

GRUPO: TECNOLIFE

ELISA FERNANDES
KEITH CARVALHO
LUCAS VILLAR
NAYANE GETIRANA
VALMON GAUDENCIO

SistemasTecnoLife.blogspot.com

Paradigmas de linguagem de programação

ALUNA:

ELISA CRAVO FERNANDES

SistemasTecnolife.blogspot.com

Introdução à LP's

Conteúdo Programático

- ▣ 1. O que é Paradigma?
- ▣ 2. O que é Paradigma de Programação?
- ▣ 3. Características de Paradigma de Programação
- ▣ 4. História
- ▣ 5. Por que estudar LP's?
- ▣ 6. Para que servem as LP's?
- ▣ 7. Propriedades Desejáveis
- ▣ 8. Por que tanta linguagem?

O que é um Paradigma?

- Modelo, padrão
- É um modelo imperativo de uma realidade
- Permite organizar as idéias com vista:
 - ▣ Ao atendimento dessa realidade
 - ▣ À determinação de qual é a melhor forma de atuar sobre essa realidade

O que é Paradigma de Programação?



- Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características comuns.
- Uma linguagem de programação é um método padronizado para expressar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.

Características de Paradigmas de Programação



- Gramática e significado bem definidos [sintaxe: gramática (forma); semântica: significado]
- Implementável (executável) com eficiência “aceitável”
- Universal: deve ser possível expressar todo problema computável

História

- A primeira linguagem de programação para computadores foi provavelmente Plankalkül, criada por Konrad Zuse na Alemanha Nazista, mas que teve pouco ou nenhum impacto no futuro das linguagens de programação.
- A primeira linguagem de programação de alto nível amplamente usada foi Fortran, criada em 1954.

Por que estudar LP's ?

- Programação é central para computação
- Linguagens de consulta a banco de dados têm muitos dos conceitos de linguagens de programação
- Linguagens de comando de sistemas operacionais têm muitos dos conceitos de linguagens de programação
- Linguagens de descrição de hardware têm muitos dos conceitos de linguagens de programação
- Processamento de linguagem natural é relacionado a processamento de linguagens de programação

Para que servem as LP's ?

- Permitem que programadores tenham uma maior produtividade, ajudando-os a expressar suas intenções mais facilmente do que quando comparado com a linguagem que um computador entende nativamente (código de máquina).
- Tornam os programas menos dependentes de computadores ou ambientes computacionais específicos

Propriedades Desejáveis em uma LP

- Legibilidade
- Regibilidade
- Confiabilidade
- Eficiência
- Facilidade de aprendizado
- Ortogonalidade
- Reusabilidade
- Modificabilidade
- Portabilidade

Por que tantas LP's ?



- Propósitos diferentes
- Avanços tecnológicos
- Interesses comerciais
- Cultura e background científico

ALUNO:

LUCAS VILLAR

SistemasTecnolife.blogspot.com

Programação Imperativa e Orientada a Objeto

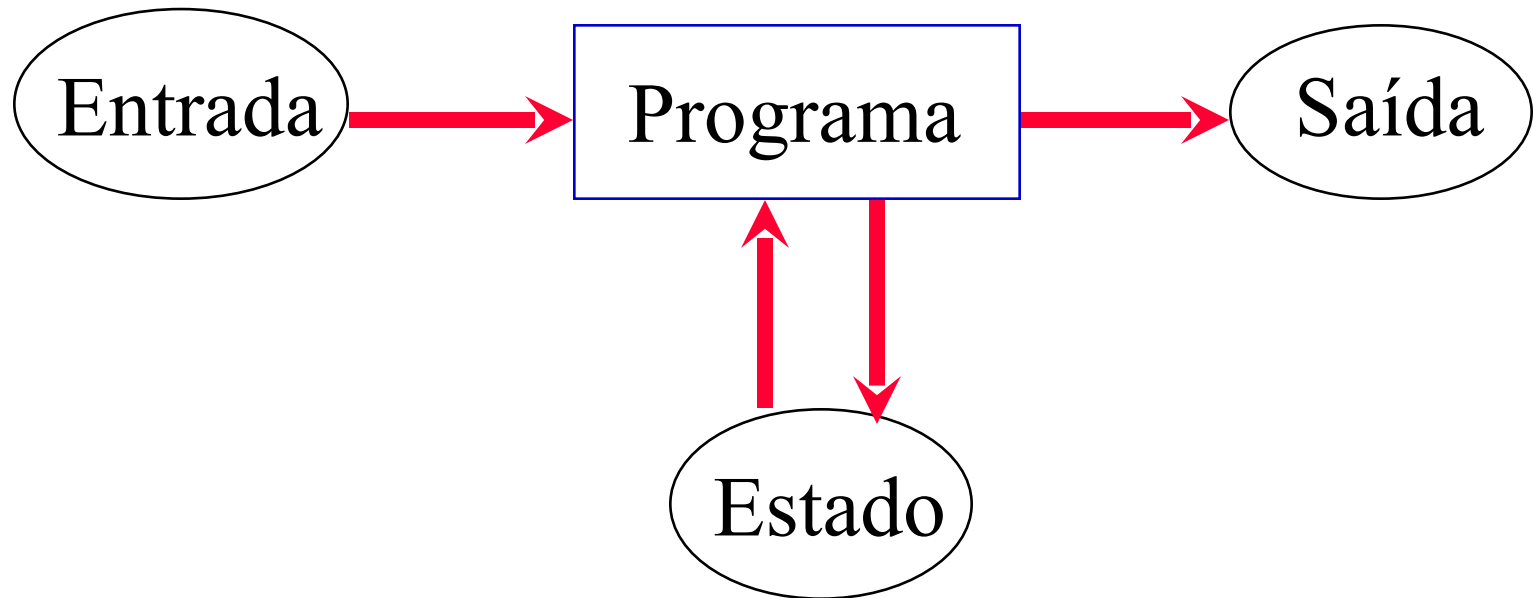
Conteúdo

- O paradigma Imperativo
- Vantagens e Desvantagens do modelo Imperativo
- Alguns exemplos de linguagens do modelo imperativo
- O paradigma Orientado a Objeto
- Vantagens e Desvantagens do modelo OO
- Alguns exemplos de linguagens do modelo OO

○ Paradigma Imperativo

- Programas centrados no conceito de um estado (modelado por variáveis) e ações (comandos) que manipulam o estado
- Paradigma também denominado de procedural, por incluir sub-rotinas ou procedimentos como mecanismo de estruturação
- Primeiro paradigma a surgir
 - ▣ Ainda é muito utilizado

Modelo Computacional do Paradigma Imperativo



Vantagens do modelo imperativo



- Eficiência (embute modelo de Von Neumann)
- Modelagem “natural” de aplicações do mundo real
- Paradigma dominante e bem estabelecido

Desvantagens do paradigma imperativo

- Relacionamento indireto entre E/S resulta em:
 - difícil legibilidade
 - erros introduzidos durante manutenção
 - descrições demasiadamente operacionais focalizam o **como** e não o **que**

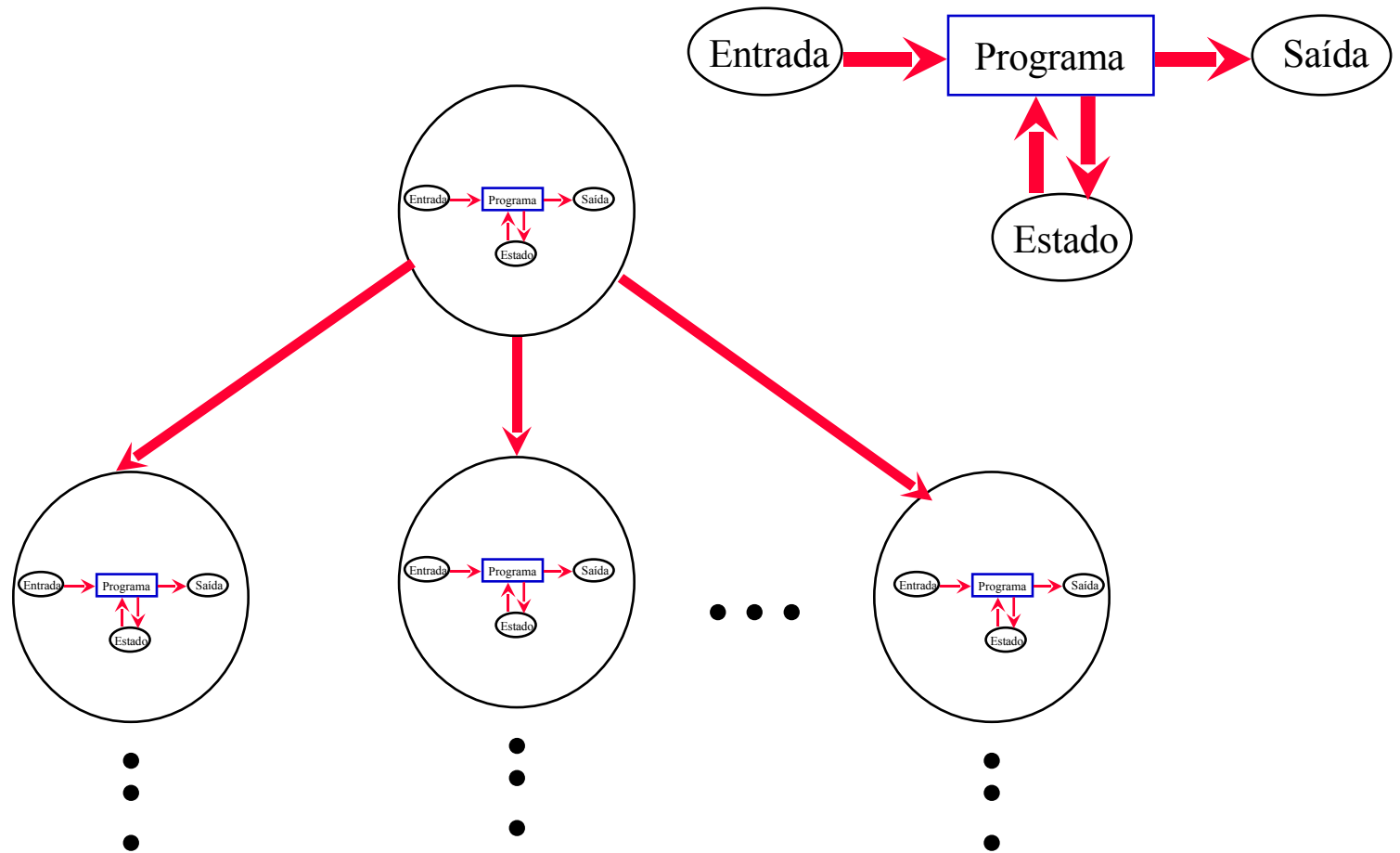
Alguns exemplos de linguagens do modelo imperativo

- Ada
- Algol
- Basic
- C
- Cobol
- Fortran
- Pascal

○ Paradigma Orientado a Objetos

- Não é um paradigma no sentido estrito: é uma subclassificação do imperativo
- A diferença é mais de metodologia quanto à concepção e modelagem do sistema
- A grosso modo, uma aplicação é estruturada em módulos (classes) que agrupam um estado (atributos) e operações (métodos) sobre este
- Classes podem ser estendidas e/ou usadas como tipos (cujos elementos são objetos)

Modelo Computacional do Paradigma Orientado a Objetos



Vantagens do Paradigma Orientado a Objetos

- Todas as do estilo imperativo
- Classes estimulam projeto centrado em dados: modularidade, reusabilidade e extensibilidade
- Aceitação comercial crescente.

Problemas do Paradigma OO

- Semelhantes aos do paradigma imperativo, mas amenizadas pelas facilidades de estruturação

Alguns exemplos de linguagens do modelo OO

- Ada
- Algol
- Basic
- C
- Cobol
- Fortran

ALUNO:

VALMON GAUDENCIO

SistemasTecnoLife.blogspot.com

Paradigma Paralelo e Distribuído

Conteúdo Programático



- 1. Introdução
- 2. Paradigma Paralelo
- 3. Paradigma Distribuído

Introdução à Paradigma Paralelo e Distribuído

- Paradigma Paralelo e Distribuído
- Vantagens e Dificuldades
- Plataformas de Execução

Panorama Atual



Paradigma paralelo

□ O que é?

- Consiste em executar simultaneamente várias partes de uma mesma aplicação.
- Tornou-se possível a partir do desenvolvimento de sistemas operacionais multi-tarefa, multi-thread e paralelos.

□ Aplicações são executadas paralelamente:

- Em um mesmo processador.
- Em uma máquina multiprocessada.
- Em um grupo de máquinas interligadas que se comporta como uma só máquina.

Paradigma Distribuído

- ❑ O que é?
- ❑ Consiste em executar aplicações cooperantes em máquinas diferentes.
- ❑ Tornou-se possível a partir da popularização das redes de computadores.
- ❑ **Aplicações são executadas em máquinas diferentes interligadas por uma rede.**
- ❑ Intranets
- ❑ Internet
- ❑ Outras redes públicas ou privadas

Diferenças

□ Acoplamento

- Sistemas paralelos são fortemente acoplados: compartilham hardware ou se comunicam através de um barramento de alta velocidade
- Sistemas distribuídos são fracamente acoplados

□ Previsibilidade

- O comportamento de sistemas paralelos é mais previsível; já os sistemas distribuídos são mais imprevisíveis devido ao uso da rede e a falhas.

□ Influência do Tempo

- Sistemas distribuídos são bastante influenciados pelo tempo de comunicação pela rede; em geral não há uma referência de tempo global.
- Em sistemas paralelos o tempo de troca de mensagens pode ser desconsiderado.

□ Controle

- Em geral em sistemas paralelos se tem o controle de todos os recursos computacionais; já os sistemas distribuídos tendem a empregar também recursos de terceiros.

Vantagens

- Usam melhor o poder de processamento.
- Apresentam um melhor desempenho.
- Permitem compartilhar dados e recursos.
- Podem apresentar maior confiabilidade.
- Permitem reutilizar serviços já disponíveis.
- Atendem um maior número de usuários.
- ...

Dificuldades

- Desenvolver, gerenciar e manter o sistema.
- Controlar o acesso concorrente a dados e a recursos compartilhados.
- Evitar que falhas de máquinas ou da rede comprometam o funcionamento do sistema.
- Garantir a segurança do sistema e o sigilo dos dados trocados entre máquinas.
- Lidar com a heterogeneidade do ambiente.
- ...

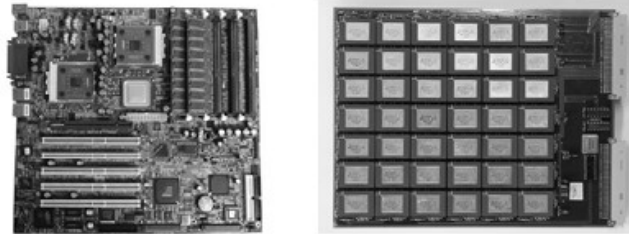
Plataformas de Execução

- Um S.O. multitarefa permite simular o paralelismo em um único processador, alternando a execução de processos.
- Um processador com núcleo múltiplo permite paralelismo real entre processos, executando múltiplas instruções por ciclo.



Plataformas de Execução

- Uma Placa-Mãe Multiprocessador permite que cada processador execute um processo.

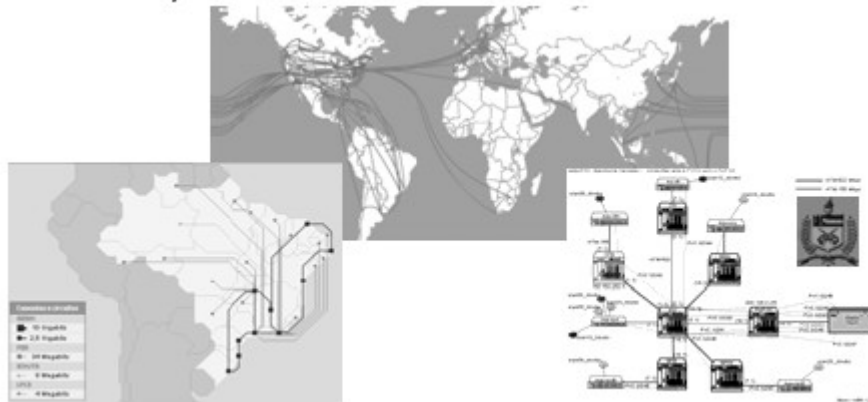


- Um Cluster é uma coleção de baixo custo para processamento de alto desempenho.



Plataformas de Execução

- Computação Distribuída é possível em redes, como numa Intranet e na Internet.



ALUNA:

KEITH CARVALHO

SistemasTecnolife.blogspot.com

Paradigma funcional

Linguagem funcional



- ▣ É um paradigma de programação baseada em funções matemáticas.
- ▣ Paradigma funcional tem sido mais usado academicamente que no desenvolvimento de software.
- ▣ O interpretador de linguagem funcional atua como uma calculadora: lê, calcula e mostra o resultado.

Vantagens do Paradigma funcional



- As linguagens funcionais fornecem um alto nível de abstração, o que faz com que os programas funcionais sejam mais pequenos, claros, rápidos.

Desvantagem do Paradigma funcional



- Os programas funcionais podem ser menos eficientes.

Exemplos

- Lisp, ML, Miranda e Haskell
- Lisp: A primeira LP funcional criada por John McCarthy no fim dos anos 50.
- Haskell: surgiu no fim dos anos 80.

ALUNA:

NAYANE GETIRANA

SistemasTecnoLife.blogspot.com

Paradigma Lógico

○ Paradigma Lógico

- Programas são relações entre E/S
- Estilo declarativo, como no paradigma funcional
- Na prática, inclui características imperativas, por questão de eficiência
- Aplicações: prototipação em geral, sistemas especialistas, banco de dados, ...

Modelo Computacional do Paradigma Lógico



Visão Crítica do Paradigma Lógico

□ Vantagens

Em princípio, todas do paradigma funcional

Permite concepção da aplicação em um alto nível de abstração (através de associações entre E/S)

□ Problemas

Em princípio, todos do paradigma funcional

Linguagens usualmente não possuem tipos, nem são de alta ordem

Exemplo mais conhecidos



- Prolog

Tendências

- Integração de paradigmas
 - ▣ Aumentar o domínio da aplicação
- Cautela
 - ▣ Não violar os princípios básicos do paradigma