

Desenvolvimento de estrutura de dados de Pilha e Fila com Listas Ligadas

Autores

Antonio Pegorelli Neto – RA:119101-4

Introdução

A evolução dos métodos computacionais atuais exige estruturas de dados flexíveis que podem evitar vazamentos de memória e ser utilizadas de acordo com a necessidade do programa. As listas ligadas oferecem a possibilidade de expansão até o limite da memória disponível no dispositivo e liberam a memória caso parte ou o total de seus dados entrem em desuso. O objetivo deste trabalho é gerar uma classe de dados de lista ligada que permite essa dinâmica e criar classes herdeiras para a geração de pilhas e filas.

Teoria

De acordo com o livro Algoritmos¹, a lista ligada é uma estrutura de dados onde os objetos são arranjados linearmente, mas diferente de um vetor, onde o arranjo dos dados são definidos pelo índice do vetor, o arranjo da lista ligada é definido por um ponteiro em cada objeto. Também definidos no livro Algoritmos¹, em uma pilha o último elemento inserido é o primeiro a ser retirado (ou LIFO – Last-In, First-Out), na fila, no entanto, o primeiro elemento a ser retirado é o primeiro que foi colocado na estrutura (ou FIFO – First-In, First-out).

Proposta

A primeira classe a ser definida é a classe nó onde será introduzido o dado armazenado e os ponteiros para o próximo nó e para o nó anterior:

- pushData – Atualiza o dado no nó;
- pushNext – Atualiza o ponteiro para o próximo nó;
- pushLast – Atualiza o ponteiro para o nó anterior;
- getData – Retorna o dado do nó
- *getNext – Retorna o ponteiro para o próximo nó
- *getLast – Retorna o ponteiro para o nó anterior

A classe lista ligada define todos os métodos para criação, deleção e verificação de nós na lista, que pode ser usada tanto para a pilha quanto para a fila:

- isEmpty – Verifica se a lista está vazia
- *getStart – Retorna o ponteiro para o nó inicial;
- *getEnd – Retorna o ponteiro para o nó final;
- addStart – Adiciona um nó no início da lista;
- addEnd – Adiciona um nó no final da lista;
- delStart – Remove o nó no início da lista;
- delEnd – Remove o nó no final da lista.

A implementação da classe pilha envolve a criação do próprio vetor com tamanho 500 onde serão colocados os dados, uma variável com o tamanho atual da pilha e suas funções de interface:

- push – Para inserir um dado novo na pilha;
- pop – Para remover um dado da pilha;
- top – Para verificar o dado do topo da pilha.

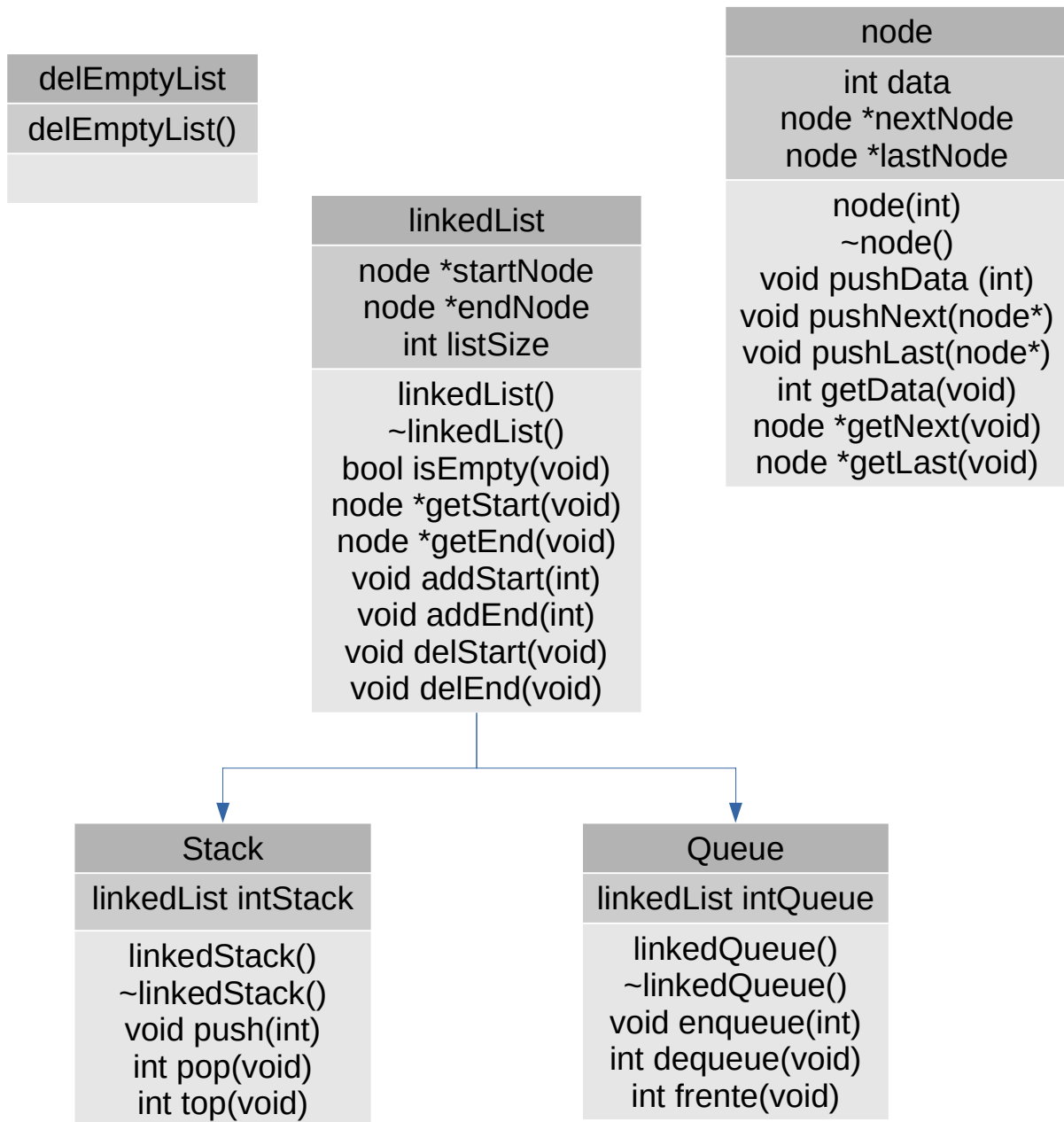
Para a fila, também é criado um vetor Para a fila é necessário as funções:

- enqueue – Para inserir um dado novo na fila;
- dequeue – Para retirar um dado da fila;
- frente – Para verificar o primeiro dado da fila.

Para finalizar, a classe delEmptyList define a exceção quando se tenta deletar um nó de uma lista vazia:

- delEmptyList – Handler da exceção.

UMLs e suas heranças:



Experimentos e Resultados

Para realizar o teste das estruturas de dados, fizemos um breve código que cria uma estrutura de teste de cada tipo com valores inteiros aleatórios e chama algumas vezes cada método das duas classes, indicando na tela o que foi feito. A resposta dos métodos foi satisfatória, sem erros na conexão dos nós da lista ligada e as excessões funcionando corretamente.

Trabalhos Correlatos

Apesar da sua simples implementação, há uma variedade no método de criação para as estruturas de listas ligadas, como se observa no Wikipedia² sobre listas ligadas, onde cita o uso de nós sentinelas, que simplificam as operações na lista.

Conclusão

As estruturas de pilha e fila são simples e eficientes soluções para armazenamento de dados, que podem ser implementadas com baixo poder computacional e dificilmente apresentam problemas. O uso das listas ligadas gerou um valor extra devido a flexibilidade de tamanho e alocação de memória. O próximo passo poderia ser a alocação de nós sentinela que simplifica as operações e proporciona melhor confiabilidade na alocação de dados, principalmente quando a lista é criada.

Bibliografia

¹Algoritmos (3ª edição - Cormen, T; Leiserson, C; Rivest R e Stein, C)

²Wikipedia (Linked List): https://en.wikipedia.org/wiki/Linked_list