

Conceitos básicos de C++ para maratona de programação

Victor Emanuel Almeida

UNIOESTE

24 de agosto de 2022



Conteúdo

- 1 Características C++
- 2 Ola Mundo
- 3 Sobrecarga de operador
- 4 Dicas c++ Maratona
- 5 Estruturas de dados STL
- 6 Exercício
- 7 Referências



Conceitos I

C++ is a superset of C

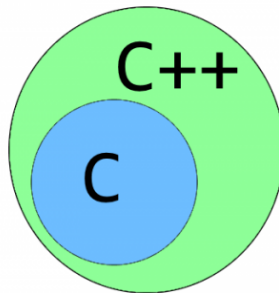


Figura 1: Diagrama de Venn C/C++



Conceitos II



C makes it easy to shoot yourself in the foot; C++ makes it harder, but when you do it blows your whole leg off.

— Bjarne Stroustrup —

AZ QUOTES



Conceitos III

- Linguagem compilada (g++);
- Fortemente tipada[1];
- Multiparadigma: Imperativa e orientada a objetos[1];
- Linguagem complexa com muitas instruções e palavras reservadas[1].



Vantagens do C++ sobre o C

- Orientação a objetos;
- Entrada e saída de dados;
- Sobrecarga de operadores;
- Referências;
- Alocação de memória e smart pointers;
- Bibliotecas padrão com algoritmos e estruturas de dados.



Exemplo de Ola Mundo Simples

```
1 #include <iostream>
2 using namespace std;
3 int main () {
4     cout << "Ola Mundo\n";
5     return 0;
6 }
```



Explicando o <<



Exemplo de Ola Mundo Maratona

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  int main () {
4      // desabilita a sincronizacao entre as
5      // streams do c com do c++
6      // retire se for usar printf e scanf
7      ios_base::sync_with_stdio(false);
8      // desabilita o flush automatico do buffer
9      // retire caso use um cout sem fim de linha
10     e logo em seguida cin
11     cin.tie(0);
12     cout << "Ola Mundo\n";
13     return 0;
14 }
```



Sequenciais I

Vetores: Tem uma “api” idêntica a vetores do C com acesso de um elemento específico, “vetor[i] = valor;”

- **array<T, size>**: Vetor do tipo T de tamanho size definido em tempo de compilação;
- **vector<T>**: Vetor do tipo T de tamanho variável;
- **deque<T>**: Fila duplamente encadeada do tipo T, funciona como o vector porém as ações de inserir e remover são um pouco mais eficientes;



Sequenciais II

Listas encadeadas: Não possui acesso a um elemento específico

- **forward_list**<T>: Lista encadeada do tipo T, so pode ser percorrida do começo para o fim;
- **list**<T>: Lista duplamente encadeada do tipo T, pode ser percorrida dos dois lados;



Interfaces FIFO e LIFO

Interfaces que por padrão utilizam estruturas como deque e list internamente:

- **stack**<T>: Pilha do tipo T, funciona como uma pilha de pratos, o último a entrar é o primeiro a sair (*LIFO, Last In First Out*);
- **queue**<T>: Fila do tipo T, funciona como uma fila de banco, o primeiro a entrar é o primeiro a sair (*FIFO, First In First Out*);



Associativas I

Implementam estruturas que permitem buscar um elemento de forma eficiente $O(\log(N))$ normalmente utilizando árvores como a rubro-negra ou tabelas hash.

- **set**<**T**>: Conjunto do tipo T, não permite elementos repetidos, é uma árvore rubro-negra;
- **map**<**K**, **V**>: Mapeia a chave do tipo K para um valor do tipo V, não permite chaves repetidas, é uma árvore rubro-negra;



Associativas II

Modificadores de estruturas associativas (multi e unordered):

- **multi**: Permite elementos repetidos, no caso da map chaves repetidas;
- **unordered**: Elementos não são alocados de maneira sequencial na estrutura;



Associativas III

- **set**: Árvore rubro-negra;
- **multiset**: Árvore rubro-negra;
- **unordered_set**: Tabela hash;
- **unordered_multiset**: Tabela hash;
- **map**: Árvore rubro-negra;
- **multimap**: Árvore rubro-negra;
- **unordered_map**: Tabela hash;
- **unordered_multimap**: Tabela hash;



Exemplo



Enunciado

You are given a list of n integers, and your task is to calculate the number of distinct values in the list.



Referências I

- 1 KHOURI, J. H. E. *Conceitos de Linguagens de Programação Evolução das principais linguagens de programação*. 2020. Acesso em: 21 de maio de 2022.

