

Sistema FIEB



PELO FUTURO DA INOVAÇÃO

MIGRANDO PARA O C/C++

Introdução à Lógica de Programação

Prof. Lucas Amparo Barbosa

Semestre letivo 2020.2

O QUE É UMA LINGUAGEM DE PROGRAMAÇÃO?

- Sistemas de comunicação estruturado, composto por conjuntos de símbolos, palavras-chave, regras semânticas e sintáticas que permitem ao programador informar ao computador o que deseja que seja executado
 - Estruturado: Significa que tem uma organização, uma ordem;
 - Símbolos: `>`, `<`, `=`, `AND`, `OR`, `&&`, `||`, etc...
 - Palavras-chave: `for`, `if`, `continue`, `repeat`, etc...
 - Regras Semânticas: A ordem como as palavras são apresentadas
`se (qualquer_coisa > 0) entao`
 - Regras Sintáticas: A combinação das palavras
`se (qualquer_coisa > 0) e (nao_executado) entao`

O QUE É UMA LINGUAGEM DE PROGRAMAÇÃO?

- Existem dois tipos básicos
 - Baixo nível
 - Praticamente manipulando energia elétrica;
 - Muito rápida, por não haver conversão;
 - Assembly é um ótimo exemplo.
 - Alto nível
 - Utiliza, em geral, linguagens conhecidas pelo programador;
 - Precisa ser convertida para linguagem de máquina em algum momento.
- Existem três paradigmas principais
 - Estruturada
 - Sequencial, utilizando decisões e repetições para executar tarefas;
 - Muito utilizada ainda hoje, principalmente para protótipos rápidos;
 - Modular
 - Utiliza o conceito de módulos (blocos de código) para agrupar funcionalidades;
 - É possível adaptar códigos estruturados para funcionar de forma modular;
 - Orientada a Objetos
 - Organiza os blocos do código em estruturas chamadas de Objetos;
 - Facilita a compreensão por tentar representar o mundo real dentro dos códigos;

COMO ESCOLHER UMA LINGUAGEM?

- Existem diversas linguagens, com vantagens e desvantagens.
 - PHP, C, C++, Python, Java, Perl, etc...
- O que diferencia cada uma?
 - Objetivo
 - Existem linguagens especializadas em determinados processos
 - PHP é uma linguagem para arquitetura cliente-servidor
 - Java é ótima para programação multiplataforma
 - Prolog e Lisp foram desenvolvidas para Inteligência Artificial
 - Existem linguagens especializadas em paradigmas
 - Assembly é o mais próximo da linguagem de máquina possível
 - C++ permite orientação a objetos
 - Desempenho
 - Java é naturalmente lenta, para permitir a portabilidade
 - Python é mais lenta que o normal pois converte o código para C
 - C é mais rápido que C++ pois não suporta orientação a objetos

COMO ESCOLHER UMA LINGUAGEM?

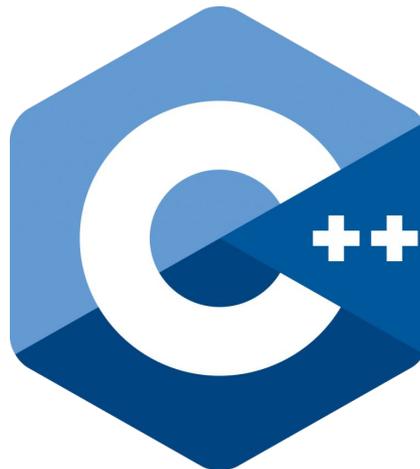
- O que diferencia cada uma?
 - Tipagem
 - Linguagens fortemente tipadas não permitem a mudança dinâmica dos tipos de dados
 - Linguagens fracamente tipadas habilitam essa mudança, dando uma maior liberdade ao programador
 - Comunidade
 - Linguagens amplamente utilizadas, como C/C++ e Python, possuem mais exemplos, tutoriais, cursos e bibliotecas disponíveis na Internet, agilizando assim o aprendizado
 - Frameworks
 - São pacotes de códigos disponíveis para determinada linguagem, agilizando o desenvolvimento
 - SciKit Learn para Python agiliza o uso de Aprendizado de Máquina
 - Keras agiliza o uso de Deep Learning para C++ e Python
 - PrimeFaces agiliza o desenvolvimento de GUI para Java
 - Bootstrap agiliza o desenvolvimento de GUI para linguagens Web
 - Doctrine agiliza o uso de Bancos de Dados para PHP

E QUAL LINGUAGEM IREMOS UTILIZAR?

- O foco da linguagem precisa ser o aprendizado
 - O compilador/interpretador precisa retornar mensagens de erro coerentes
- Simples de utilizar e que não exija muito conhecimento para executar
 - Um clique no objeto ou um comando simples no terminal
- Quanto mais conteúdo online, melhor
 - Vocês aprendem mais pesquisando por conta própria do que comigo
- Não permitir liberdade demais para os programadores (no caso, os alunos)
 - Vocês tem que aprender a fazer o certo primeiro, depois se especializa em gambiarra

E QUAL LINGUAGEM IREMOS UTILIZAR?

- O foco da linguagem precisa ser o aprendizado
 - O compilador/interpretador precisa retornar mensagens de erro coerentes
- Simples de utilizar e que não exija muito conhecimento para executar
 - Um clique no objeto ou um comando simples no terminal
- Quanto mais conteúdo online, melhor
 - Vocês aprendem mais pesquisando por conta própria do que comigo
- Não permitir liberdade demais para os programadores (no caso, os alunos)
 - Vocês tem que aprender a fazer o certo primeiro, depois se especializa em gambiarra



PORQUE C/C++?

- Fortemente tipada
 - Não vai permitir que vocês usem o dado da forma errada
- Compilador coerente
 - Retorna mensagens de erro que te ajudam a entender o problema
(na maioria das vezes *hehehe*)
- Comunidade gigantesca
 - Absolutamente toda dúvida que vocês tiverem tem resposta na internet
 - A linguagem existe a mais de 30 anos e é muito utilizada para ensinar novatos
 - Google e Stackoverflow, seus melhores amigos
- Fácil de ser executado, em qualquer sistema operacional
 - DevC++ no Windows
 - Nativo no Linux (dá pra escrever no bloco de notas e compilar na mão se quiser)
 - CodeBlocks para o MacOS
 - Até online, usando o `cpp.sh` ou OnlineGDB
 - E mais uma infinidade de IDE's: Eclipse, VisualStudio, Netbeans, Notepad++, etc.

PORQUE C/C++?

- A maioria das funções padrão da linguagem estão em bibliotecas nativas
 - iostream - Entrada e Saída padrão
 - cmath - Operações matemáticas como potência ou radiciação
 - fstream - Manipulação de arquivos
- A declaração de variáveis é permitida em qualquer local do código
- A comparação dos tipos de dados também é relevante
 - Inteiro 1 **não vai ser igual a** Real 1.0

**MÃOS!
A
OBRA!**

TRADUZINDO PORTUGOL PARA C/C++ - TIPOS DE DADOS

Visualg	C/C++
inteiro	int
real	float/double
caractere (única letra)	char
caractere (cadeia)	string
logico	bool

Tipo	Tamanho	Intervalo
Char	8 bits	-127 a 128
Unsigned char	8 bits	0 a 255
signed char	8 bits	-127 a 128
short int	16 bits	-32768 a 32767
unsigned shor int	16 bits	0 a 65535
signed short int	16 bits	-32768 a 32767
int	32 bits	-2.147.483.648 a 2.147.483.647
signed int	32 bits	-2.147.483.648 a 2.147.483.647
unsigned int	32 bits	0 a 4.294.967.295
long int	32 bits	-2.147.483.648 a 2.147.483.647
signed long int	32 bits	-2.147.483.648 a 2.147.483.647
unsigned long int	32 bits	0 a 4.294.967.295
float	32 bits	$3,4 \times 10^{-38}$ a $3,4 \times 10^{+38}$
Double	64 bits	$1,7 \times 10^{-308}$ a $1,7 \times 10^{+308}$
Long Double	80 bits	$3,4 \times 10^{4932}$ a $1,1 \times 10^{4932}$

TRADUZINDO PORTUGOL PARA C/C++ - ESTRUTURA DO CÓDIGO

Algoritmo "estrutura"

Var

// Seção de Declarações das variáveis

Início

// Seção de Comandos, procedimento, funções, operadores, etc...

Fimalgoritmo

TRADUZINDO PORTUGOL PARA C/C++ - ESTRUTURA DO CÓDIGO

Algoritmo "estrutura"

Var

// Seção de Declarações das variáveis

Início

// Seção de Comandos, procedimento, funções, operadores, etc...

Fimalgoritmo

```
// Inclusão de Bibliotecas
#include <iostream>

int main() {
    // Código fonte
}
```

TRADUZINDO PORTUGOL PARA C/C++ - ATRIBUIÇÃO

```
Algoritmo "estrutura"  
Var  
// Seção de Declarações das variáveis  
a, b, c : inteiro  
|  
Inicio  
  
a := 5  
b := 10  
  
c := a * b  
  
Fimalgoritmo
```

TRADUZINDO PORTUGOL PARA C/C++ - ATRIBUIÇÃO

Algoritmo "estrutura"

Var

// Seção de Declarações das variáveis

a, b, c : inteiro

|

Inicio

a := 5

b := 10

c := a * b

Fimalgoritmo

// Inclusão de Bibliotecas

`#include <iostream>`

```
int main() {  
    int a, b, c;
```

```
    a = 5;
```

```
    b = 10;
```

```
    c = a * b;
```

```
}
```

TRADUZINDO PORTUGOL PARA C/C++ - OPERADORES BÁSICOS

Operador	Visualg	C/C++
Matemática Básica	+, -, *, /	+, -, *, /
Relacionais	>, <, =, <>	>, <, ==, !=
Lógicos	E, OU, NAO	&&, , !

TRADUZINDO PORTUGOL PARA C/C++ - DECISÕES

```
se condicao entao
    // Faca algo se for verdade
senao
    // Faca algo se for falso
fimse
```

```
escolha valor
    caso 0 faca
        // caso valor = 0
    caso 1 faca
        // caso valor = 1
    outrocaso faca
        // qualquer outro caso
fimescolha
```

TRADUZINDO PORTUGOL PARA C/C++ - DECISÕES

```
se condicao entao
    // Faca algo se for verdade
senao
    // Faca algo se for falso
fimse
```

```
escolha valor
    caso 0 faca
        // caso valor = 0
    caso 1 faca
        // caso valor = 1
    outrocaso faca
        // qualquer outro caso
fimescolha
```

```
if (condicao) {
    // Faça algo se for verdade
} else {
    // Faça algo se for falso
}
```

```
switch(valor) {
    case 0:
        // se valor = 0
        break;
    case 1:
        // se valor = 1
        break;
    default:
        // qualquer outro caso
        break;
}
```

TRADUZINDO PORTUGOL PARA C/C++ - REPETIÇÕES

```
repita
    // qualquer coisa
ate condicao

enquanto condicao
    // qualquer outra coisa
fimenquanto

para i de 1 ate 10 faca
    // repetindo o bloco
fimpara
```

TRADUZINDO PORTUGOL PARA C/C++ - REPETIÇÕES

```
repita
    // qualquer coisa
ate condicao

enquanto condicao
    // qualquer outra coisa
fimenquanto

para i de 1 ate 10 faça
    // repetindo o bloco
fimpara
```

```
do {
    // qualquer coisa
} while(condicao);

while(condicao) {
    // qualquer outra coisa
}

for(int i = 1; i <= 10; i++) {
    // repetindo o bloco
}
```

TRADUZINDO PORTUGOL PARA C/C++ - VETORES E MATRIZES

TRADUZINDO PORTUGOL PARA C/C++ - OLÁ MUNDO!

Algoritmo "olamundo"

Var

texto : caractere

Inicio

texto := "Olá Mundo"

escreval(**texto**)

Fimalgoritmo

TRADUZINDO PORTUGOL PARA C/C++ - OLÁ MUNDO!

Algoritmo "olamundo"

Var

texto : caractere

Inicio

texto := "Olá Mundo"

escreval(**texto**)

Fimalgoritmo

```
// Inclusão de Bibliotecas  
#include <iostream>  
#include <string>  
  
int main() {  
    std::string texto = "Ola Mundo!";  
  
    std::cout << texto << std::endl;  
}
```

TRADUZINDO PORTUGOL PARA C/C++ - OLÁ FULANO!

Algoritmo "olafulano"

Var

texto : caractere

Inicio

escreval("Insira o nome")

leia(**texto**)

escreval("Ola, ", **texto**)

Fimalgoritmo

TRADUZINDO PORTUGOL PARA C/C++ - OLÁ FULANO!

Algoritmo "olafulano"

Var

texto : caractere

Inicio

escreval("Insira o nome")

leia(**texto**)

escreval("Ola, ", **texto**)

Fimalgoritmo

```
// Inclusão de Bibliotecas
#include <iostream>
#include <string>

int main() {
    std::string texto;

    std::cout << "Informe o nome:" << std::endl;
    std::getline(std::cin, texto);

    std::cout << "Ola, " << texto << std::endl;
}
```

TRADUZINDO PORTUGOL PARA C/C++ - QUATRO OPERAÇÕES

Algoritmo "olafulano"

Var

a, b : inteiro

Inicio

escreval("Insira o valor de A")

leia(**a**)

escreval("Insira o valor de B")

leia(**b**)

escreval("Soma", **a + b**)

escreval("Subtracao", **a - b**)

escreval("Multiplicacao", **a * b**)

escreval("Divisao", **a / b**)

Fimalgoritmo

TRADUZINDO PORTUGOL PARA C/C++ - QUATRO OPERAÇÕES

Algoritmo "olafulano"

Var

a, b : inteiro

Inicio

escreval("Insira o valor de A")

leia(**a**)

escreval("Insira o valor de B")

leia(**b**)

escreval("Soma", **a + b**)

escreval("Subtracao", **a - b**)

escreval("Multiplicacao", **a * b**)

escreval("Divisao", **a / b**)

Fimalgoritmo

```
// Inclusão de Bibliotecas
#include <iostream>
#include <string>

int main() {
    int a, b;

    std::cout << "Informe o valor de A:" << std::endl;
    std::cin >> a;

    std::cout << "Informe o valor de b:" << std::endl;
    std::cin >> b;

    std::cout << "Soma " << a + b << std::endl;
    std::cout << "Subtracao " << a - b << std::endl;
    std::cout << "Multiplicacao " << a * b << std::endl;
    std::cout << "Divisao " << a / b << std::endl;
}
```

TRADUZINDO PORTUGOL PARA C/C++ - PAR OU ÍMPAR

Algoritmo "olafulano"

Var

a : inteiro

Inicio

escreval("Insira o valor de A")

leia(**a**)

se **a** % 2 = 0 entao

 escreval(**a**, " eh par")

senao

 escreval(**a**, " eh impar")

fimse

Fimalgoritmo

TRADUZINDO PORTUGOL PARA C/C++ - PAR OU ÍMPAR

Algoritmo "olafulano"

Var

a : inteiro

Inicio

escreval("Insira o valor de A")

leia(**a**)

se **a** % 2 = 0 entao

 escreval(**a**, " eh par")

senao

 escreval(**a**, " eh impar")

fimse

Fimalgoritmo

```
// Inclusão de Bibliotecas
#include <iostream>

int main() {
    int a;

    std::cout << "Informe o valor de A:" << std::endl;
    std::cin >> a;

    if (a % 2 == 0) {
        std::cout << a << " eh par" << std::endl;
    } else {
        std::cout << a << " eh impar" << std::endl;
    }
}
```

E essa ruma de
comando novo aí, professor?
Como vou saber o que é cada
coisa nisso aí?



BIBLIOTECAS PADRÃO - IOSTREAM

- Biblioteca responsável pela entrada e saída padrão do sistema
- Similar ao `escreva` e `leia` do VisuALG
- Para escrever em tela, utilizamos o `cout`
- Para ler algo da entrada padrão, utilizamos o `cin`
 - Se for ler uma string, é melhor utilizar o comando `getline()`;

```
// Inclusão de Bibliotecas
#include <iostream>
#include <string>

int main() {
    std::string texto;
    int valor;

    std::cout << "Insira o valor" << std::endl;
    std::cin >> valor;

    std::cout << "Insira o nome" << std::endl;
    std::getline(std::cin, texto);

    std::cout << valor << " || " << texto << std::endl;

    return 0;
}
```

BIBLIOTECAS PADRÃO - CMATH

- Biblioteca responsável pela manipulação matemática além das operações básicas
 - Trigonometria: cos, sin, acos, asin, etc.
 - Exponenciais e Logaritmo
 - Potenciação e Radiciação
 - Arredondamento
 - Absolutos, máximos e mínimos
 - Entre outras coisas

```
// Inclusão de Bibliotecas
#include <iostream>
#include <cmath>

int main() {
    int a = 5, b = 10;

    int c = std::abs(a - b);
    int d = std::pow(a, b);
    int e = std::round(a / 2.0);
}
```

BIBLIOTECAS PADRÃO - STRING

- Biblioteca responsável pela manipulação de cadeias de caracteres
 - Tamanho da cadeia
 - Operações de busca na cadeia
 - Copiar apenas partes da cadeia
 - Entre outras coisas

```
// Inclusão de Bibliotecas
#include <iostream>
#include <string>

int main() {
    std::string texto = "Texto qualquer para aula";

    std::cout << texto.size() << std::endl;

    std::cout << texto.find("qualquer") << std::endl;

    std::string parte = texto.substr(texto.find("qualquer"));

    std::cout << parte << std::endl;

    return 0;
}
```

DETALHES DO C/C++ - NAMESPACES

- A princípio não parece muito útil, porém conforme o seu conhecimento vai crescendo, mais bibliotecas vão sendo necessárias;
- Algumas dessas bibliotecas tem reimplementações do mesmo método, ou até mesmo funções diferentes com o mesmo nome;
 - A biblioteca padrão do c++ tem a função *max*
 - A biblioteca Eigen (Álgebra) também tem uma função chamada *max*
- Para informar ao compilador de qual função você está falando, torna-se necessário utilizar o *namespace*
 - `std::max` para pegar a função *max* da biblioteca standard
 - `Eigen::max` para pegar a função *max* da biblioteca Eigen
- É possível suprimir o uso dessa declaração de namespace antes da função



HORA DE PRATICAR!!!

Refazendo a lista 1 do Classroom