



Trabalho Prático Individual n.º 1

Implementação de um ADT de fila em C++

Paulo Correia, nº2851



Índice / Índice de Figuras

Introdução.....	3
Ponto 1.....	4
Ponto 2.....	5
Ponto 3.....	6
Conclusão.....	7
Figura 1 - ponto 1.....	4
Figura 2 - Ponto 2.....	5
Figura 3 – Ponto 3.....	6

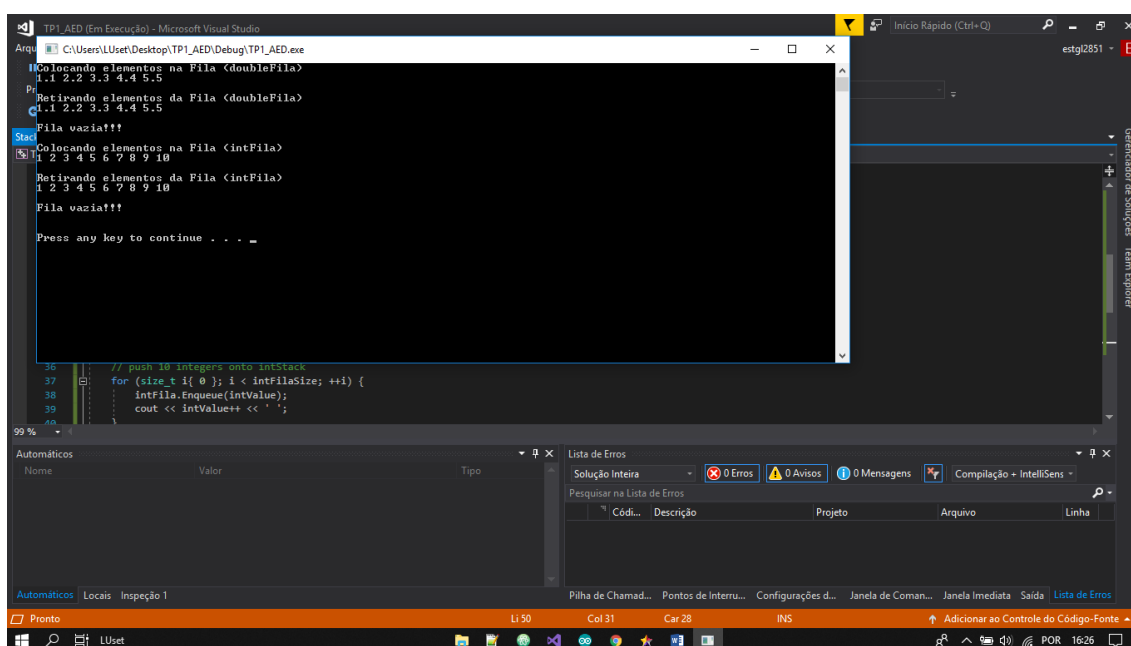
Introdução

Uma fila ou queue em inglês é uma estrutura de dados que usa o método FIFO (acrônimo para First In, First Out, que em português significa primeiro a entrar, primeiro a sair). A ideia fundamental da fila é que só podemos inserir um novo elemento no final da fila e só podemos retirar o elemento do início.

No Trabalho Prático foi pedido que se utiliza-se o exemplo de criação de um *template* da classe *Stack<T>*, constante da secção 18.2.1 *Creating Class Template Stack<T>* de (Deitel & Deitel, 2017), assim como, o exemplo do estudo e caso: *Array Class*, constante da secção 10.10.2 *Array Class Definition*, do mesmo livro, também experimentado nas aulas.

Ponto 1

O Ponto1 consiste em duas pilhas, uma que recebe elementos do tipo inteiro e outra do tipo *double*. Através dos comandos *push* e *pop*, os elementos eram colocados e retirados da pilha. Neste programa criaram-se duas pilhas com os mesmos elementos que eram adicionados ou retirados através dos comandos *Enqueue* e *Dequeue*.



```

TP1_AED (Em Execução) - Microsoft Visual Studio
Arq: C:\Users\LUset\Desktop\TP1_AED\Debug\TP1_AED.exe
Colocando elementos na Fila (doubleFila)
1 1 2 2 3 3 4 4 5 5
Retirando elementos da Fila (doubleFila)
1 1 2 2 3 3 4 4 5 5
Fila vazia!!!
Colocando elementos na Fila (intFila)
1 2 3 4 5 6 7 8 9 10
Retirando elementos da Fila (intFila)
1 2 3 4 5 6 7 8 9 10
Fila vazia!!!
Press any key to continue . . .

36 // push 10 integers onto intStack
37 for (size_t i{ 0 }; i < intFilaSize; ++i) {
38     intFila.Enqueue(intValue);
39     cout << intValue++ << ' ';
40 }

```

Figura 1 - ponto 1

Ponto 2

Neste ponto 2 foi feita a sobrecarga dos operadores de inserção (<<) e extração (>>) em *streams*, para obter e submeter, respetivamente, elementos de uma fila, no âmbito da implementação do ADT de fila.

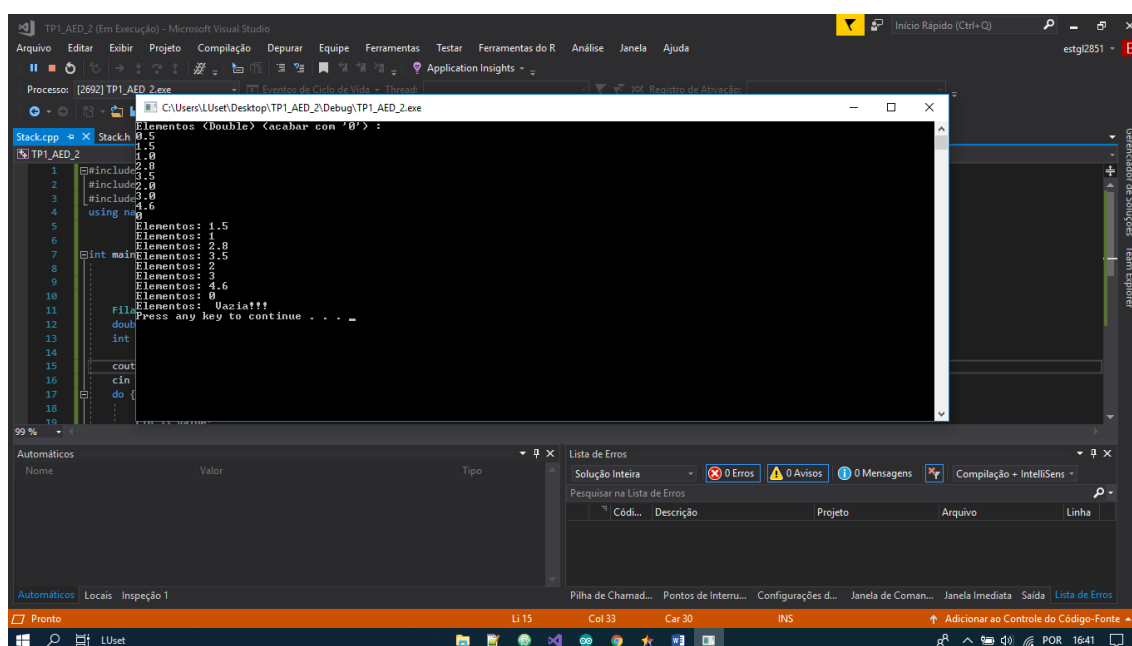


Figura 2 - Ponto 2

Ponto 3

Com a implementação da alínea 1., no caso de invocação de `dequeue()` ou `front()`, numa fila vazia, deverá ser provocada uma exceção. Esta poderá ser uma instância da exceção `std::logic_error (<stdexcept>)`, com a mensagem conveniente ("fila vazia"). Poderá ainda ser uma instância de uma classe especializada desta a ser desenvolvida (e.g., `EmptyQueueExpt`).

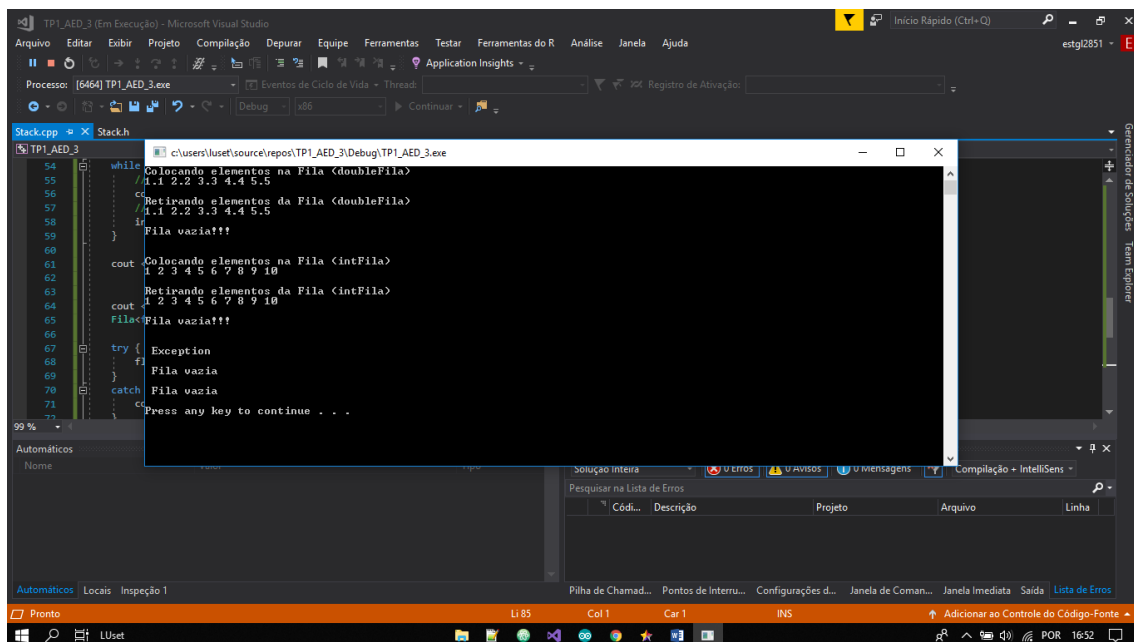


Figura 3 – Ponto 3

Este ponto do trabalho irá informar o utilizar que ocorreu um erro, quando este tenta tirar um elemento de uma fila que está vazia.

Conclusão

Com a realização desta atividade prática consegui entender melhor o funcionamento de uma fila no âmbito da programação em c++, no final da realização foi medida a complexidade das funções que é do tipo $O(1)$, isto é provado com o acesso direto aos dados o que faz com que a complexidade seja a mais simples. Descobrimos também que os elementos inseridos nas filas podem ser strings, inteiros, floats, doubles, structs, etc. Mas o essencial a ser retirado do trabalho prático é que os objetos que vão sendo inseridos, apenas o que foi colocado há mais tempo pode ser retirado.