

# Laboratório de Programação I

---

- Conceitos básicos
- Estrutura de um programa em C

# Introdução

---

- Processamento de dados
  - **Sequência** de operações aplicadas a um conjunto de dados (**entrada**) p/ se obter um conjunto de dados/resultados (**saída**)
    - Ex.: Compensação bancária, Folha de pagamento, ...
  - Elementos:
    - Dados **iniciais**: informações sujeitas a transformações
    - **Transformações**: modificações realizadas no conteúdo/forma dos dados iniciais
    - Resultados  **finais**: produto dos dados iniciais após as transformações



# Programação - conceitos relacionados

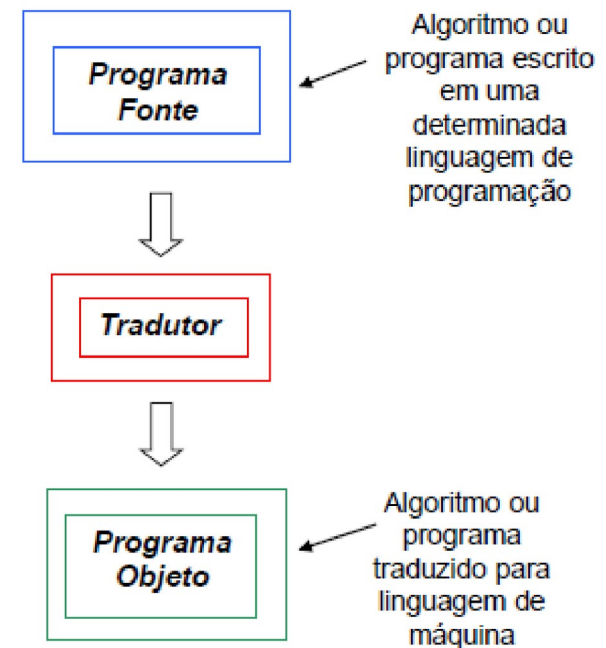
---

- **Algoritmo**
    - sequência finita de instruções, logicamente ordenadas para resolver um problema
  - **Programa**
    - Roteiro que orienta o computador, mostrando-lhe a sequência de operações necessárias para executar uma determinada tarefa
  - **Linguagem de programação**
    - Conjunto de regras (sintaxe) que especificam como instruir o computador a executar determinada tarefa
-

# Tradutor

---

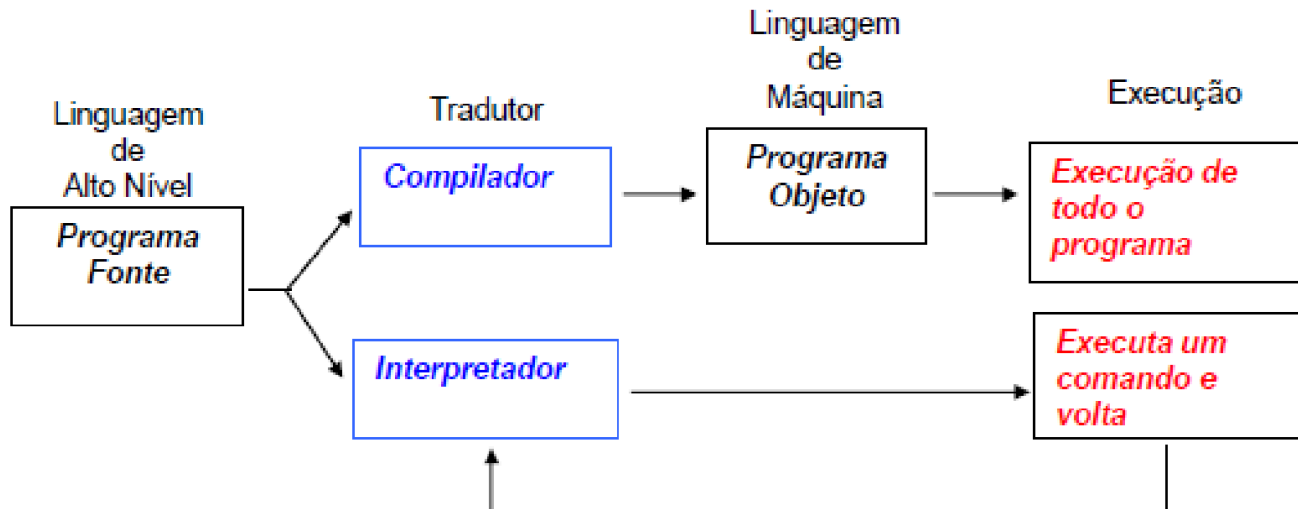
- Os computadores só podem executar diretamente os algoritmos expressos em linguagem de máquina (LM)
  - **Linguagem de máquina** = conjunto de instruções que ativam diretamente os dispositivos eletrônicos do computador
- Tradutor: programa que **traduz** um algoritmo escrito em **uma LP em LM**



# Tradução

---

- Compilação: lê, analisa e traduz todos os comandos do programa fonte, criando o programa objeto
- Interpretação: traduz ou interpreta cada comando ao executá-lo



# IDE (Integrated Development Environment) – Ambiente Integrado de Desenvolvimento

---

- Conjunto de ferramentas integradas com o objetivo de desenvolver programas
  - **Editor de programa:** específico para edição de código fonte
  - **Compilador:** Traduz código fonte para código objeto
  - **Linkeditor:** Linka o código objeto às bibliotecas, resultando no programa executável
  - **Loader:** Carrega o programa executável na memória para que o processador possa rodá-lo
  - **Debugger:** auxilia na verificação e correção de erros do programa

# Ambientes de Programação

---

- gcc (nativo do Linux)
- Dev-C++ <http://www.bloodshed.net/devcpp.html>
- Code::Blocks <http://www.codeblocks.org/>
- Eclipse <https://www.eclipse.org/cdt/>
- Netbeans <https://netbeans.org/features/cpp/index.html>
- Visual Studio <https://code.visualstudio.com/>
- Atom <https://atom.io/>
- Sublime <https://www.sublimetext.com/>
- KDevelop <https://www.kdevelop.org/>

# Linguagem C – histórico e características

---

- Década de 70
- Objetivo: desenvolvimento do Unix
- Padronização (1980): ANSI
- Simples
- Fácil aprendizado
- Estruturada (base = funções)
- Compilada



# Programa em C

---

- Composto por funções
  - Funções de biblioteca (pré-definidas)
  - Funções do programador
- Função *main* é obrigatória (principal)
  - 1ª a ser executada
  - Início – 1ª linha do main
  - Fim – última linha do main

# Estrutura de um programa em C

---

```
#include <stdio.h>    // biblioteca de I/O

                        //definição de constantes, variáveis e funções

int main(){           // início da função principal

                        // declarações e comandos

    return 0;
}
```

# Primeiro programa

---

|   |  |   |                            |
|---|--|---|----------------------------|
| 1 | <code>#include &lt;stdio.h&gt;</code>  | → | Biblioteca de E/S          |
| 2 |  |   |                            |
| 3 | ▼ <code>int main(){</code>             | → | Início da função principal |
| 4 | <code>printf("Hello, world\n");</code> | → | Escrita na saída padrão    |
| 5 | <code>return 0;</code>                 |   |                            |
| 6 | <code>}</code>                         |   |                            |
| 7 |  |   |                            |

- Execução de instruções: sequencial
- Separador de comandos → ;
- Case sensitive (idade != Idade != IDADE)
- Delimitadores de bloco { }
- Indentação

# Bibliotecas

---

`#include <stdio.h>    // funções de E/S`

- Avisa ao compilador que o programa irá usar procedimentos, funções e variáveis declaradas no arquivo `stdio.h`
  - Outras bibliotecas:
    - `math.h`    // funções matemáticas
    - `stdlib.h`    // funções de propósito geral
    - `string.h`    // funções p/ manipulação de strings
    - ...
-

# Comentários

---

- ignorados pelo compilador
- entendimento/lembretes no código
- não podem ser aninhados

// este comentário ocupa apenas 1 linha

/\* este comentário pode ocupar várias linhas,  
portanto possui indicação de início e indicação  
de término \*/

# Exercícios:

---

- 1) Edite o programa do slide 11 (programa que imprime Hello World), compile e teste.
- 2) Adapte o programa anterior de forma que ele imprima o seu nome.
- 3) Com base no exercício 2, faça um programa que imprima o seu nome completo, com uma palavra por linha. Utilize quantos comandos printf forem necessários.
- 4) Com base no exercício 3, faça um programa que imprima o seu nome completo, com uma palavra por linha. Utilize apenas 1 comando printf.

Para aula de hoje, acessar:

<https://ideone.com/>

<https://www.codechef.com/ide>

<https://www.onlinegdb.com/>

# Tarefa para casa:

---

Para as próximas aulas, acessar:

<http://linguagemc.com.br/ides-e-compiladores-c/> com os seguintes objetivos:

- Conhecer as diversas ferramentas disponíveis
- Instalar uma ferramenta (IDE) e aprender suas funcionalidades (com o auxílio de tutoriais e vídeos)