Programação Aplicada I

Prof. Thiago Pavão Marquesini

contato@marquesini.com.br

Sobre programação

O que programadores fazem?

- Em geral, os trabalhos de programação são converter soluções de problemas em instruções para o computador. Isto é, o programador prepara as instruções de um programa de computador e executa as instruções no computador, testa o programa para verificar se está funcionando corretamente e faz correções quando necessárias.

- Também cabe ao programador documentar os processos que levaram até a finalização do programa. As formas e métodos de documentação podem variar conforme a empresa ou o projeto e se fazem necessárias para facilitar possíveis manutenções nos programas ou organizar o trabalho em equipe.

O processo da programação

O desenvolvimento de um programa envolve passos, similar a qualquer tarefa de resolver um problema.

No processo de programação, podemos definir cinco passos principais, que são:

- Definir o problema
- Planejar a solução
- Codificar o programa
- Testar o programa
- Documentar o programa

Definindo o problema

- Suponha que você, como programador, seja contratado por uma empresa. Você deve se reunir com os envolvidos para que descrevam o projeto.
- Especificamente, a tarefa de definir o problema consiste em reunir o que você sabe (dados fornecidos) e o que você deseja obter (saída – o resultado).
- Eventualmente, você documenta por escrito, entre outras coisas, o tipo de entrada, processamento e saídas necessários.

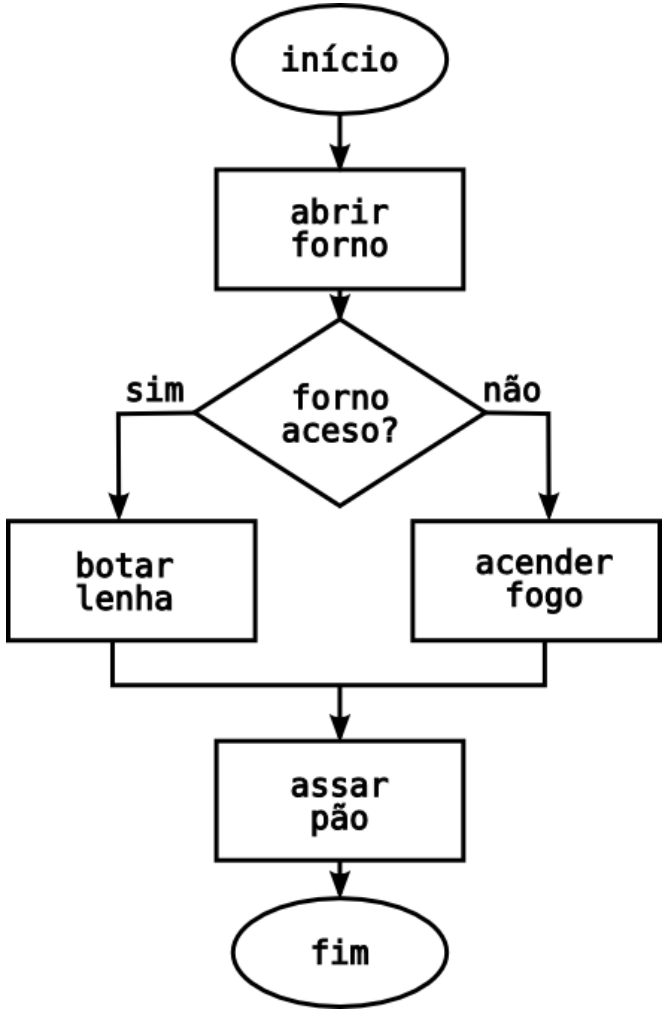
Planejando a solução

- Duas maneiras mais comuns de planejar uma solução são desenhar um fluxograma e escrever pseudocódigo, ou possivelmente ambos.

- Um fluxograma é representação por desenhos de uma solução passo-a-passo para um problema. Consiste em setas representando a direção do programa e caixas e outros símbolos que representam ações. É um mapa do que seu programa vai fazer e como vai ser feito.

- O pseudocódigo é uma forma genérica de declarar sua solução sem precisar se preocupar com a sintaxe de uma linguagem de programação, se concentrando na lógica envolvida. Não é executável no computador.

Planejando a solução - Fluxograma



Planejando a solução - Pseudocódigo

Algoritmo 1 Exemplo de Pseudocódigo.

```
leia (x, y) {Esta linha é um comentário}
se x > y então
    escreva ("x é maior")
senão
    se y > x então
        escreva ("y é maior")
    senão
        escreva ("x e y são iguais")
    fim-se
fim-se
```

Codificando o programa

- O próximo passo é codificar o programa, ou seja, expressar sua solução em linguagem de programação. Neste momento o programador traduz o pseudocódigo ou o fluxograma para uma linguagem de programação.
- Uma linguagem de programação é um conjunto de regras que fornece uma maneira de instruir o computador sobre quais operações executar.
- Para que um programa funcione corretamente, você precisa seguir exatamente a sintaxe (regras) da linguagem que está usando.
- Os códigos podem ser escritos inicialmente em papel ou diretamente em um editor de textos simples ou editores de códigos específicos.

Testando o programa

Eventualmente, após codificar seu programa, é importante testá-lo no computador. Algumas técnicas de teste são:

- Teste de mesa ou checagem de mesa: consiste em verificar a lógica do programa para garantir que este está livre de erros e funcionando. É recomendado que outra pessoa além do programador do código realize a verificação em busca de erros ou sugestões.

- Depuração: termo usado extensivamente em programação, significa detectar, localizar e corrigir erros, geralmente executando o programa. Nesta fase você executa o programa com dados de testes e verifica os resultados obtidos se satisfazem os objetivos do programa.

Documentando o programa

- Documentação é uma descrição detalhada do ciclo de programação e fatos específicos do programa. É necessária para suplementar a memória humana e ajudar na organização, planejamento e desenvolvimento do programa.

- A documentação pode conter, mas não se limitando a, origem e natureza do problema, breve descrição narrativa do programa, ferramentas lógicas como pseudocódigos e fluxogramas, descrições de registros de dados e resultados de testes.

- Comentários no próprio código do programa também são considerados parte essencial da documentação.

Paradigmas de programação

Paradigmas de programação

- Estruturada

  Tem ênfase no uso de estruturas em blocos, condicionais, laços de repetição (iterações) e uso de sub-rotinas de maneira simples e linear. É o paradigma mais simples e ainda largamente utilizado.

- Orientada a objetos

  Modelo de análise, projeto e programação de software baseado na interação entre unidades chamadas objetos, que possuem atributos e métodos. Um pouco mais complexa que a programação estruturada, permite uma maior abstração das soluções.

- Orientada a eventos

  O controle do fluxo dos programas são guiados por indicações externas, chamadas eventos. Mais aplicada no desenvolvimento de sistemas de interfaces com o usuário.

Linguagens de programação

Níveis de linguagem

- 1 - Linguagem de máquina

  Nível mais baixo de linguagem, representa dados e instruções dos programas através de binários (0 e 1) correspondentes aos estados elétricos ligado e desligado no computador. De difícil compreensão e manipulação humana, demanda um grande esforço. Cada computador tem sua própria linguagem de máquina.

- 2- Linguagem de baixo nível (montagem)

  Linguagens *assembly*, foram consideradas um grande avanço por utilizar códigos mnemônicos, abreviações fáceis de lembrar. A para *Add*, C para *Compare*, MP para *Multiply*, STO para armazenar informações na memória e assim por diante. Além disso, as linguagens *assembly* permitem a utilização de rótulos para locais de memória em vez de números de endereços reais. Assim como as linguagens de máquina, cada computador tem sua própria linguagem de montagem.

Níveis de linguagem – Parte 2

- 3 - Linguagem de alto nível

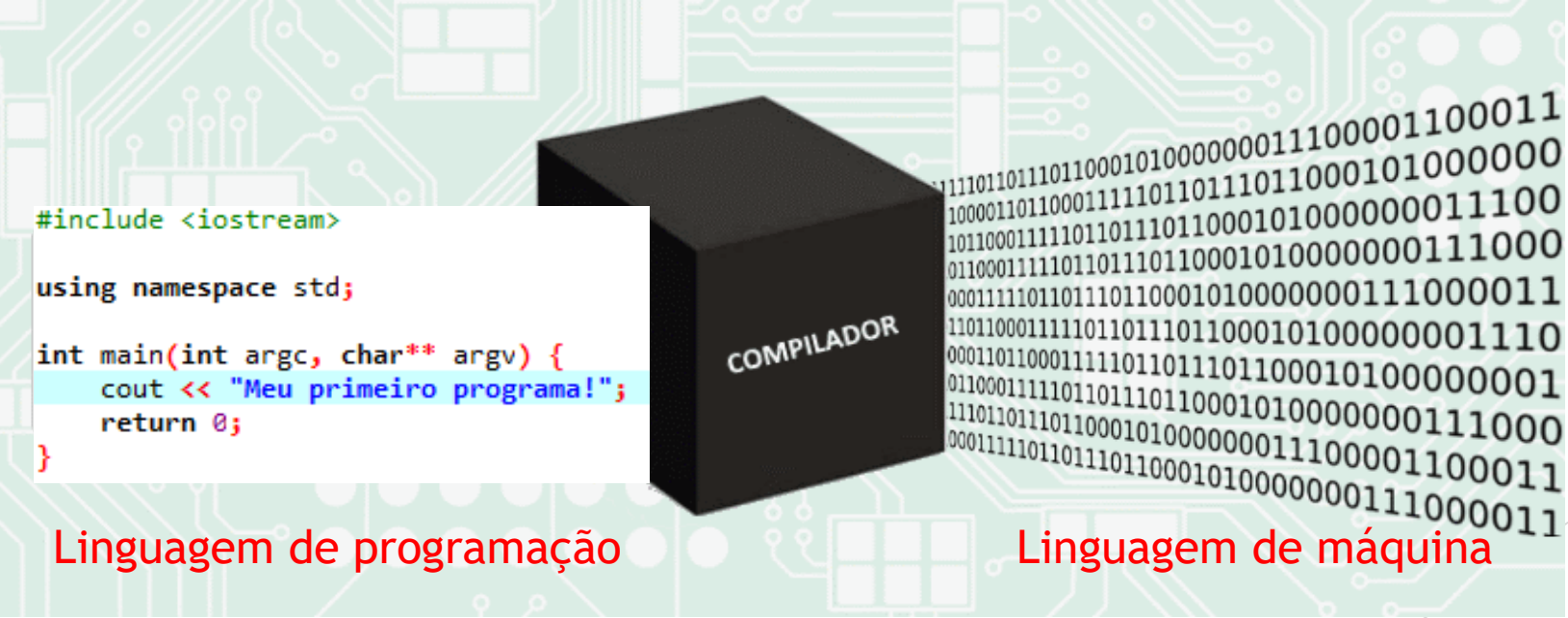
  Programas escritos de maneira semelhante ao inglês, tornando-os mais convenientes de usar. Como resultado, um programador consegue realizar mais com menos esforço e os programas agora podem trabalhar tarefas muito mais complexas.

- 4 - Linguagem natural

  Linguagem que usamos no dia-a-dia como humanos: português, inglês, espanhol, italiano, alemão, francês e etc.

Linguagens de alto nível

- Necessita de um tradutor para traduzir as declarações simbólicas de uma linguagem de alto nível em linguagem de máquina que possa ser executada por um computador. Este tradutor é chamado de compilador.



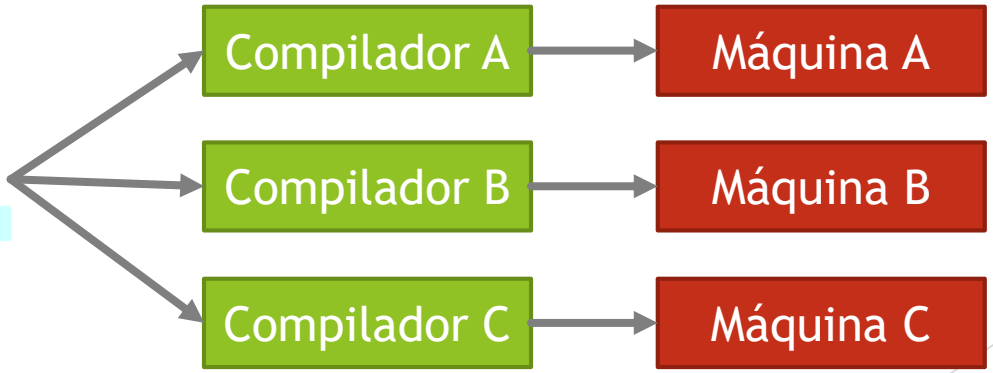
Linguagens de alto nível

- Como cada computador possui sua própria linguagem de máquina, um mesmo código de um programa precisa ser compilado conforme o computador de destino, ou seja, um mesmo programa pode funcionar em diferentes computadores, desde que tenha sido compilado (traduzido) para cada um deles.

```
#include <iostream>

using namespace std;

int main(int argc, char** argv) {
    cout << "Meu primeiro programa!";
    return 0;
}
```



Revisando

Passos da programação

- Definir o problema
- Planejar a solução
- Codificar o programa
- Testar o programa
- Documentar o programa

Programação estruturada

- Paradigma linear e simples que usa estruturas em blocos, condicionais, laços de repetição e sub-rotinas.

Linguagem de alto nível

- Linguagem de programação mais próxima da linguagem natural, permitindo escrever programas de maneira mais fácil e produtiva que em linguagens de baixo nível ou de máquina.

Compilador

- Converte um código fonte escrito em uma determinada linguagem de programação para linguagem de máquina compreensível para determinado computador.

Obrigado!

Prof. Thiago Pavão Marquesini

contato@marquesini.com.br