

Pilha

- Ultimo que entra é o primeiro que sai (LIFO)

```
In [1]: #include <iostream>
using namespace std;

#define MAX 5
```

Interface Classe

```
In [2]: class Pilha {
private:
    int contador;
    int capacidade;
    int conteudo[MAX];
public:
    Pilha();
    bool vazia();
    bool cheia();
    bool empilhar(int valor);
    int desempilhar();
    int retornaTopo();
};
```

Construtor:

```
In [3]: Pilha::Pilha() {
    contador = 0;
    capacidade = MAX;
}
```

```
In [4]: Pilha minhaPilha;
```

Verifica se a pilha está vazia

```
In [5]: bool Pilha::vazia() {
    if (contador == 0) {
        return true;
    } else {
        return false;
    }
}
```

```
In [6]: if (minhaPilha.vazia())
    cout << "Esta vazia!\n";
else
    cout << "Não esta vazia!\n";
```

Esta vazia!

Verifica se a pilha está cheia

```
In [7]: bool Pilha::cheia() {  
        if (contador == capacidade) {  
            return true;  
        } else {  
            return false;  
        }  
    }
```

```
In [8]: if (minhaPilha.cheia())  
        cout << "Esta cheia!\n";  
else  
        cout << "Não esta cheia!\n";
```

Não esta cheia!

Empilhar elemento

```
In [10]: bool Pilha::empilhar(int valor) {  
        if (cheia())  
            return false;  
        conteudo[contador] = valor;  
        contador++;  
        return true;  
    }
```

```
In [11]: if (minhaPilha.empilhar(4))  
        cout << "Empilhado\n";  
else  
        cout << "Não empilhou\n";
```

Empilhado

```
In [12]: if (minhaPilha.empilhar(3))  
        cout << "Empilhado\n";  
else  
        cout << "Não empilhou\n";
```

Empilhado

```
In [13]: if (minhaPilha.empilhar(2))  
        cout << "Empilhado\n";  
else  
        cout << "Não empilhou\n";
```

Empilhado

```
In [14]: if (minhaPilha.empilhar(1))  
        cout << "Empilhado\n";  
else  
        cout << "Não empilhou\n";
```

Empilhado

```
In [15]: if (minhaPilha.empilhar(0))  
        cout << "Empilhado\n";
```

```
else
    cout << "Não empilhou\n";
```

Empilhado

```
In [16]: if (minhaPilha.empilhar(-1))
          cout << "Empilhado\n";
          else
            cout << "Não empilhou\n";
```

Não empilhou

```
In [17]: if (minhaPilha.cheia())
          cout << "Esta cheia!\n";
          else
            cout << "Não esta cheia!\n";
```

Esta cheia!

Desempilhar

```
In [18]: int Pilha::desempilhar() {
          if (vazia())
              return false;
          contador--;
          return conteudo[contador];
          }
```

```
In [19]: cout << "Desempilhado: " << minhaPilha.desempilhar() << "\n";
```

Desempilhado: 0

```
In [20]: cout << "Desempilhado: " << minhaPilha.desempilhar() << "\n";
```

Desempilhado: 1

```
In [21]: cout << "Desempilhado: " << minhaPilha.desempilhar() << "\n";
```

Desempilhado: 2

```
In [22]: cout << "Desempilhado: " << minhaPilha.desempilhar() << "\n";
```

Desempilhado: 3

```
In [23]: cout << "Desempilhado: " << minhaPilha.desempilhar() << "\n";
```

Desempilhado: 4

```
In [24]: if (minhaPilha.vazia())
          cout << "Esta vazia!\n";
          else
            cout << "Não esta vazia!\n";
```

Esta vazia!

```
In [25]: cout << "Desempilhado: " << minhaPilha.desempilhar() << "\n";
```

Desempilhado: 0

Esse 0 indica que não foi possível desempilhar ou desempilhou o valor 0?

In [26]:

```
cout << "Desempilhado: " << minhaPilha.desempilhar() << "\n";
```

Desempilhado: 0

Correção:

Interface Classe

In [54]:

```
class Pilha {
private:
    int contador;
    int capacidade;
    int conteudo[MAX];
public:
    Pilha();
    bool vazia();
    bool cheia();
    bool empilhar(int valor);
    bool desempilhar(int &valor);
    bool retornaTopo(int &valor);
};
```

Construtor:

In [55]:

```
Pilha::Pilha() {
    contador = 0;
    capacidade = MAX;
}
```

In [56]:

```
Pilha minhaPilha;
```

Verifica se a pilha está vazia

In [57]:

```
bool Pilha::vazia() {
    if (contador == 0) {
        return true;
    } else {
        return false;
    }
}
```

In [58]:

```
if (minhaPilha.vazia())
    cout << "Esta vazia!\n";
else
    cout << "Não esta vazia!\n";
```

Esta vazia!

Verifica se a pilha está cheia

```
In [59]: bool Pilha::cheia() {  
        if (contador == capacidade) {  
            return true;  
        } else {  
            return false;  
        }  
    }
```

```
In [60]: if (minhaPilha.cheia())  
        cout << "Esta cheia!\n";  
        else  
            cout << "Não esta cheia!\n";
```

Não esta cheia!

Empilhar elemento

```
In [61]: bool Pilha::empilhar(int valor) {  
        if (cheia())  
            return false;  
        conteudo[contador] = valor;  
        contador++;  
        return true;  
    }
```

```
In [62]: if (minhaPilha.empilhar(4))  
        cout << "Empilhado\n";  
        else  
            cout << "Não empilhou\n";
```

Empilhado

```
In [63]: if (minhaPilha.empilhar(3))  
        cout << "Empilhado\n";  
        else  
            cout << "Não empilhou\n";
```

Empilhado

```
In [64]: if (minhaPilha.empilhar(2))  
        cout << "Empilhado\n";  
        else  
            cout << "Não empilhou\n";
```

Empilhado

```
In [65]: if (minhaPilha.empilhar(1))  
        cout << "Empilhado\n";  
        else  
            cout << "Não empilhou\n";
```

Empilhado

```
In [66]: if (minhaPilha.empilhar(0))  
        cout << "Empilhado\n";
```

```
else
    cout << "Não empilhou\n";
```

Empilhado

```
In [67]: if (minhaPilha.empilhar(-1))
          cout << "Empilhado\n";
          else
          cout << "Não empilhou\n";
```

Não empilhou

```
In [68]: if (minhaPilha.cheia())
          cout << "Esta cheia!\n";
          else
          cout << "Não esta cheia!\n";
```

Esta cheia!

Desempilhar

```
In [69]: bool Pilha::desempilhar(int &valor) {
          if (vazia())
              return false;
          contador--;
          valor = conteudo[contador];
          return true;
          }
```

```
In [70]: int recebeValor;
```

```
In [71]: if (minhaPilha.desempilhar(recebeValor))
          cout << "Desempilhou: " << recebeValor;
          else
          cout << "Não desempilhou!";
```

Desempilhou: 0

```
In [72]: if (minhaPilha.desempilhar(recebeValor))
          cout << "Desempilhou: " << recebeValor;
          else
          cout << "Não desempilhou!";
```

Desempilhou: 1

```
In [73]: if (minhaPilha.desempilhar(recebeValor))
          cout << "Desempilhou: " << recebeValor;
          else
          cout << "Não desempilhou!";
```

Desempilhou: 2

```
In [74]: if (minhaPilha.desempilhar(recebeValor))
          cout << "Desempilhou: " << recebeValor;
          else
          cout << "Não desempilhou!";
```

Desempilhou: 3

```
In [75]: if (minhaPilha.desempilhar(recebeValor))
        cout << "Desempilhou: " << recebeValor;
        else
        cout << "Não desempilhou!";
```

Desempilhou: 4

```
In [76]: if (minhaPilha.desempilhar(recebeValor))
        cout << "Desempilhou: " << recebeValor;
        else
        cout << "Não desempilhou!";
```

Não desempilhou!

```
In [77]: if (minhaPilha.vazia())
        cout << "Vazia";
```

Vazia

```
In [102... for (int i = 10 ; i < 20 ; i++) {
        if (minhaPilha.empilhar(i))
            cout << "Empilhou: " << i << endl;
        else
            cout << "Não empilhou!" << endl;
    }
```

Empilhou: 10
Empilhou: 11
Empilhou: 12
Empilhou: 13
Empilhou: 14
Não empilhou!
Não empilhou!
Não empilhou!
Não empilhou!
Não empilhou!

Retorna Topo

```
In [79]: bool Pilha::retornaTopo(int &valor) {
        if (vazia())
            return false;
        valor = conteudo[contador-1];
        return true;
    }
```

```
In [103... if (minhaPilha.cheia())
        cout << "Cheia";
```

Cheia

```
In [104... if (minhaPilha.retornaTopo(recebeValor))
        cout << recebeValor;
        else
        cout << "Vazia!";
```

```
In [105... if (minhaPilha.retornaTopo(recebeValor))
               cout << recebeValor;
            else
               cout << "Vazia!";
```

14

```
In [106... minhaPilha.desempilhar(recebeValor);
            if (minhaPilha.retornaTopo(recebeValor))
               cout << recebeValor;
            else
               cout << "Vazia!";
```

13

```
In [107... if (minhaPilha.retornaTopo(recebeValor))
               cout << recebeValor;
            else
               cout << "Vazia!";
```

13

```
In [108... minhaPilha.desempilhar(recebeValor);
            if (minhaPilha.retornaTopo(recebeValor))
               cout << recebeValor;
            else
               cout << "Vazia!";
```

12

```
In [109... if (minhaPilha.retornaTopo(recebeValor))
               cout << recebeValor;
            else
               cout << "Vazia!";
```

12

```
In [110... minhaPilha.desempilhar(recebeValor);
            if (minhaPilha.retornaTopo(recebeValor))
               cout << recebeValor;
            else
               cout << "Vazia!";
```

11

```
In [111... minhaPilha.desempilhar(recebeValor);
            if (minhaPilha.retornaTopo(recebeValor))
               cout << recebeValor;
            else
               cout << "Vazia!";
```

10

```
In [112... minhaPilha.desempilhar(recebeValor);
            if (minhaPilha.retornaTopo(recebeValor))
               cout << recebeValor;
            else
               cout << "Vazia!";
```

Vazia!

Exemplo Listar:

In []: