1. Desenvolva um projeto no GitHub que Implemente os testes o TAD Lista Arranjo.

------OK-------

1. Desenvolva um projeto no GitHub que estenda a classe de testes do TAD Pilha implementado usando array que teste uma pilha de String.

-------OK-------

1. Desenvolva um projeto no GitHub que implemente os testes do TAD Pilha usando LSE.

------OK-------

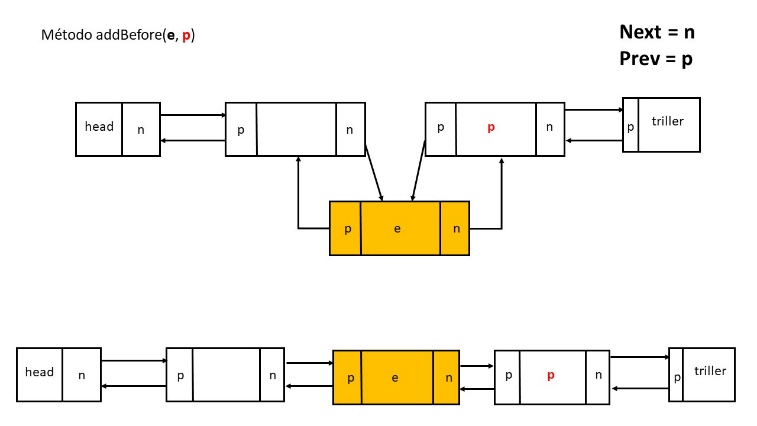
1. Exercícios: ------OK----------
   1. Crie testes e programas Java que:
   2. Inverta os dados de um arranjo usando o TAD Pilha usando LSE.
   3. Verifique se parênteses, colchetes e chaves estão corretos numa expressão matemática, por exemplo: [(5 + x)/4 – 2\*(y + z)]
      1. Correto: ()(( )){([( )])}
      2. Correto: ((( )(( )){([( )])}))
      3. Incorreto: )(( )){([( )])}
      4. Incorreto: ({[])}
      5. Incorreto: (
2. Suponha que uma lista inicialmente vazia S tenha executado um total de 25 operações push, 12 operações top e 10 operações pop, 3 das quais geraram StackEmptyExceptions, que foram capturadas e ignoradas. Qual é o tamanho corrente de S? O tamanho é de 15.
3. Se implementarmos a pilha S do problema anterior usando um arranjo, então qual será o valor corrente da variável de instância top? O tamanho é de 15.
4. Descreva a saída resultante da seguinte série de operações de pilha: push(5), push(3), pop( ),push(2), push(8), pop( ), pop( ), push(9), push(1), pop( ), push(7), push(6), pop(), pop(), push(4), pop(), pop( ).

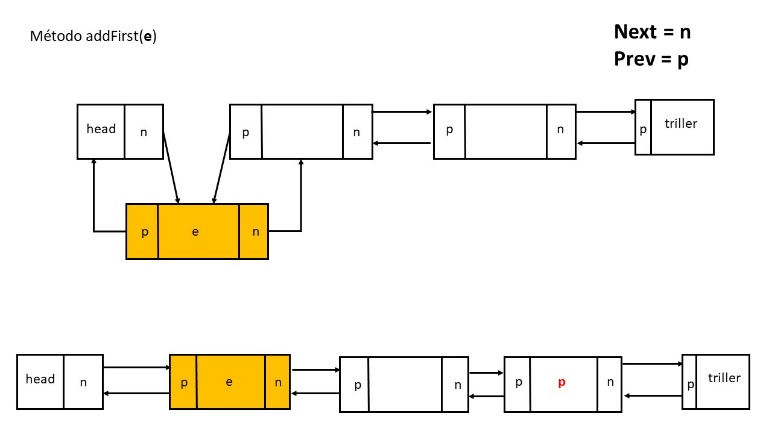
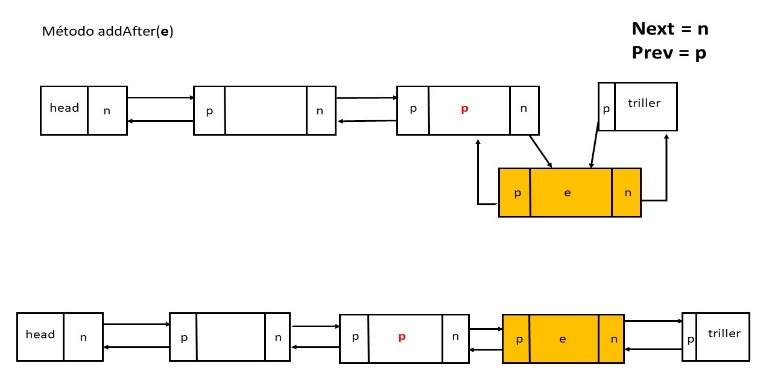
|  |  |  |
| --- | --- | --- |
| **Operação** | **Saída** | **Conteúdo da pilha** |
| push(5) | - | "(5)" |
| push(3) | - | "(5,3)" |
| pop() | 3 | "(5)" |
| push(2) | - | "(5,2)" |
| push(8) | - | "(5,2,8)" |
| pop() | 8 | "(5,2)" |
| push(9) | - | "(5,2,9)" |
| push(1) | - | "(5,2,9,1)" |
| pop() | 1 | "(5,2,9)" |
| push(7) | - | "(5,2,9,7)" |
| push(6) | - | "(5,2,9,7,6)" |
| pop() | 6 | "(5,2,9,7)" |
| pop() | 7 | "(5,2,9)" |
| push(4) | - | "(5,2,9,4)" |
| pop() | 4 | "(5,2,9)" |
| pop() | 9 | "(5,2)" |

1. Crie os testes e implemente o TAD Fila. Use implementação do TAD Pilha como exemplo.--------OK----------
2. Implemente o TAD Fila com base nos testes e no fragmento de implementação de duas operações apresentados a seguir (Tarefa 13 - TAD-Fila.pptx, slidese 19, 20 e 21).

--------OK-----------

1. Desenhe figuras demonstrando cada um dos passos principais dos métodos addBefore(p, e), addFirst(e) e addLast(e) do TAD lista de nodos.





1. Implemente um método não recursivo para inverter uma lista de nodos.
2. Implemente um novo método, makeFirst(p), que move o elemento na posição p para a primeira posição, mantendo a ordem relativa dos demais elementos inalterada.

--------OK-------

1. A implementação de NodePositionList não faz verificações de erro para testar se uma dada posição p é realmente membro dessa lista em particular.
   1. Por exemplo, se p é uma posição da lista S, a execução T.addAfter(p,e) deveria lançar a exceção InvalidPositionException pois p não é uma posição de T.
   2. Descreva como alterar a implementação de NodePositionList de uma forma eficiente que impeça esses maus usos: Para isso será criado o método “checkposition(Position<E> p)” que checa se a posição é válida para a lista, e cria um DNode se a posição for válida.