

INF01202

Algoritmos e Programação

Profa. Cora Pinto Ribeiro
2008/2

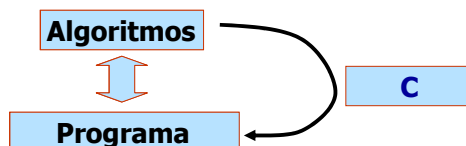
Slide 1

Profa. Cora H.F. Pinto Ribeiro



Objetivos da Disciplina

- ✓ Introduzir o uso do computador na solução de problemas, através da construção de **PROGRAMAS**.

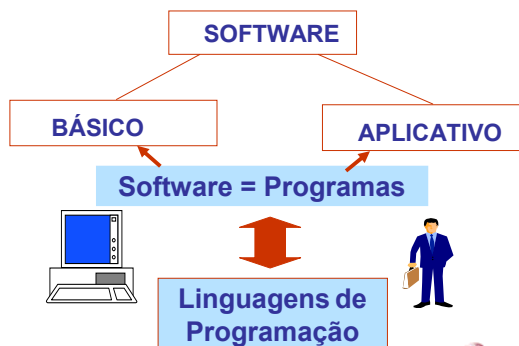


Slide 2

Profa. Cora H.F. Pinto Ribeiro



Programação

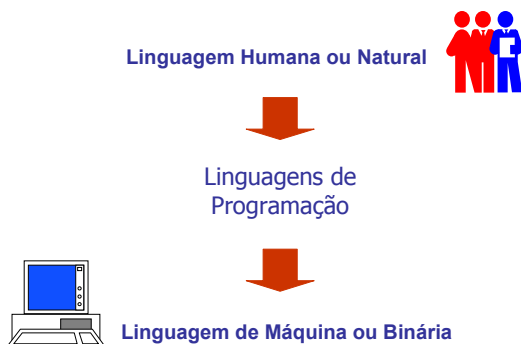


Slide 3

Profa. Cora H.F. Pinto Ribeiro



Linguagens de Programação



Slide 4

Profa. Cora H.F. Pinto Ribeiro



Modelos e Paradigmas de Linguagens

Objetivo: atender a diferentes domínios de problemas

Modelo imperativo

- ✓ Programa: seqüências de comandos que efetuam transformações sobre dados.
- ✓ Paradigma:
 - procedural
 - orientadas a objetos

Modelo declarativo

- ✓ Linguagens não possuem o conceito de seqüências de comandos.
- ✓ Paradigma:
 - funcional
 - lógico

Paradigmas de Linguagens de Programação

Modelo Imperativo

Paradigma Procedural (ou Imperativo):

- ✓ Solução implementada através de ações, executadas seqüencialmente.
- ✓ Conceitos: variáveis, atribuição, seqüenciação.
- ✓ Linguagens Procedurais: **C**, Pascal, Java, Algol, Fortran, PL/I, Basic, Ada.

Paradigma de Orientação a Objetos:

- ✓ Solução implementada através de objetos, que coexistem e trocam informações entre si.
- ✓ Conceitos: objetos, mensagem, herança e polimorfismo.
- ✓ Linguagem orientada a Objetos: Smalltalk, Java, C++, Pascal OO.

Slide 5

Profa. Cora H.F. Pinto Ribeiro



Slide 6

Profa. Cora H.F. Pinto Ribeiro



Paradigmas de Linguagens de Programação

Modelo Declarativo

Paradigma Funcional:

- ✓ Solução através de **mecanismos** de controle **recursivos**.
- ✓ Conceitos: função, comportamento, aplicação, avaliação
- ✓ Linguagens Funcionais ou Aplicativas: Lisp, Scheme, ML, Miranda, Haskell.

Paradigma Lógico:

- ✓ Solução implementada **regras de inferência** aplicadas a uma base de conhecimento.
- ✓ Conceitos: lógica simbólica, regras, relação, inferência.
- ✓ Linguagens Lógicas ou Declarativas: Prolog.

Slide 7

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

INF01202 - Algoritmos e Programação

- Paradigma: **Procedural**.

- Baseada nos princípios da **Programação Estruturada**.

- Linguagem de Programação: **C**.

Slide 8

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Programação Estruturada

- Se refere a técnicas de construção de soluções e de critérios para estruturar e codificar programas;
- Regula e limita:
 - estruturas de controle utilizadas;
 - forma de modularização do programa;
 - utilização de identificadores e comandos;
 - estrutura da codificação (alinhamentos, separação de comandos...);
 - documentação do programa.

Slide 9

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Programação Estruturada

Objetivos:

- ✓ Aumentar confiabilidade e legibilidade do programa;
- ✓ Minimizar a complexibilidade do programa;
- ✓ Estabelecer uma metodologia disciplinada de programação;
- ✓ Aumentar a produtividade do programador;
- ✓ Facilitar a manutenção dos programas.

Slide 10

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Programação Estruturada

Técnicas adotadas:

- ✓ Refinamento gradual da solução, associado a diferentes níveis de abstração;
- ✓ Projeto de solução modular: o problema se decompõem em etapas ou módulos hierárquicos;
- ✓ Programa composto de módulos básicos (blocos), cada um para uma tarefa específica;
- ✓ Programação *top_down* e *bottom-up*.
- ✓ Existe correspondência entre algoritmo e programa;

Slide 11

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Programação Estruturada

Estrutura de um programa:

- ✓ Eliminação das transferências incondicionais (**sem gotos**);
- ✓ Construção de **módulos** de tamanho adequado;
- ✓ Declaração do âmbito de atuação das variáveis globais (o programa inteiro) e locais (dentro das funções);
- ✓ Programa escrito de forma padronizada, incluindo **identificadores** utilizados, **endentação** (margens), **espaços em branco** e **comentários**, para facilitar a leitura, entendimento e manutenção do mesmo.

Slide 12

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Programação Estruturada

Estrutura de um módulo:

- ✓ Cada módulo:
 - **início** - topo - um único ponto inicial da execução.
 - **fim** - fim do bloco - um único ponto final da execução.
- ✓ Em cada módulo, o fluxo de execução de instruções:
 - **seqüência** - fluxo seqüencial: seqüência de ações, onde a saída de uma é seguida do início da próxima;
 - **seleção** - fluxo condicional: ações são ou não executadas, dependendo do resultado da avaliação de condições;
 - **iteração** - laços de repetição de ações, fluxo repetitivo: ações são repetidas um número determinado de vezes ou dependendo da avaliação de condições.

Slide 13

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Programa

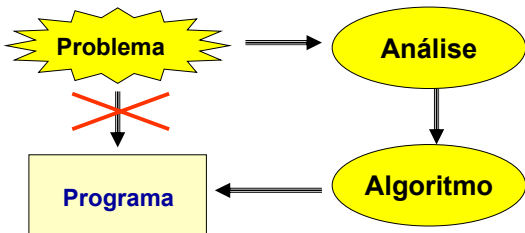
- Seqüência correta de instruções:
 - o que fazer
 - quando fazer
- Instrução:
 - ação **elementar** a ser **executada**.

Slide 14

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Elaboração de um Programa



Slide 15

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Processo de geração de um Programa

- Análise e Definição do Problema
 - Projeto do Algoritmo
 - Validação do Algoritmo (teste de mesa)
- Off Line**
- Tradução do Algoritmo para uma linguagem de programação (codificação)
 - Compilação
 - Teste e Depuração
 - Implementação
- On Line**
- Linguagem de Programação

Slide 16

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Processo de geração de um Programa

- **Análise e Definição do Problema**
 - Projeto do Algoritmo
 - Validação do Algoritmo (teste de mesa)
 - Tradução do Algoritmo para uma linguagem de programação (codificação e compilação)
 - Teste e Depuração
 - Implementação

Slide 17

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Processo de geração de um Programa

- ✓ **Análise e Definição do Problema:**
 - Ler atentamente o enunciado do problema, até entendê-lo bem.
 - Identificar os dados de entrada.
 - Identificar as saídas (resultados esperados).
 - O que o programa deve fazer (objetivo), isto é, como transformar as entradas em saídas?
 - Identificar se existem valores ou dados intermediários, usados para transformar as entradas nas saídas.
 - Pode ser dividido em subproblemas?

Slide 18

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Processo de geração de um Programa

✓ Análise e Definição do Problema

➤ Projeto do Algoritmo

- Validação do Algoritmo (teste de mesa)
- Tradução do Algoritmo para uma linguagem de programação (codificação e compilação)
- Teste e Depuração
- Implementação

Slide 19

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Algoritmo

Definição:

"Algoritmo é uma sequência finita e lógica de instruções ou passos básicos, especificados de acordo com uma determinada linguagem e que serve para resolver um determinado problema"

Paradigma

Slide 20

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Projeto do Algoritmo

✓ Determinar as ações que levam ao resultado:

- identificar as ações e a sequência destas ações para obter o resultado final (como transformar as entradas nas saídas desejadas).

✓ Construir o algoritmo:

- descrever os passos que levam à resolução do problema.

Slide 21

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

ALGORITMO

✓ Propriedades:

- possui um estado inicial;
- contém uma sequência lógica e finita de ações (comandos), claras e precisas, com fluxo de execução baseado em:
 - sequência;
 - seleção condicional (seleção de ações);
 - iteração (repetição de ações);
- possui dados de entrada;
- produz dados de saída corretos;
- possui estado final previsível;
- deve ser eficaz.

Programação Estruturada

Slide 22

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Normas importantes para algoritmos

- ✓ Identificar o algoritmo (identificador).
- ✓ Incluir com clareza, no topo do algoritmo, a finalidade do algoritmo e suas entradas e saídas.
- ✓ Usar apenas 1 comando por linha (substantivo + verbo).
- ✓ Usar indentação (reco de margens) para indicar o nível (hierarquia) de cada linha;
- ✓ Escolher nome significativos para variáveis e identificadores, mas que não sejam longos;
- ✓ Utilizar espaços e linhas em branco para maior legibilidade do algoritmo.
- ✓ Nunca utilizar desvios através de 'vai para' (*go to*).

Slide 23

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Formas de Representação de um Algoritmo

- Textual :
 - Pseudo-linguagem natural
- Gráfica :
 - Fluxograma
 - Diagramas de Chapin

Slide 24

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Análise e Definição do Problema

✓ Especificação do Problema

- Enunciado:

Trocar uma lâmpada queimada

Entradas: lâmpada queimada, lâmpada boa

Saída: lâmpada acesa

Objetivo: substituir uma lâmpada queimada por uma boa

- ✓ **Análise e Definição do Problema:**
 - Ler atentamente o enunciado do problema, entendê-lo bem.
 - Identificar os **dados de entrada**.
 - Identificar as **saídas** (resultados esperados).
 - O que o programa deve fazer (objetivo)?
 - **como transformar as entradas em saídas**
 - Identificar se existem **valores ou dados intermediários**, usados para transformar as entradas nas saídas.
 - Pode ser dividido em subproblemas?

Slide 25

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Algoritmo

- ✓ Algoritmo para trocar uma lâmpada queimada de uma luminária no teto:

Algoritmo Troca Lâmpada

{ substituição de uma lâmpada queimada, no teto, por outra boa,
entrada: lâmpada queimada e nova
saída: lâmpada acesa }

1. Início
2. Remove a lâmpada queimada
3. Coloca a lâmpada nova
4. Fim

Propriedades:

- ✓ possui um estado inicial;
- ✓ contém uma sequência lógica e finita de ações (comandos), claras e precisas, com fluxo de execução baseado em:
 - sequência;
 - seleção condicional (seleção de ações);
 - iteração (repetição de ações);
- ✓ possui dados de entrada;
- ✓ produz dados de saída corretos;
- ✓ possui estado final previsível; deve ser eficaz.

- Identificar o algoritmo (identificador).
- Incluir com clareza, no topo do algoritmo, a finalidade do algoritmo e suas entradas e saídas.
- Usar apenas 1 comando por linha.
- Usar indentação (recuo de margens) para indicar o nível (hierarquia) de cada linha.
- Escolha nomes significativos para variáveis e identificadores, mas que não sejam longos.
- Utilize espaços e linhas em branco para maior legibilidade do algoritmo.
- Nunca utilizar desvios através de 'vai para' (go to).

Slide 26

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Algoritmo

- ✓ Algoritmo para trocar uma lâmpada queimada de uma luminária no teto:

1. Início
2. Pega uma escada
3. Posiciona escada embaixo de lâmpada
4. Busca lâmpada nova
5. Sobe na escada até alcançar a lâmpada
6. Gira a lâmpada queimada no sentido anti-horário, até que se solte
7. Fixa a lâmpada nova, girando no sentido horário
8. Desce da escada
9. Guarda a escada
10. Fim

Lâmpada está queimada ?

Slide 27

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Algoritmo

- ✓ Algoritmo para trocar uma lâmpada queimada de uma luminária no teto:

1. Início
2. Pega uma escada
3. Posiciona escada embaixo de lâmpada
4. Aciona o interruptor
5. Se a lâmpada não acender, então
 - 5.1 Busca lâmpada nova
 - 5.2 Sobe na escada até alcançar a lâmpada
 - 5.3 Gira a lâmpada queimada no sentido anti-horário, até que se solte
 - 5.4 Fixa a lâmpada nova, girando no sentido horário
 - 5.5 Desce da escada
6. Guarda a escada
7. Fim

Slide 28

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Algoritmo

✓ Propriedades:

- possui um estado inicial;
- contém uma sequência lógica e finita de ações (comandos), claras e precisas, com fluxo de execução baseado em:
 - sequência;
 - seleção condicional (seleção de ações);
 - iteração (repetição de ações);
- possui dados de entrada;
- produz dados de saída corretos;
- possui estado final previsível;
- **deve ser eficaz.** ↔ **habilidade cognitiva**

Slide 29

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Algoritmo

- ✓ Algoritmo para trocar uma lâmpada queimada de uma luminária no teto:

1. Início
2. Aciona o interruptor
3. Se a lâmpada não acender, então
 1. Pega uma escada
 2. Posiciona escada embaixo da lâmpada
 3. Busca lâmpada nova
 4. Sobe na escada até alcançar a lâmpada
 5. Gira a lâmpada queimada no sentido anti-horário, até que se solte
 6. Fixa a lâmpada nova, girando no sentido horário
 7. Desce da escada
 8. Guarda a escada
4. Fim

Lâmpada nova está queimada ?

Slide 30

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Algoritmo

- ✓ Algoritmo para trocar uma lâmpada queimada de uma luminária no teto:

1. Início
2. Aciona o interruptor
3. Se a lâmpada não acender, então
 1. Pega uma escada
 2. Posiciona escada embaixo da lâmpada
 3. Busca caixa com lâmpadas novas
 4. Sobe na escada até alcançar a lâmpada
 5. Enquanto a lâmpada não acender, faça:
 1. Gira a lâmpada queimada no sentido anti-horário, até que se solte
 2. Fixa a lâmpada nova, girando no sentido horário
 6. Desce da escada
 7. Guarda a escada
4. Fim

Slide 31

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Processo de geração de um Programa

- ✓ Análise e Definição do Problema
- ✓ Projeto do Algoritmo
 - Validação do Algoritmo (teste de mesa)
 - Tradução do Algoritmo para uma linguagem de programação (codificação e compilação)
 - Teste e Depuração
 - Implementação

Slide 32

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Análise e Definição do Problema

Especificação do problema:

- ✓ Enunciado:
 - Ler um valor, informado através do teclado, que pode ser igual a 1 ou a 2.
 - Se o valor lido for 1, informar 2.
 - Se o valor lido for 2, informar 1.

Objetivo: ler um valor e informar outro valor alternativo.

Entradas: 1 ou 2

Saída: 2 ou 1

Slide 33

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Projeto do Algoritmo

Algoritmo Alternale2

Objetivo: ler valor;
se = 1, informa 2;
se = 2, informa 1.

Entradas: 1 ou 2
Saídas: 2 ou 1

1. Início
2. Lê x
3. Se x = 2
 - 3.1 então x ← 1
4. Se x = 1
 - 4.1 então x ← 2
5. Informa x
6. Fim

Slide 34

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Validação do Algoritmo - Teste de Mesa

- ✓ Algoritmo deve ser **correto**:

Algoritmo Alternale2

Objetivo: ler valor;
se = 1, informa 2;
se = 2, informa 1.

Entradas: 1 ou 2
Saídas: 2 ou 1

1. Início
2. Lê x
3. Se x = 2
 - 3.1 então x ← 1
4. Se x = 1
 - 4.1 então x ← 2
5. Informa x
6. Fim

- ✓ Contexto: x=1 ou x=2 (garantido!)
- ✓ Dica - simular execução:
 - 1º valor válido
 - último valor válido
 - 1º valor inválido inferior
 - 1º valor inválido superior
- ★ Erro - valor informado sempre = 2
- ✓ Vacina para este tipo de erro -
 - nunca alterar diretamente valores lidos¹
 - utilizar variáveis diferentes para armazenar resultados¹
- ★ Usar mnemônicos¹ (identificadores)!

¹ Programação Estruturada

Slide 35

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Validação do Algoritmo - Teste de Mesa

- ✓ Algoritmo deve ser **eficiente**:

Algoritmo Alternale2

Objetivo: ler valor;
se = 1, informa 2;
se = 2, informa 1.

Entradas: 1 ou 2
Saídas: 2 ou 1

1. Início
2. Lê val_lido
3. Se val_lido = 2
 - então val_inf ← 1
4. Se val_lido = 1
 - então val_inf ← 2
5. Informa val_inf
6. Fim

- ✓ Contexto: x=1 ou x=2 (garantido!)
- ✱ Utilização de nomenclatura significativa - **mnemônicos!** (val_lido e val_inf)
- ☹ Limitação:
 - redundância de operações (se x = 2, então, obviamente, x ≠ 1, e vice-versa)
- ✓ Dica - evitar operações repetidas ou redundantes

Slide 36

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Validação do Algoritmo - Teste de Mesa

- ✓ Algoritmo deve ser **eficiente**:

Algoritmo Altern1e2

Objetivo: ler valor;
se = 1, informa 2;
se = 2, informa 1.

Entradas: 1 ou 2
Saídas: 2 ou 1

```
1. Início
2. Lê val_lido
3. Se val_lido = 2
   então val_inf ← 1
   senão val_inf ← 2
4. Informa val_inf
5. Fim
```

- ✓ Contexto: x=1 ou x=2 (garantido!)
- ✱ Eficiência melhorada
- ✓ Dica – eliminar repetição de teste de condições mutuamente exclusivas

Slide 37

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Validação do Algoritmo - Teste de Mesa

- ✓ Algoritmo deve ser **facilmente entendido**:

Algoritmo Altern1e2

Objetivo: ler valor;
se = 1, informa 2;
se = 2, informa 1.

Entradas: 1 ou 2
Saídas: 2 ou 1

```
1. Início
2. Lê val_lido
{alterna valores entre 1 e 2}
3. val_inf ← 3 - val_lido
4. Informa val_inf
5. Fim
```

- ✓ Contexto: x=1 ou x=2 (garantido!)
- ✱ Código muito eficiente
- ☹ Perigo - solução não trivial
- ✓ Vacina -
 - documentar bem um programa
 - informar o que faz (e não como)

Slide 38

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Processo de geração de um Programa

- ✓ Análise e Definição do Problema
- ✓ Projeto do Algoritmo
- ✓ Validação do Algoritmo (teste de mesa)
- Tradução do Algoritmo para uma linguagem de programação (codificação e compilação)
- Teste e Depuração
- Implementação

Próxima aula!

Slide 39

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Avaliação da Disciplina

- Presença controlada aula à aula !
- ✓ Material e tarefas:
 - página da disciplina.

<http://moodle.inf.ufrgs.br/>

Slide 40

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Avaliação da Disciplina

$$\text{Média Final} = 0,3 \cdot P1 + 0,45 \cdot P2 + 0,25 \cdot E$$

Prova 1

Prova 2

Exercícios
+
Trabalho*

* pré-requisito para
conceito final A

Datas das provas

24/set - Prova 1
17/nov - Prova 2
26/nov - Recuperação

Slide 41

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática