

01_filha_estatica

September 17, 2021

1 Filas

- Fila é uma lista na qual as inserções são realizadas em uma extremidade (fim da fila) e as remoções em outra (começo da fila)
- FIFO: First In, First Out ou primeiro que entra, primeiro que sai
- Veremos duas implementações estática e dinâmica

1.1 Implementação Fila Estática

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

#define MAX 5
```

1.1.1 Definição da Classe

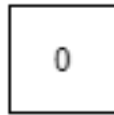
```
[2]: class Fila {
    private:
        int contador;
        int capacidade = MAX;
        int elementos[MAX];
    public:
        Fila();
        bool vazia();
        bool cheia();
        bool inserir(int x);
        bool remover(int &x);
};
```

1.1.2 Método Construtor

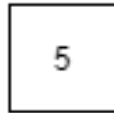
```
[3]: Fila::Fila() {
    contador = 0;
}
```

```
[4]: Fila minhaFila;
```

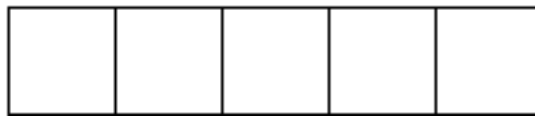
contador



capacidade



elementos



1.1.3 Verifica se fila está vazia

```
[5]: bool Fila::vazia() {  
    return contador == 0;  
}
```

```
[6]: if (minhaFila.vazia()) {  
    cout << "Está vazia!";  
} else {  
    cout << "Não está vazia!";  
}
```

Está vazia!

1.1.4 Verifica se fila está cheia

```
[7]: bool Fila::cheia() {  
    return contador == capacidade;  
}
```

```
[8]: if (!minhaFila.cheia()) {  
    cout << "Não está cheia!";  
}
```

Não está cheia!

```
[9]: if (minhaFila.cheia() == false) {  
    cout << "Não está cheia!";  
}
```

Não está cheia!

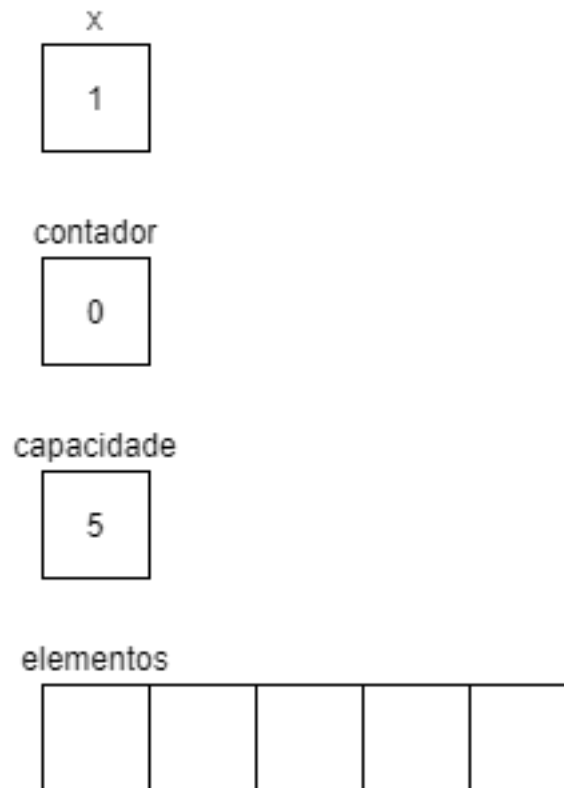
1.1.5 Inserção elemento na fila

```
[10]: bool Fila::inserir(int x) {  
    if (cheia()) {  
        return false;  
    }  
    elementos[contador] = x;  
    contador++;  
    return true;  
}
```

```
[11]: minhaFila.inserir(1)
```

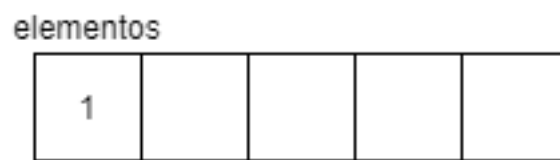
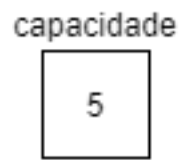
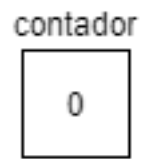
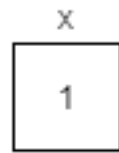
```
[11]: true
```

Chamando o método:



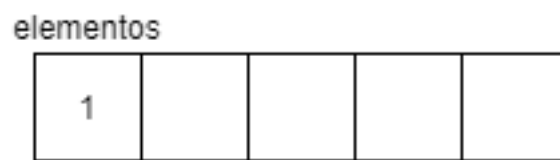
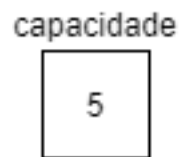
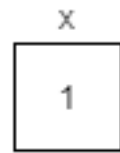
Armazenando valor no vetor de elementos:

```
elementos[contador] = x;
```



Incrementando contador de elementos:

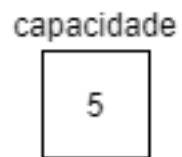
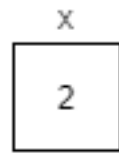
```
contador++;
```



```
[12]: minhaFila.inserir(2)
```

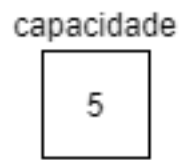
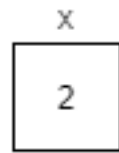
```
[12]: true
```

Chamando o método:



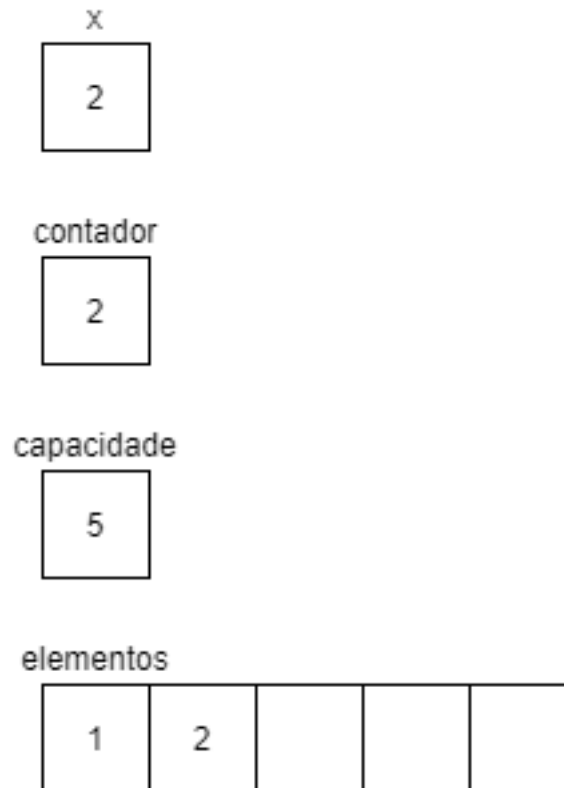
Armazenando valor no vetor de elementos:

```
elementos[contador] = x;
```



Incrementando contador de elementos:

`contador++;`



```
[13]: for (int i = 3; i < 10; i++)  
      if (minhaFila.inserir(i))  
          cout << i << " inserido com sucesso!\n";  
      else  
          cout << i << " não foi inserido!\n";
```

```
3 inserido com sucesso!  
4 inserido com sucesso!  
5 inserido com sucesso!  
6 não foi inserido!  
7 não foi inserido!  
8 não foi inserido!  
9 não foi inserido!
```

```
[14]: if (minhaFila.cheia())  
      cout << "Está cheia!";
```

Está cheia!

Situação final da fila após inserções:

contador

5

capacidade

5

elementos

1	2	3	4	5
---	---	---	---	---

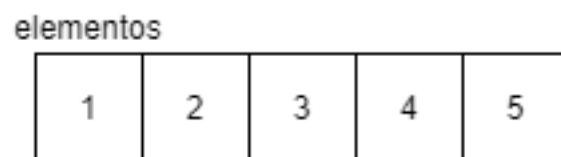
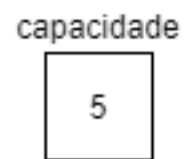
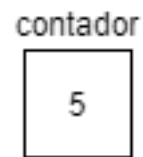
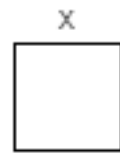
1.1.6 Remoção de elemento da fila

```
[15]: bool Fila::remover(int &x) {  
    if (vazia()) {  
        return false;  
    }  
    x = elementos[0];  
    for (int i = 1 ; i < contador ; i++) {  
        elementos[i-1] = elementos[i];  
    }  
    contador--;  
    return true;  
}
```

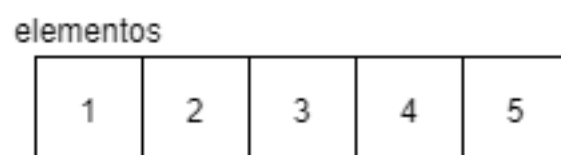
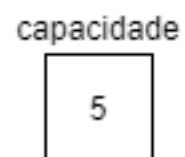
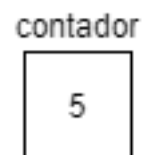
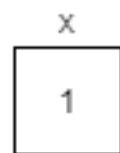
```
[16]: int y;  
if (minhaFila.remover(y))  
    cout << y << " removido com sucesso!\n";  
else  
    cout << "não houve remoção!\n";
```

1 removido com sucesso!

Chamando o método remover:

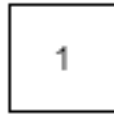


```
x = elementos[0];
```

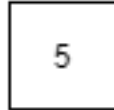


```
for (int i = 1; i < contador; i++)  
    elementos[i-1] = elementos[i];
```

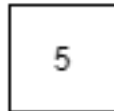
x



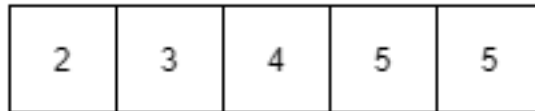
contador



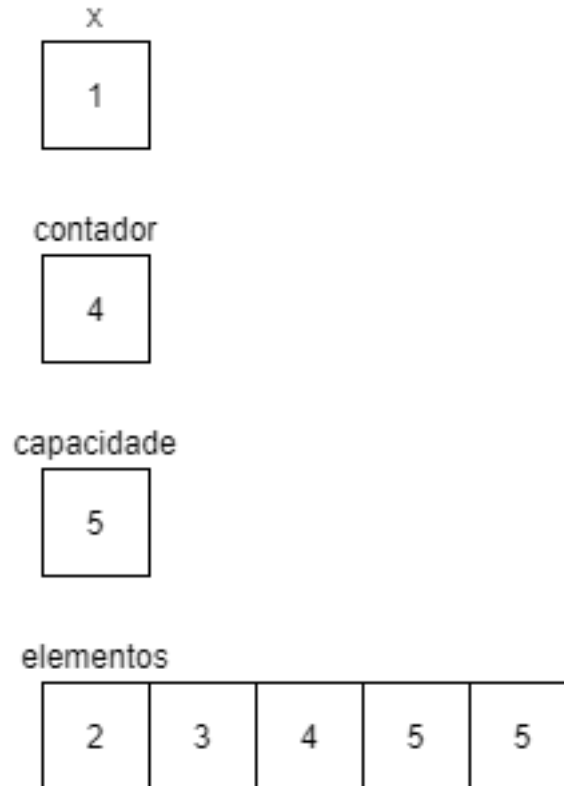
capacidade



elementos



```
contador--;
```



```
[17]: int y;  
for (int i = 1 ; i < 10 ; i++) {  
    if (minhaFila.remover(y))  
        cout << i << " - " << y << " removido com sucesso!\n";  
    else  
        cout << i << " - " << "não houve remoção!\n";  
}
```

```
1 - 2 removido com sucesso!  
2 - 3 removido com sucesso!  
3 - 4 removido com sucesso!  
4 - 5 removido com sucesso!  
5 - não houve remoção!  
6 - não houve remoção!  
7 - não houve remoção!  
8 - não houve remoção!  
9 - não houve remoção!
```

```
[18]: if (minhaFila.vazia())  
    cout << "Fila vazia!!!";
```

Fila vazia!!!

```
[19]: minhaFila.elementos[2] = 3;
```

```

input_line_30:2:12: error: 'elementos' is a private
member of '__cling_N53::Fila'
  minhaFila.elementos[2] = 3;
                  ^

input_line_9:5:13: note: declared private here
    int elementos[MAX];
        ^

```

Interpreter Error:

1.2 Arquivos:

```

[20]: %%file estatica/Fila.h
      #define MAX 5

      class Fila {
      private:
          int contador;
          int capacidade = MAX;
          int elementos[MAX];
      public:
          Fila();
          bool inserir(int x);
          bool remover(int &x);
          bool vazia();
          bool cheia();
      };

```

Overwriting estatica/Fila.h

```

[21]: %%file estatica/Fila.cpp
      #include "Fila.h"

      Fila::Fila() {
          contador = 0;
      }

      bool Fila::vazia() {
          return contador == 0;
      }

      bool Fila::cheia() {
          return contador == capacidade;
      }

```

```

bool Fila::inserir(int x) {
    if (cheia()) {
        return false;
    }
    elementos[contador] = x;
    contador++;
    return true;
}

bool Fila::remover(int &x) {
    if (vazia()) {
        return false;
    }
    x = elementos[0];
    for (int i = 1 ; i < contador ; i++) {
        elementos[i-1] = elementos[i];
    }
    contador--;
    return true;
}

```

Overwriting estatica/Fila.cpp

```

[22]: %%file estatica/main.cpp
#include <iostream>
#include <cstdlib>
#include <locale.h>

using namespace std;

#include "Fila.cpp"

int main() {
    setlocale(LC_ALL, "Portuguese");
    Fila minhaFila;
    Fila fila01, fila02;

    for (int i = 10 ; i < 20 ; i++) {
        if (minhaFila.inserir(i))
            cout << "i: " << i << " - inserido com sucesso!!!\n";
        else
            cout << "i: " << i << " - não inserido!!!\n";
    }
}

```

Overwriting estatica/main.cpp

[23]: !g++ estatica/main.cpp -o estatica/main

[24]: !./estatica/main

```
i: 10 - inserido com sucesso!!!  
i: 11 - inserido com sucesso!!!  
i: 12 - inserido com sucesso!!!  
i: 13 - inserido com sucesso!!!  
i: 14 - inserido com sucesso!!!  
i: 15 - não inserido!!!  
i: 16 - não inserido!!!  
i: 17 - não inserido!!!  
i: 18 - não inserido!!!  
i: 19 - não inserido!!!
```

[]: