Programação de Computadores I

Aula 04

O Processo de Programação

José Romildo Malaquias

Departamento de Computação
Universidade Federal de Ouro Preto

2011-1

Algoritmo

- Um algoritmo é uma **seqüência de instruções** bem definidas para a **solução de um problema** em um **tempo finito**.
- Um algoritmo não é a solução de um problema, é o caminho para a solução.
- Todo algoritmo deve satisfazer:
  - **Entrada**: zero ou mais valores de entrada
  - **Saída**: pelo menos um valor deve ser produzido
  - **Claridade**: toda instrução deve ser clara e não ambígua
  - **Término**: o algoritmo deve terminar depois de um número finito de passos
  - **Efetividade**: toda instrução deve ser factível

Programação I

- **Programação:** ato de escrever um algoritmo em alguma linguagem de programação.
- **Programas** são **formulações concretas** de **algoritmos abstratos**, baseados em **representações e estruturas** específicas de dados

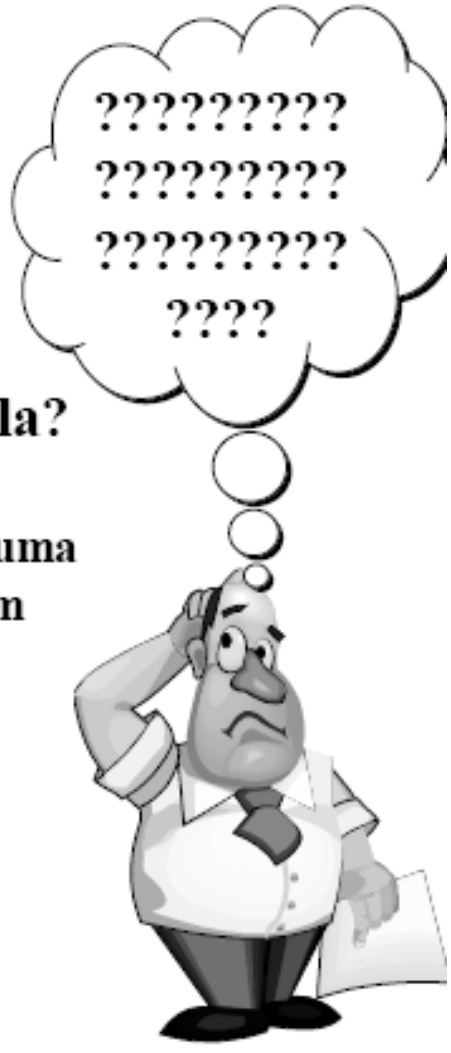
Algoritmos

Problema

Como desenvolver uma solução?

Como expressá-la?

Através de uma linguagem



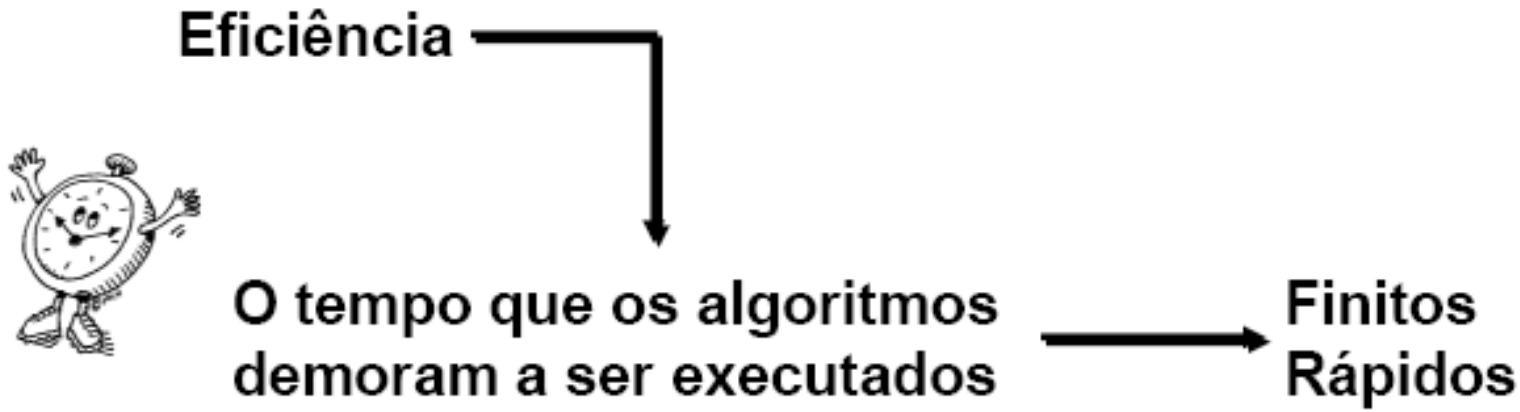
Algoritmo

Método de resolução de um dado problema expresso em linguagem natural.

Concretização de um algoritmo numa dada linguagem de programação.

Programa

- Características:

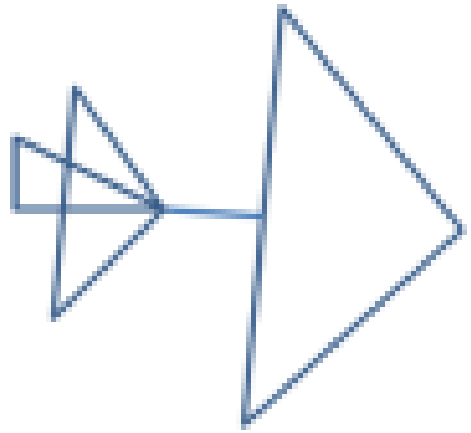


Abstração I

- **Abstração:** está associada à **remoção dos detalhes** de algo ou alguma coisa, com o intuito de **concentrar foco** em suas **características mais importantes.**
- Os dados ou problemas a serem processados pelo computador representam uma abstração de parte da realidade.
- A abstração sugere a distinção que deve ser feita entre **o que** o programa faz e **como** ele é implementado.

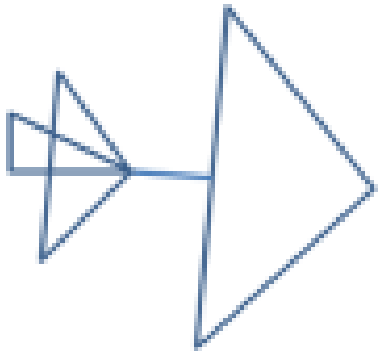
Abstração II

Concreto: todos os detalhes da realidade



Abstrato: só características importantes

Abstração III



Quanto mais **abstrato** mais geral.

A figura ao lado poderia representar qualquer avião.

Quanto mais **concreto**, mais específico

O avião ao lado é tão específico, com tantos detalhes, que só representa ele mesmo

Desenvolvimento de software

- Etapas do desenvolvimento de um programa:
  - Análise
    - Estudar o enunciado do problema;
    - Definir os dados de entrada, o processamento e os dados de saída.
  - Algoritmo
    - Descrever o problema com suas soluções.
  - Codificação
    - Expressar o algoritmo numa linguagem de programação escolhida.

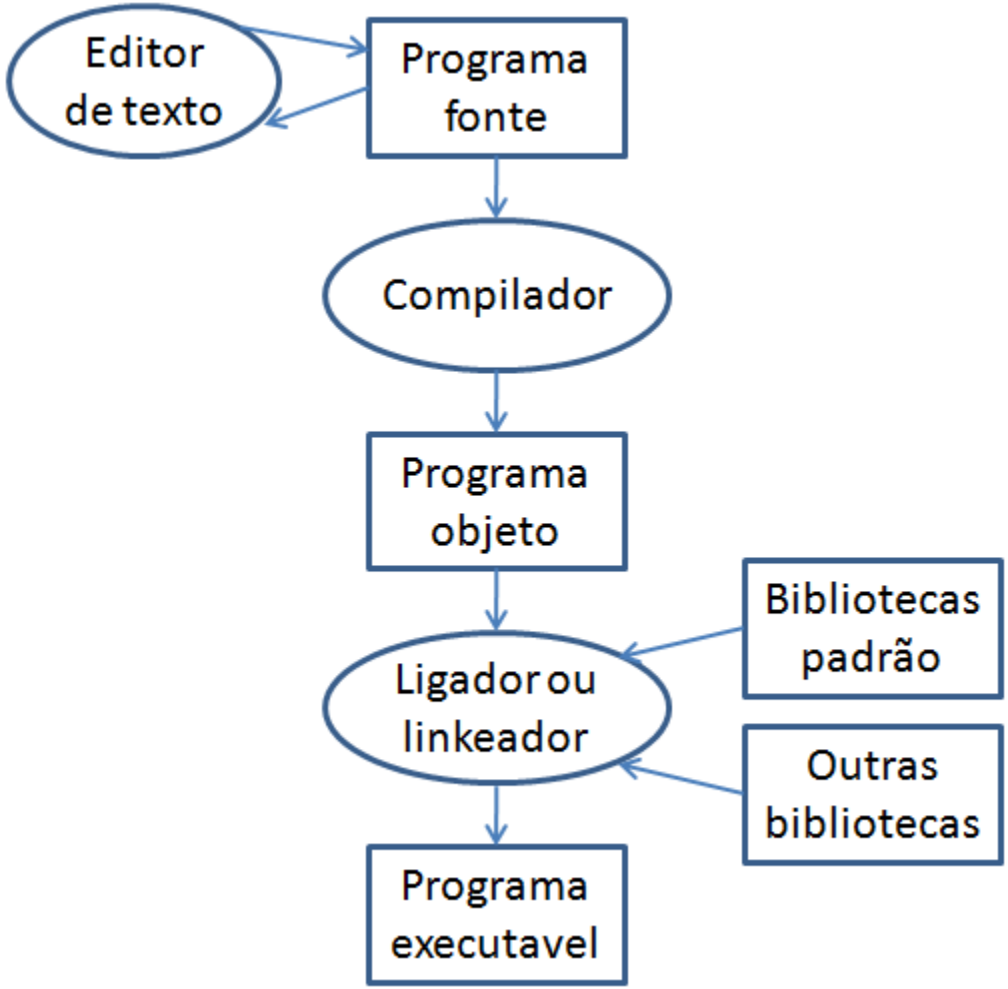
Processo de programação I

- Inicia com a edição de um **programa-fonte** e termina com a geração de um **programa-executável**.
- Passos:
  1. O programa-fonte é criado em um **editor** de textos e gravado em um arquivo com extensão c.
  2. O **compilador** analisa o código-fonte e o converte para um código-objeto (versão em linguagem de máquina do programa-fonte) que é gravado em um arquivo com extensão o ou obj, dependendo do sistema operacional.
  3. Se o programa contiver chamadas a funções das bibliotecas (função cosseno, por exemplo) o **ligador** (*linker*) junta o programa-objeto com a(s) respectiva(s) biblioteca(s) e gera um código-executável, que é gravado em um arquivo com extensão .exe no Windows, ou sem extensão, mas com permissão de execução em sistemas Unix.

Processo de programação II

- Edição de ligações: (*link-edição*)
  - Programas em C tipicamente contém chamadas a funções definidas em outros locais, tais como as bibliotecas padrões ou bibliotecas de um projeto particular
  - O código-objeto produzido contém *buracos* devido a essas chamadas
  - O *linker* liga o código-objeto com o código dessas chamadas para produzir o código-executavel.

Processo de programação III



Processo de programação IV

- Principais termos:
  - **Código-fonte**: contem os comando da linguagem de programação
  - **Código-objeto**: criado pela conversão do código-fonte em linguagem de máquina. É gerado pelo compilador.
  - **Ligador** ou ***linkeditor***: junta o código-objeto com as bibliotecas necessárias para gerar o programa-executável.

Processo de programação V

- **Programa executável**: código que pode ser executado pelo sistema operacional.
- **Tempo de compilação**: durante o processo de conversão entre código-fonte e código-objeto.
- **Tempo de execução**: após a ativação do programa executável.

Pre-processamento I

- Em um sistema desenvolvido em C, o **pré-processador** é executado automaticamente antes do compilador
  - Obedece comandos especiais chamados de **diretivas de pré-processamento**.
  - Indicam que manipulações devem ser realizadas no texto do programa antes da compilação (inclusão de outros arquivos no código-fonte, por exemplo).
  - Todas as diretivas começam com #
  - Diretivas do pré-processador não são comandos C, assim elas **não** terminam com "**;**"

Pre-processamento II

- Diretivas de inclusão de arquivo em C:
  - **`#include`** `<stdio.h>`
    funções de entrada/saída
  - **`#include`** `<math.h>`
    funções matemáticas

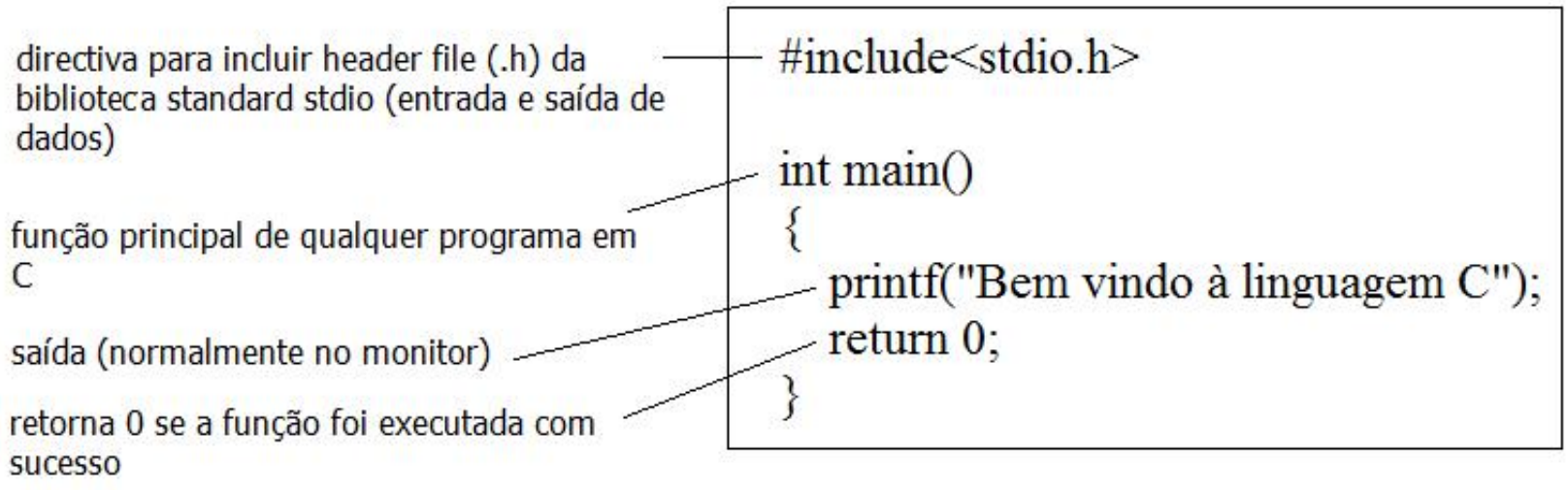
Primeiro programa em C I

```c
/* primeiro programa em C */

#include <stdio.h>

int main(void)
{
    printf("Bem vindo à linguagem C!\n");
    return 0;
}
```

Primeiro programa em C II



Primeiro programa em C III

1. Digite o texto do programa-fonte em um editor de texto e salve-o no arquivo ola.c. Alguns bons editores de texto são:
   - notepad++ (Windows)
   - gedit (Linux)
   - editor do ambiente de desenvolvimento Code::Blocks
2. Compile o programa-fonte para obter o programa executável.
   - Na linha de comando, usando o compilador gcc:

```
$ gcc ola.c -o ola
```

   - Usando o ambiente de desenvolvimento Code::Blocks

   O ligador é executado indiretamente pelo próprio comando que que faz a compilação.
3. Execute o programa
   - Na linha de comando:

```
$ ./ola
```

   - Usando o ambiente de desenvolvimento Code::Blocks

Estrutura básica de um programa em C I

```
<diretivas do pré-processador>

<declarações globais>

int main(void)
{
    /* <comentário
       de bloco> */
    <declarações locais>   // <comentário de linha>
    <instruções>
    return <status>;
}
```

Estrutura básica de um programa em C II

- As diretivas de pré-processamento tipicamente são para incluir arquivos de cabeçalho (*header files*) de bibliotecas.
- Um programa em C é formado por várias funções que podem ser chamadas umas pelas outras (assunto a ser estudado mais adiante). **main** é a função principal do programa. É o ponto de entrada para a execução do programa. Quando pedimos ao sistema operacional para executar o programa, ele começa pela execução das instruções colocadas no corpo da função `main`.
- A instrução **return** deve ser a última instrução em `main` e encerra a execução do programa. Ela informa ao sistema operacional um código de *status*, geralmente indicando se o programa funcionou corretamente (valor zero) ou não (valor diferente de zero).

Estrutura básica de um programa em C III

- **Comentário** é um segmento do programa que é completamente ignorado pelo compilador, servindo para documentar o programa quando alguém o lê. O comentário pode ser:
  - de bloco: delimitados pelas sequências de caracteres `\*` e `*/`, pode incluir várias linhas do texto do programa
  - de linha: começa com a sequência de caracteres `//` e termina no final da linha
- Na linguagem C letras maiúsculas e minúsculas fazem diferença ao escrever o texto do programa. Assim os nomes `main` e `Main` são considerados diferentes.

FIM

Créditos:
Baseado no material preparado pelo
Prof. Guillermo Cámara-Chávez.