

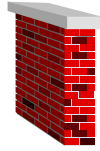
Programação III

Mudança de Paradigma
Jocélio Passos
joceliopassos@bol.com.br

Paradigma

■ O que é um paradigma ?

- Modelo
- Protótipo
- Forma
- Método
- Proposta
- Maneira
- Conjunto de teorias, padrões e métodos que, juntos, representam uma forma de organizar o conhecimento
- Modo de perceber o mundo



2/2/2007

2

Tipos de Paradigma de Programação

- Funcional
- Lógico
- Imperativo
- Orientado a Objeto

2/2/2007

3

Paradigma Funcional

- Linguagens baseadas em funções matemáticas
- $\text{cubo}(x) \equiv x * x * x$
- Não usa variáveis
- Nem instruções de atribuição
- Queda de eficiência (em relação a imperativa)
- Nível de programação mais elevado
- Dita o que deve ser feito e não como
- Uso intenso da recursividade (Repetição)
- Frequentemente são interpretadas
- Podem ser também compiladas

2/2/2007

4

Paradigma Funcional (Exemplos)

- Lisp
 - Década de 60
 - List Processor
 - Uso generalizado
 - Aplicações de IA
 - Desenvolvimento de Sistemas Especialistas
 - Representação do Conhecimento
 - Aprendizagem de máquina
 - Processamento de Linguagem Natural
 - Sistemas de Treinamento Inteligente
 - Modelagem da fala e visão
 - $(+ 5 7)$ $+5 (7)$: resultado 12

2/2/2007

5

Paradigma Funcional

- Scheme
 - Década de 70
 - Dialeto simples
 - Aplicações educacionais
 - Popular em colégios e universidades
 - $(\text{CAR } '(A B C))$ A
 - $(\text{CDR } '(A B C))$ (B C)

2/2/2007

6

Paradigma Funcional

- COMMON LISP
 - Início da década de 80
 - Objetivou combinar os diversos dialetos LISP
 - Pretensão comercial em IA
- ML
 - Década de 80
 - MetaLanguage
 - Sintaxe semelhante ao Pascal
 - Funcional / imperativa
 - Manipulação de exceções

2/2/2007

7

Paradigma Funcional

- Miranda
 - Início da década de 80
- Haskell
 - Baseada na ML e Miranda
 - É puramente funcional
 - Uso em Laboratórios de Pesquisa e Universidades
 - Usa a técnica chamada **avaliação preguiçosa**
 - $\text{fact } 0 = 1$
 $\text{fact } n = n * \text{fact } (n-1)$
- APL

2/2/2007

8

Paradigma Lógico

- Baseada em Regras
- Uso de declaração (não de procedimento)
- Usa Cálculo de Predicados
- Uso de proposições
- Baseada na lógica formal
- Uso na Implementação de SGBD
- Implementação de Sistemas Especialistas
- Linguagem Natural
- Área Educacional

2/2/2007

9

Paradigma Lógico

- Prolog
 - Década de 70
 - Desenvolvida por Alain Colmerauer e Philippe Roussel
 - Uso generalizado
 - Utiliza o backtracking
 - Não é pura
 - Execução lenta
 - Exemplo de linhas de código
 - homem (joao).
 - gosta (joao, fruta).
 - Pai (carlos,joao).

2/2/2007

10

Paradigma Imperativo

- Baseia-se na arquitetura de von Neumann
- Algoritmos especificado com grandes detalhes
- Dita o que deve ser feito e como
- A ordem dos comandos faz diferença
- Programação estruturada (ou não) GOTO
- Desenvolvimento top-down
- Programação Procedural / modular
- Sub-programas (procedimentos e funções)
- Não há organização dos dados
- Dados locais e públicos
- Baseada em verbos (abrir, fechar, ler etc)

2/2/2007

11

Paradigma Imperativo

- Desvantagens
 - Parte de um modelo mental anômalo
 - Dificuldade de modificar e expandir programas
 - Dificuldade em reutilizar programas

2/2/2007

12

Paradigma Imperativo

- Exemplos :
 - Fortran (1957)
 - Algol (1958)
 - COBOL (1960)
 - BASIC (1964)
 - PL/I (1964)
 - Pascal (1972)
 - C (1970)
 - Modula-2 e 3 (1977 e 1989)
 - ADA - DoD (1983)
 - xBase - dBase, FoxBase, Joinner (década de 80)
 - Clipper (final da década de 80)

2/2/2007

13

Paradigma Orientado a Objeto

- Linguagens desenvolvidas a partir das imperativas
- Simulação do mundo real
- Características fundamentais
 - Abstração de dados
 - Vinculação dinâmica
 - Herança
- Atributos/ Métodos
- Dados e código organizados em classes

2/2/2007

14

Paradigma Orientado a Objeto

- SIMULA 67
 - Desenvolvida entre 1962 e 1964
 - Uso de Classes
 - Simulação do mundo real
- SmallTalk
 - Década de 70
 - Unidades de programas são objetos
 - Tudo é objeto
 - Envio de mensagens
 - Ambiente completo de desenvolvimento
 - Interface gráfica, janelas e mouse

2/2/2007

15

Paradigma Orientado a Objeto

- C++
 - Criada na década de 1980 (AT&T Bell Lab.)
 - Extensão da Linguagem C
 - Linguagem Híbrida
 - Construtores e Destrutores
 - Sobre carga (overloading)
 - Chamada inicialmente de C com Classes
 - Herança múltipla
 - Classes Abstratas
 - Manipulação de exceções
 - Linguagem grande e complexa
 - Desalocação explícita de memória
- Manteve os recursos inseguros do C

2/2/2007

16

Paradigma Orientado a Objeto

- Visual Object (VO)
 - 1994
 - Produzido pela Computer Associates (CA)
 - Baseada no Clipper (Clipper Visual)
 - Fácil implementação dos conceitos OO
 - Pretendia ser a linguagem comercial que implementou a orientação a objeto
 - O projeto capotou

2/2/2007

17

Paradigma Orientado a Objeto

- Delphi
 - 1995
 - Produzido pela Borland/Inprise
 - Derivado do Pascal
 - Híbrido
- Java

2/2/2007

18

Bibliografia

- Sebesta, Robert W
Conceitos de Linguagens de Programação,
Bookman, 2000

2/2/2007

19

Exercícios

- 1) Qual o melhor paradigma ?
- 2) Por que as linguagem funcionais são mais lentas que as imperativas ?
- 3) Quais as aplicações da Linguagem LISP ?
- 4) Que técnica difere Haskell das demais linguagens funcionais ?
- 5) Quais as aplicações das linguagens Lógicas ?
- 6) O que povoa o mundo da SmallTalk ?
- 7) Que novidade no tocante ao desenvolvimento de software foi criada na SmallTalk ?
- 8) Quais os três conceitos básicos da programação orientada a objeto ?

2/2/2007

20

Exercícios

- 9) Por que o C++ inclui os recursos do C que são conhecidos como inseguros ?
- 10) O que é uma linguagem híbrida ?
- 11) Cite uma diferença básica entre a linguagem imperativa e a funcional !
- 12) Cite uma diferença básica entre a linguagem imperativa e a lógica !
- 13) Cite uma diferença básica entre a linguagem imperativa e a orientada a objeto !
- 14) O que levou a não adesão do VO no mundo da programação ?

2/2/2007

21

Exercícios

- 15) Porque a função abaixo escrita em C não é compilavel em C++ ?

```
int MinhaFuncao (int x)
{int class;
  class = x + 1;
  return class;}
```

2/2/2007

22