

Vector

PROFESSOR: EDUARDO HABIB BECHELANE MAIA
HABIB@CEFETMG.BR

Introdução

- Estende a noção de vetor, tornando-a mais poderosa.
- Permite o armazenamento de uma sequência de objetos arbitrários
- Os elementos de um vector, assim como de um vetor, podem ser acessados especificando seu índice.
- **Principal vantagem:** Não é necessário saber o tamanho que você deseja que o vetor tenha ao declará-lo;
- Pode-se adicionar novos elementos ao final de um vetor usando a função `push_back`.
 - Permite que se insira novos elementos em qualquer posição do vetor, mas esta é uma operação muito ineficiente
 - Use Lista

Funções da classe Vector

- **size** : retorna o número de elementos no vetor
- **clear** : remove todos os elementos do vetor
- **empty** : retorna **true** se o vetor não tem elementos
- **push_back** : adiciona x ao final do vetor
- **pop_back** : Remove o objeto no final do vetor
- **back** : Retorna o objeto no final do vetor
- **front** : retorna o objeto na frente do vetor
- **erase**: Remove o elemento em uma posição do vetor
- **erase**: Remove um intervalo de elementos, incluindo o elemento first, mas não o last

Exemplos de Vector

```
#include <iostream>
#include <vector>
using namespace std;
int main(){
    vector<int> vetor;
    int i;
    // INSERINDO 5 VALORES DE 1 A 5 USANDO PUSH_BACK
    for(i = 0; i < 5; i++){
        vetor.push_back(i);
    }
    // PEGANDO O TAMANHO DO VETOR
    cout << "Tamanho do Vetor = " << vetor.size() << endl;
    // ACESSANDO OS 5 VALORES DO VETOR PASSANDO PELO
    INDEX
    for(i = 0; i < 5; i++){
        cout << "Valor do vetor [" << i << "] = " << vetor[i] << endl;
    }
    // OU USANDO O ITERATOR PARA ACESSAR OS VALORES.
    vector<int>::iterator v = vetor.begin();
    while( v != vetor.end()) {
        cout << "Valor do vetor = " << *v << endl;
        v++;
    }
    return 0;
}
```

Exemplo com o Erase

```
#include <iostream>
#include <vector>

int main ()
{
    std::vector<int> myvector;

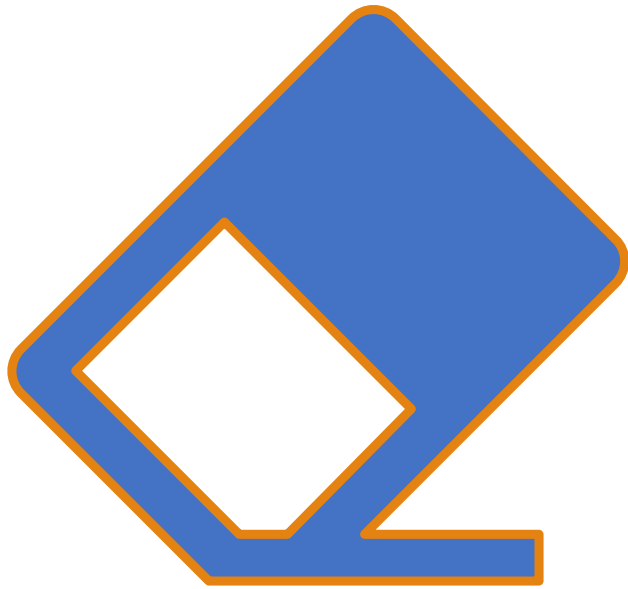
    // colocando valores no vetor
    for (int i=1; i<=10; i++)
        myvector.push_back(i);

    // Apagando o sexto elemento
    myvector.erase (myvector.begin()+5);

    // apagando os 3 primeiros elementos
    myvector.erase (myvector.begin(),myvector.begin()+3);

    std::cout << "Vetor:";
    for (unsigned i=0; i<myvector.size(); ++i)
        std::cout << ' ' << myvector[i];
    std::cout << '\n';

    return 0;
}
```



Mais exemples de vector

[HTTPS://WWW.CPLUSPLUS.COM/REFERENCE/VECTOR/VECTOR/](https://www.cplusplus.com/reference/vector/vector/)

Vector de classes

- A utilização seria similar a de um vector qualquer. Ex:
 - `vector<Pais> vetor;`

Pais.hpp

```
#ifndef __PAIS_HPP
#define __PAIS_HPP #include<vector>
#include <string>
using namespace std;
class Pais {
    private:
        string nome;
        float dimensao;
        string nomeCapital;
        vector <Pais> listaVizinhos;
    public:
        static int cont;
        int getCont();
        string getNome();
        void setNome(string nome);
        float getDimensao();
        void setDimensao(float dimensao);
        string getNomeCapital();
        void setNomeCapital(string nome);
        void setListaVizinhos(vector <Pais> listaPais);
        void verificarSePaisIgual(Pais x);
        void imprimirFronteiras();};#endif
```


Pais.cpp

```
#include <iostream>
#include <string>
#include <math.h>
#include <vector>
#include "Pais.hpp"
using namespace std;

int Pais::cont;

void Pais::setListaVizinhos(vector <Pais>
lp){
    listaVizinhos=lp;
}

int Pais::getCont(){
    return cont;
}

string Pais::getNome(){
    return nome;
}

void Pais::setNome(string nome){
    this->nome=nome;
}

float Pais::getDimensao()
{
    return this->dimensao;
}

void Pais::setDimensao(float dimensao)
{
    this->dimensao=dimensao;
}

string Pais::getNomeCapital(){
    return nomeCapital;
}

void Pais::setNomeCapital(string
nomeCapital){
    this->nomeCapital=nomeCapital;
}

void Pais::verificarSePaisIgual(Pais x){
    if ((x.getNome()== nome) &&
(x.getNomeCapital() == nomeCapital))
    {
        cout << "Países Iguais";
    } else cout << "São diferentes";
}

void Pais::imprimirFronteiras(){
    cout << "\nVizinhos: \n";
    for (int i=0;i<listaVizinhos.size();i++){
        cout <<
        listaVizinhos[i].getNome()<<endl;
    }
}
```

Main.cpp

```
#include <iostream>
#include <vector>
#include "Pais.hpp"
using namespace std;
int main(){
    int i;
    //Vetor de pais
    vector <Pais> vet;
    vector <Pais> vetVizinhos;
    Pais p, vizinho;
    string nome, capital;
    float dimensao;
    int numVizinhos;
    cout<< "Digite o nome do pais: ";
    getline(cin,nome);
    cout << "\nDigite o Dimensao: \n";
    cin >> dimensao;
    cout << "\nDigite a capital:";
    cin.ignore();
    getline(cin, capital);
    p.cont=0;
    p.setNome(nome);
    p.setDimensao(dimensao);
    p.setNomeCapital(capital);
    cout << "Digite numero de vizinhos: ";
    cin >> numVizinhos;
    i=0;
    cin.ignore();
    while (i<numVizinhos)
    {
        cout<< "Digite o nome do pais: \n";
        getline(cin,nome);
        cout << "\nDigite o Dimensao: \n";
        cin >> dimensao;
        cout << "\nDigite a capital:";
        cin.ignore();
        getline(cin, capital);
        vizinho.setNome(nome);
        vizinho.setDimensao(dimensao);
        vizinho.setNomeCapital(capital);
        vetVizinhos.push_back(vizinho);
        i++;
    }
    p.setListaVizinhos(vetVizinhos);
    vet.push_back(p);
    for (int i=0;i<vet.size();i++)
    {
        cout << "Nome: " << vet[i].getNome()
        << endl;
        vet[i].imprimirFronteiras();
    }
    return 0;
}
```