

Nome: Pedro Loureiro Morone Branco Volpe
TIA: 42131936

– TAD Fila estática

1-

O TAD Fila estática é uma estrutura de dados capaz de armazenar e organizar outros tipos de dados.

A Fila é um vetor linear, a qual funciona com um sistema First in First out, isto é, o primeiro elemento adicionado na Fila, será o primeiro elemento desta, isto é, o Topo dela, e por consequência, o primeiro a ser removido; já o último elemento a ser inserido será o “fundo” da fila, e o último a ser removido. Assim percebe-se como a Fila pode se assemelhar a uma fila comum do dia-a-dia, como encontramos em padarias, e supermercados, onde os elementos que chegam primeiro são os primeiros a sair. A Fila tem diversas utilidades, uma delas é armazenar ações a serem realizadas em ordem de execução; armazenar informações com ordens específicas; ou até para ações práticas, como filas em hospitais que possuem duas filas, uma prioritária e outra comum.

2-

Funções de Fila criadas e implementadas:

(nota* - as funções foram feitas de maneira específica para o problema apresentado)

TAD Fila estática	
Enqueue('Fila', 'Coordenadas') Realiza a inserção de um elemento do tipo de dado da fila, no começo desta.	- Pré Condição: A fila não pode estar cheia (Contador = tamanho da Fila) - Pós condição: Nenhuma
Dequeue('Fila') Realiza a remoção do elemento presente no começo da fila.	-Pré Condição: A fila não pode estar vazia (contador = 0)

	-Pós condição: Retorna o elemento removido.
Front ('Fila') Retorna o elemento presente no primeiro elemento da fila, sem alterá-la.	-Pré Condição: A fila não pode estar vazia (contador = 0) -Pós condição: Retorna o primeiro elemento da fila.
Create() Cria uma Fila vazia.	-Pré Condição: Nenhuma -Pós condição: Retorna a Fila criada
IsFull(' Fila ') Verifica se a fila está cheia	-Pré Condição: Nenhuma -Pós condição: Retorna uma flag de verificação (true or false)
IsEmpty('Fila') Verifica se a fila está vazia	-Pré Condição: Nenhuma -Pós condição: Retorna uma flag de verificação (true or false)
PrintFila(' Fila ') Exibe a fila para o usuário	-Pré Condição: Nenhuma -Pós condição: Nenhuma

3-

Vide os anexos .cpp, .h e main.cpp

4-

Problema proposto:

Implementar 5 funções determinadas, com base em um código-base:

1 – Alterar uma coordenada

- 2 – Executar sequência de comandos
- 3 – Salvar comandos em um arquivo
- 4 – Carregar comandos de um arquivo
- 5 – Encerrar o programa

Solução:

Primeiramente, junto com a implementação da Fila, criei um struct de Coordenadas, o qual armazena as coordenadas dadas pelo usuário, e fiz a fila ser do mesmo tipo desse struct de Coordenadas, isto é, criei uma fila de Coordenadas.

1-

Para implementar o primeiro requisito (Alterar uma coordenada), criei uma função armazenaCoord(Coordenada &coordenadas_dadas), a qual recebe a entrada do usuário, verifica se está é válida (dentro do intervalo do canvas(matriz)), e armazena as coordenadas X e Y, no struct Coordenadas.

Em seguida, utilizo a função de fila Enqueue('Fila', 'Coordenadas'), para armazenar na fila, as coordenadas dadas.

E por fim, usei a função ToggleCanvas(canvas, coordenadas_dadas), para exibir o resultado no canvas(matriz).

2-

Para realizar a implementação do segundo item, primeiramente verifico se a fila está vazia, usando a função de fila IsEmpty('Fila'), se sim, retorno apenas um texto de fila vazia ao usuário.

Caso não, eu, primeiramente, limpo a tela e o canvas, à afim de simplificar a exibição, em seguida na função executaComandos(canvas, fila), exibo o canvas(matriz) e o conteúdo da fila a partir da função de fila PrintFila(' Fila ').

Assim usando um laço de repetição que ocorrerá enquanto houverem comandos a serem executados(isto é, enquanto a fila não estiver vazia), uso a função de fila Front('Fila'), para pegar o primeiro comando (Coordenada) a ser executado, e aplico-o no canvas usando a função ToggleCanvas(canvas, coordenadas_dadas), e por fim removo esse elemento da fila usando a função Dequeue('Fila'), e assim permitindo que o proximo comando seja acessível, no próximo loop.

3-

Para a terceira implementação, criei uma função salvaArquivo(fila), que chama a função de escrita de arquivo WriteStringToFile(const std::string& filename, Fila &fila), que foi completamente alterada, nela:

um dos parâmetros de entrada, anteriormente uma string a ser armazenada, foi alterado para a fila.

Dentro da função, realizei um loop que ocorre enquanto houver elementos dentro da fila, e nesse loop, usando a função Front, recebo o primeiro elemento da fila, converto-o em uma string e depois o armazeno no arquivo, usando as funções de arquivo.

4-

Nesse item, criei uma função chamada lerArquivo(Canvas &canvas, Fila &fila), para executar os comando desejados, nela chamo a função de leitura de arquivo ReadFileAsString(filename, isOK), que me retorna uma string com todas as coordenadas salva e uma flag de confirmação.

Essa flag de confirmação, verifica se o arquivo foi aberto com sucesso ou não, se não, exibe uma mensagem na tela.

Caso sim, o programa entra em um loop, para receber as coordenadas e separá-las nos blocos de X e Y, convertê-las em inteiros e assim re-armazená-las na fila.

5-

Exibe na tela a mensagem : Fim
e termina o programa

5 - AUTO AVALIAÇÃO

Item avaliado	Pontuação
Explicação sobre o que é e para que serve o TAD Fila estática (individual).	1.8 - Poderia melhorar e exemplificar melhor
Documento PDF descrevendo o TAD Fila estática que foi elaborado pela dupla.	0.25

TAD contendo operações básicas (fundamentais) da fila.	0.75 (contém todas as básicas)
TAD contendo pelo menos quatro operações adicionais da fila.	0.2(contém 4 funções extras)
Estrutura de dados da fila (struct).	0.5 [funciona :)]
Implementação das operações básicas da fila, seguindo o TAD elaborado pela dupla.	2.25
Implementação das operações adicionais da fila, seguindo o TAD elaborado pela dupla	0.8
Solução do problema, de acordo com o enunciado (uso de fila).	2.25
Explicação da solução do problema no documento PDF.	0.75 (acho q ta bem explicado :)
Autoavaliação no documento PDF (individual)	0.25

Nota: 9.8

Referências:

- Aula de fila dada pelo professor André Kishimoto
- https://www.w3schools.com/cpp/cpp_strings_length.asp (len da string)
- <https://cplusplus.com/reference/fstream/ofstream/>
- https://www.w3schools.com/cpp/cpp_strings_concat.asp
- https://cplusplus.com/reference/string/to_string/
- <https://learn.microsoft.com/pt-br/cpp/cpp/try-throw-and-catch-state-ments-cpp?view=msvc-170>