AC2 - Estrutura de Dados

Integrantes:

- Arthur Silva
- Carolina Gabrielle
- Larissa Ionafa
- Lucas Silva
- Roberta Yumi

Respostas:

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

2. Desenvolva um projeto no GitHub que estenda a classe de testes do TAD Pilha.
//----------OK

3.Desenvolva um projeto no GitHub que implemente os testes do TAD Pilha usando LSE.
//----------OK

4.Exercícios:
a. Crie testes e programas Java que:
//----------OK

b. Inverta os dados de um arranjo usando o TAD Pilha usando LSE.
//----------OK

c. Verifique se parênteses, colchetes e chaves estão corretos numa expressão matemática, por exemplo: [(5 + x)/4 – 2*(y + z)]
i. Correto: ()(( )){([( )])}
ii. Correto: ((( )(( )){([( )])}))
iii. Incorreto:)(( )){([( )])}
iv. Incorreto: ({[])}
v. Incorreto: (
//----------OK

5.Suponha que uma lista inicialmente vazia S tenha executado um total de:
- 25 operações push
- 12 operações top
- 10 operações pop
- 3 das quais geraram StackEmptyExceptions, que foram capturadas e ignoradas.
Qual é o tamanho corrente de S?
Pilha após 25 operações de push
O tamanho da pilha é: 18

//----------OK

6.Se implementarmos a pilha S do problema anterior usando um arranjo, então qual será o valor corrente da variável de instância top?
O valor de corrente de top é: 17

//----------OK

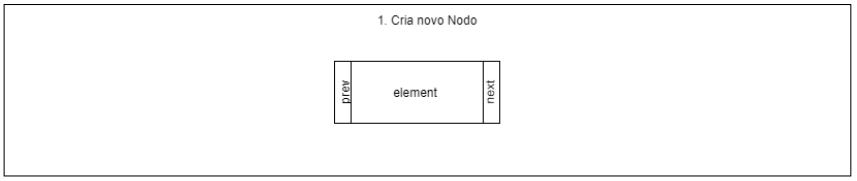
7.Descreva a saída resultante da seguinte série de operações de pilha:
[5]
[5, 3]
[5]
[5, 2]
[5, 2, 8]
[5, 2]
[5]
[5, 9]
[5, 9, 1]
[5, 9]
[5, 9, 7]
[5, 9, 7, 6]
[5, 9, 7]
[5, 9]
[5, 9, 4]
[5, 9]
[5]
//----------OK

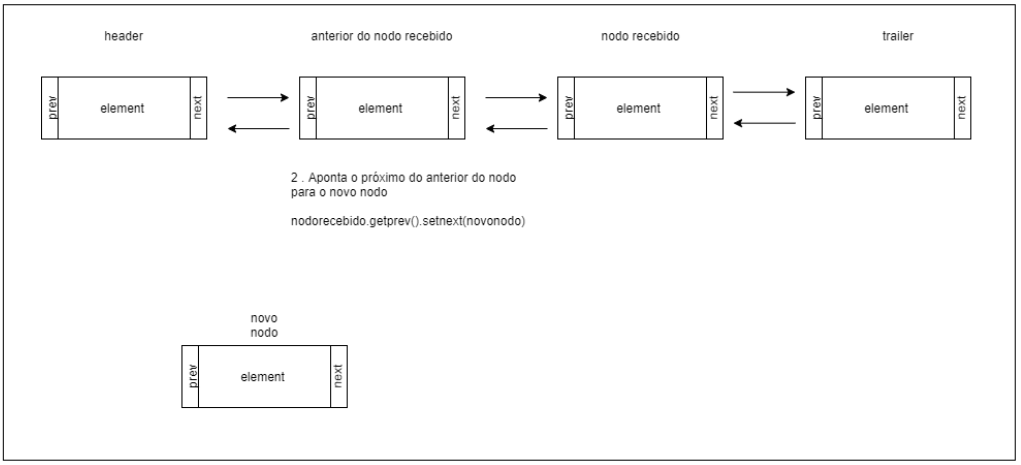
8. Crie os testes e implemente o TAD Fila. Use implementação do TAD Pilha como exemplo.
//----------OK

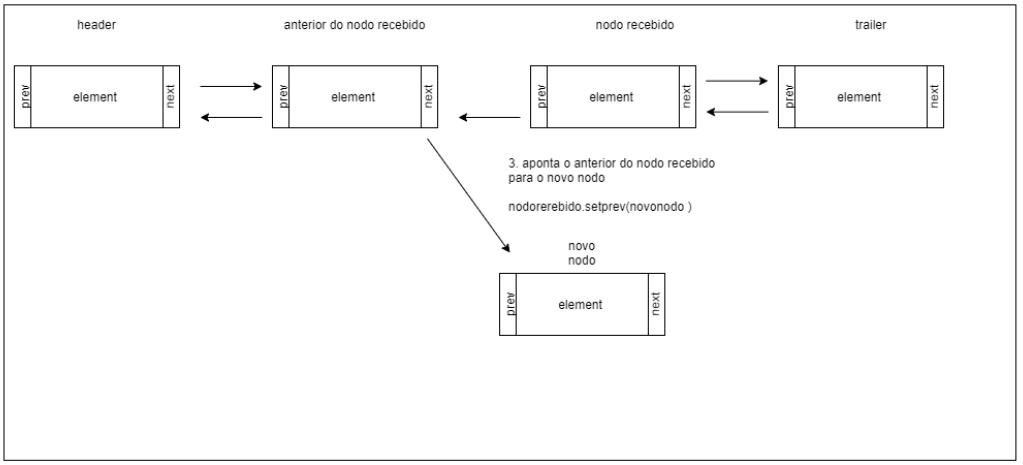
9. Implemente o TAD Fila com base nos testes e no fragmento de implementação de duas operações apresentadas a seguir (Tarefa 13 - TAD-Fila.pptx, slides 19, 20 e 21).
//----------OK

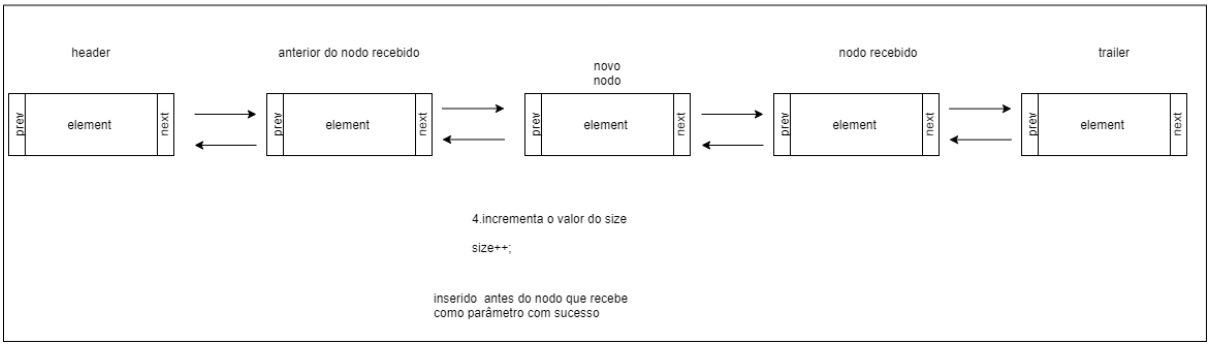
10. Desenhe figuras demonstrando cada um dos passos principais dos métodos do TAD lista de nodos.

addBefore(p, e)

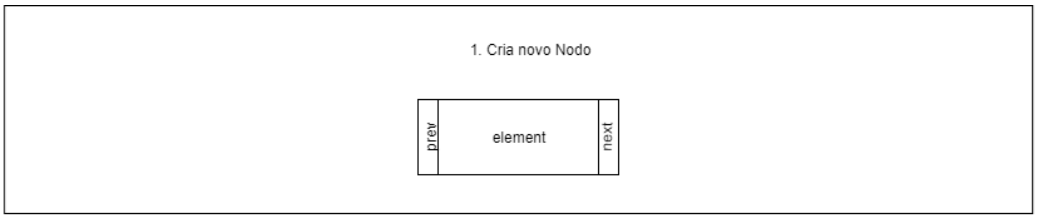


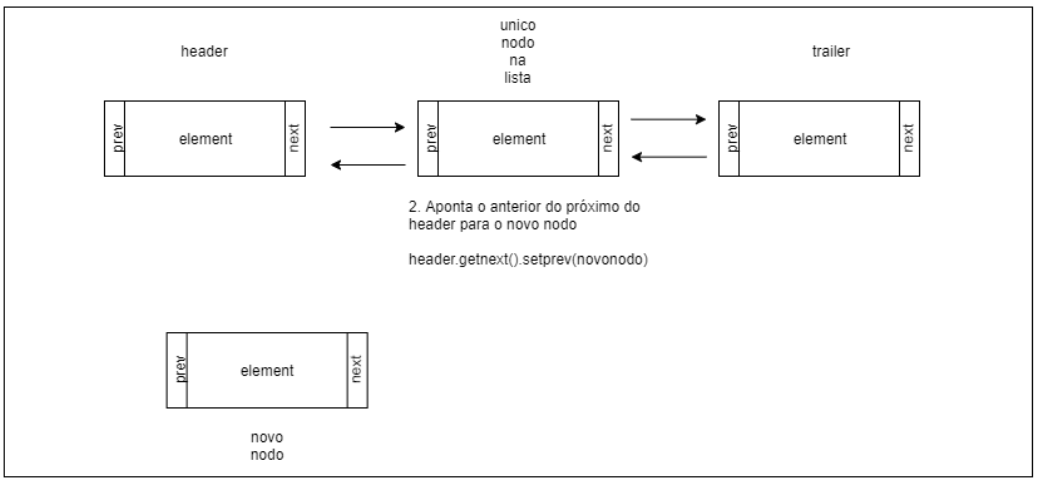


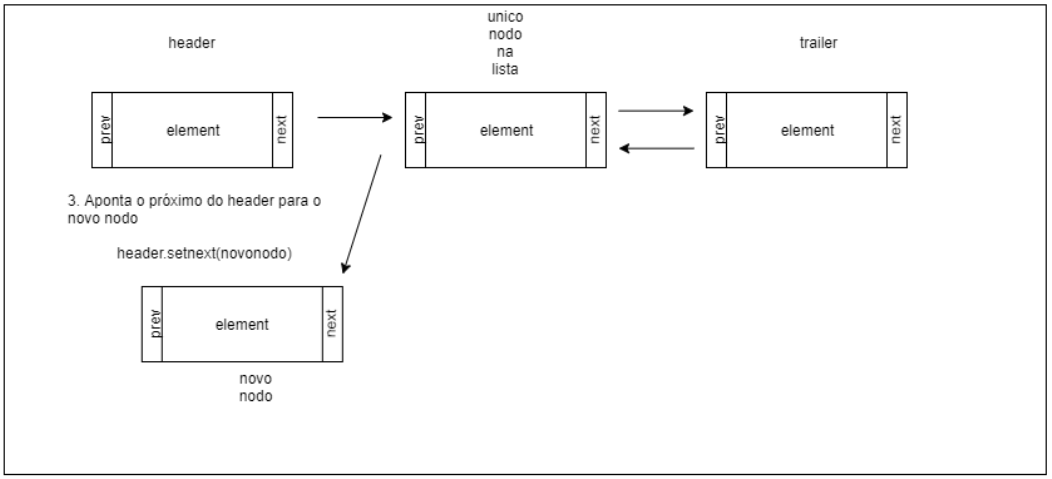


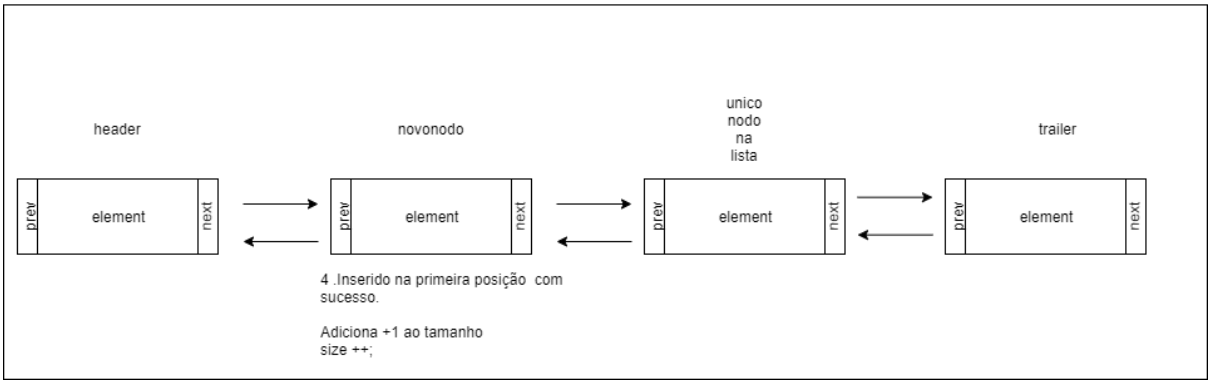


addFirst(e)

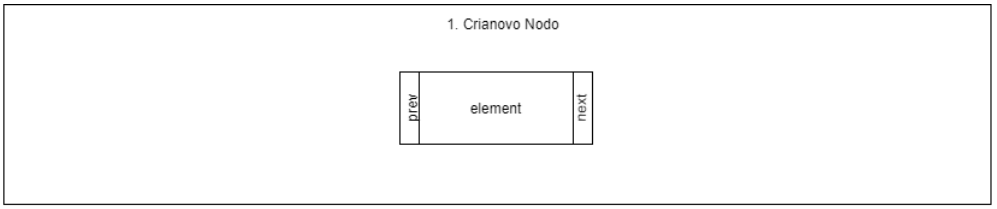

1. Cria novo Nodo
prev
element
next

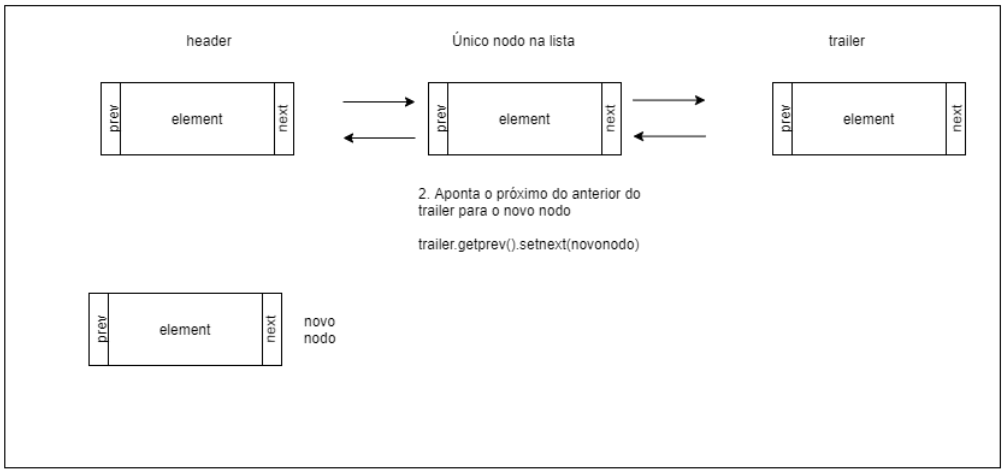

header
unico
nodo
na
lista
trailer
prev
element
next
2. Aponta o anterior do próximo do
header para o novo nodo
header.getnext().setprev(novonodo)
novo
nodo

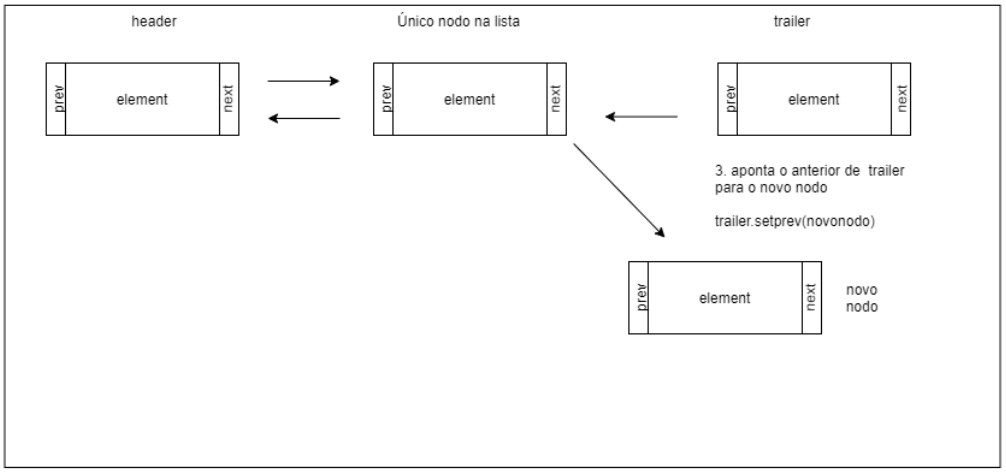

header
unico
nodo
na
lista
trailer
prev
element
next
3. Aponta o próximo do header para o
novo nodo
header.setnext(novonodo)
novo
nodo

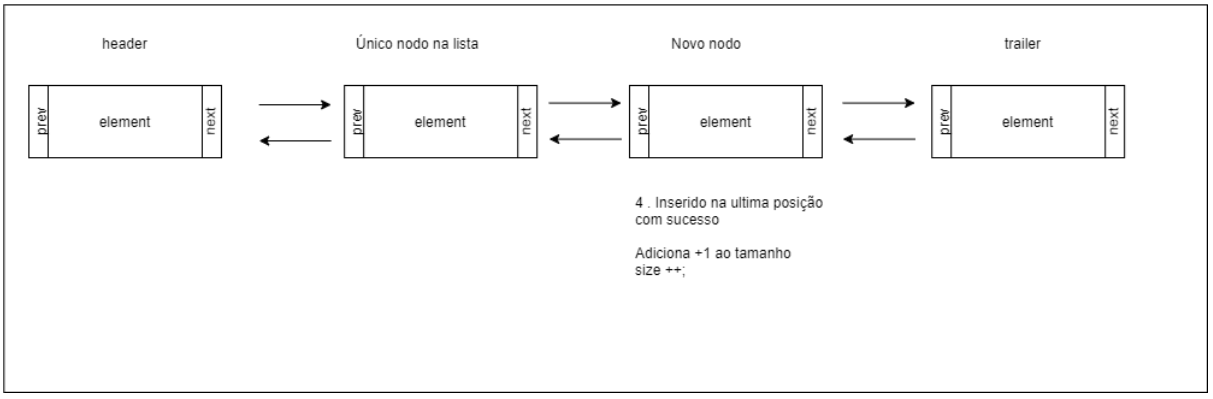

header
novonodo
unico
nodo
na
lista
trailer
prev
element
next
4 .Inserido na primeira posição com
sucesso.
Adiciona +1 ao tamanho
size ++;

addLast(e)









11.Implemente um método não recursivo para inverter uma lista de nodos.

//----------OK

12.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

13.A implementação de NodePositionList não faz verificações de erro para testar se uma dada posição p é realmente membro dessa lista em particular.

a. 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.

b. Descreva como alterar a implementação de NodePositionList de uma forma eficiente que impeça esses maus usos.
//----------OK