. SIMÃO

***Slides* elaborados pelo Prof. Robson Linhares**

http://www.dainf.ct.utfpr.edu.br/~robson/

Jean Marcelo SIMÃO

Fundamentos de Programação I

Linguagens e paradigmas de programação

Tópicos

- Classificações das linguagens de programação
- Paradigmas de programação
- Paradigma estruturado x paradigma OO
- Histórico das linguagens de programação

Classificação das linguagens de programação

- Vários critérios de classificação
  - **Por tecnologia de execução**
  - **Por grau de abstração**
  - **Por paradigma**
  - Por geração
  - Por estrutura de tipos
  - ...

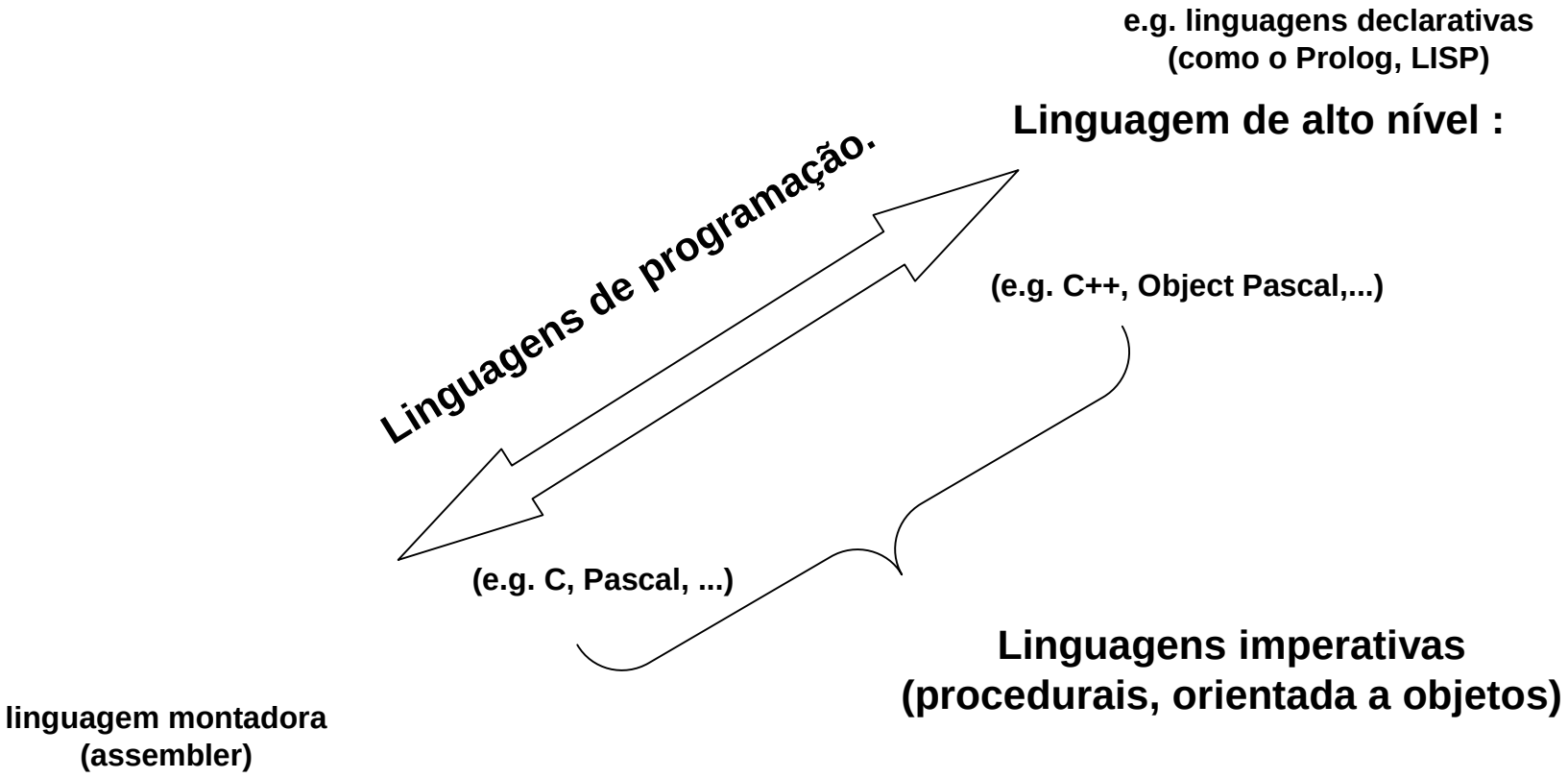
Classificação das linguagens de programação – por tecnologia de execução

- Linguagem compilada
  - Programas são traduzidos (compilados) para instruções de máquina antes de serem executados
  - Programa é compilado uma vez e executado várias vezes
  - Tendem a ter melhor desempenho – quem executa já é código de máquina...
  - Tendem a não fazer verificações sofisticadas de erros de tempo de execução
- Exemplos
  - C, C++, Pascal, Java (parcialmente),...

Classificação das linguagens de programação – por tecnologia de execução

- Linguagem interpretada
  - Programas são traduzidos (interpretados) para instruções de máquina durante a sua execução
    - Processo pode ser efetuado por um interpretador ou por uma máquina virtual
  - Interpretação ocorre sempre que o programa é executado
    - Exceção: JIT (just in time compiling)
  - Tendem a ter pior desempenho – interpretador/máquina virtual utilizam tempo de processamento
  - Tendem a ser mais sofisticados na detecção de erros em tempo de execução – depende da tecnologia do interpretador/máquina virtual
- Exemplos
  - BASIC, Java (parcialmente), C# (parcialmente), linguagens de script (VBScript, JavaScript, etc.)

Classificação das linguagens de programação – por grau de abstração



Classificação das linguagens de programação – por grau de abstração

- Linguagem de baixo nível
  - Mais “próxima” da máquina
  - Difícil implementação pelo ser humano - abstrações muito simples
  - Em comparação com alto nível, programas em baixo nível tendem a ser:
    - Menores – ocupam menos bytes
    - Mais rápidos
    - **MUITO mais lentos para se implementar**
  - Exemplos: diversos tipos de assembly (linguagem de montagem)

Classificação das linguagens de programação – por grau de abstração

- Linguagem de alto nível
  - Mais “próxima” do ser humano – oferece abstrações mais complexas
  - Não executável diretamente pela máquina
  - Em comparação com baixo nível, programas em alto nível tendem a ser:
    - Maiores – ocupam mais bytes
    - Mais lentos
    - **MUITO mais rápidos para se implementar**
  - Exemplos: uma infinidade de linguagens de programação...
    - C, Pascal, Fortran, LISP, C++, Java, C#, Cobol, Perl, Python, ...

Paradigmas de programação

- Paradigma – “idéia”
- Várias definições para paradigma de programação:
  - Visão que o programador possui sobre a estruturação e a execução de um programa
  - Modelos, padrões e estilos suportados por linguagens de programação com características comuns
  - Técnicas ou conceitos amplamente utilizados por programadores para programar
- Paradigmas de programação podem definir um conjunto de “permissões” e “proibições” em técnicas de programação – objetivo é sempre facilitar o desenvolvimento em um ou mais aspectos
- Uma linguagem de programação pode suportar mais do que um paradigma

Paradigmas de programação

- Exemplos de paradigmas de programação:
  - **Programação estruturada**
  - Programação imperativa
  - Programação declarativa
  - Programação orientada a eventos
  - Programação lógica
  - Programação orientada a aspecto
  - **Programação orientada a objetos**
  - ...

Programação estruturada

- Paradigma de programação de alto nível
- Define que os programas podem ser reduzidos a *sequências de comandos, decisões* e *iterações*
  - Sequência de comandos: ações que são tomadas sempre em sequência, respeitando uma ordem
  - Decisões: situações em que se deve decidir por prosseguir a execução por um caminho dentre vários, de acordo com alguma condição
  - Iteração: repetição condicionada de determinada sequência de comandos

Programação estruturada

- Exemplo de algoritmo estruturado:

```
Algoritmo AreaTri
   real base, altura, area;
   repita
        escreva “Forneça base e altura:”;
        leia base, altura;
        se base <=0 ou altura <= 0 então
               escreva “Valor(es) inválido(s)”;
        Fim se
   enquanto base <= 0 ou altura <= 0;
   Fim repita
   area <- base * altura /2;
   escreva “A área do tri é “, area;
Fim algoritmo
```

iteração

decisão

sequência

Programação estruturada

- Um programa estruturado pode ser *modularizado* – decomposto em módulos com tarefas específicas
- Vantagens de um módulo:
  - Reutilização
  - Facilidade de manutenção
  - Divisão de tarefas
- Pode-se afirmar que um programa estruturado consiste em um conjunto de módulos “disparados” de acordo com uma lógica

Programação estruturada

- Exemplo de algoritmo estruturado com módulos:

```
Função inteiro FATORIAL(N)
   inteiro N, RES;
   RES <- 1;
   enquanto N > 1 faça
        RES <- RES * N;
        N <- N – 1;
   Fim enquanto
   FATORIAL <- RES;
Fim função
Algoritmo Combinações
   inteiro N, P, COMB;
   N <- 12;
   P <- 5;
   COMB <- FATORIAL(N) / (FATORIAL(P) * FATORIAL(N-P));
   escreva “O número de combinações de 12 elementos 5 a 5 é “, COMB;
Fim algoritmo
```

módulo

Programação orientada a objetos (OO)

- Paradigma de programação de alto nível
- OO apresenta uma visão diferente da programação estruturada
  - Estruturada: programas são módulos com lógica (decisões, iterações, comandos) executados em uma determinada sequência
  - OO: programas são compostos por **objetos** que interagem entre si
- Os objetos em um programa OO representam entidades que existem no contexto daquele programa
- Os objetos de um programa OO são criados a partir de classes (modelos de representação)
- Exemplo: diagrama de classes simplificado do sistema de RH de uma empresa

Programação orientada a objetos (OO)

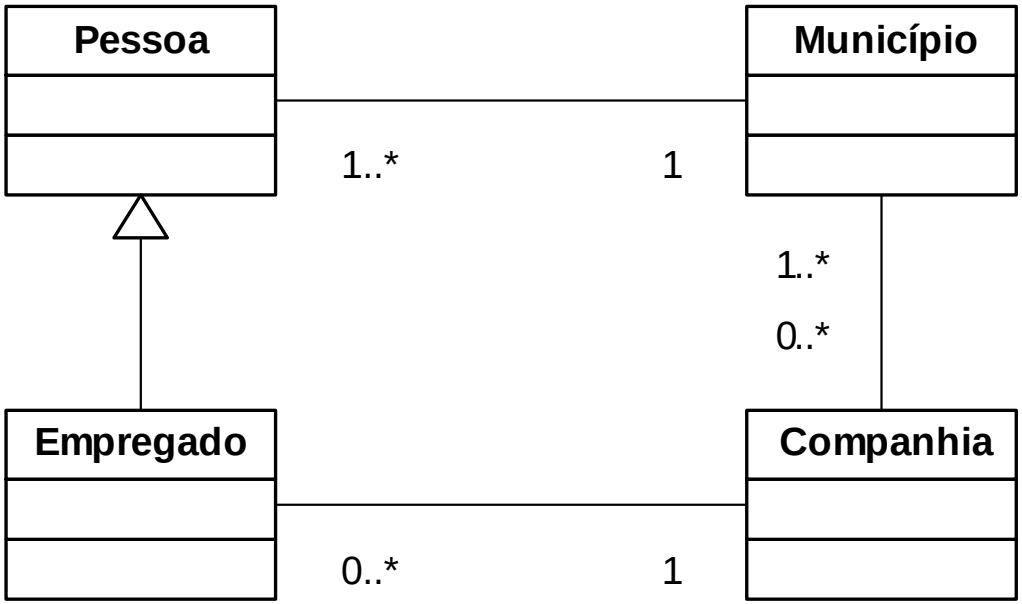


Diagrama de classes de um sistema de RH

Programação estruturada x OO

- Não cabe comparação – filosofias diferentes que se adaptam melhor ou pior a sistemas com naturezas diferentes
- Via de regra, sistemas mais complexos são melhor modelados com a abordagem OO
  - Preço: programas maiores, menor desempenho

Classificação das linguagens de programação – por paradigma

- Exemplos de linguagens puramente estruturadas
  - C, Basic, Fortran, Pascal, Cobol, ...
- Exemplos de linguagens com suporte a OO
  - C++, Java, Perl, Object Pascal (Delphi), Visual Basic, C#,...

Histórico das linguagens de programação

- Dezenas de linguagens foram desenvolvidas, com diferentes propósitos
- Cronologicamente (por gerações):
  - Primeira geração
    - linguagens de máquina – códigos binários das instruções utilizados diretamente
  - Segunda geração
    - linguagens de baixo nível de montagem (assembly). Representação mnemônica dos códigos de máquina

Histórico das linguagens de programação

- Terceira geração
  - linguagens de programação estruturadas
  - Exemplos: Fortran (1954), COBOL (1959), BASIC, Pascal, C, etc.
- Quarta geração
  - Linguagens de programação orientadas a objeto (OO)
  - Linguagens de programação não-procedurais (declarativas)
  - Linguagens com suporte a RAD (*Rapid Application Development)*
  - Exemplos: C++ (1983), Java (1995), SQL (, Delphi (1985), Mathematica, etc.
- Quinta geração
  - linguagens lógicas
  - Exemplos: Prolog, etc.

Referências online

- Tipos de linguagens de programação http://www.criarweb.com/artigos/685.php
- Computer languages history (preview) - http://www.levenez.com/lang/history.html
- Wikipedia