Algoritmos e Estrutura de Dados - AC3

Professor Osvaldo Kotaro Takai

- Geovane Donizete Laera.....RA: 1902679
- Isaque Ribeiro dos Santos Junior.....RA: 1903978
- Marcelo Martinez Mesa Campos.....RA: 1905076
- Paulo Ricardo Costa da Silva.....RA: 1905013
- Vinícius da Cruz Pera.....RA: 1903144

1. Implementar e testar o TAD Pilha usando array conforme o que está nos slides.
   R: // ---------- Implementação Java

2. Criar testes para empilhar strings usando o TAD Pilha usando array conforme o que está nos slides.
   R: // ---------- Implementação Java

3. Implementar e testar o TAD Pilha usando LSE conforme o que está nos slides.
   R: // ---------- Implementação Java

4. Implementar e testar um programa que inverte os dados de um arranjo usando o TAD Pilha usando LSE.
   R: // ---------- Implementação Java

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

   R: // ---------- Implementação Java

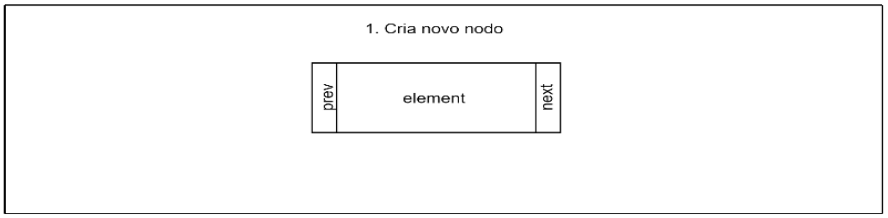
6. Implementar e testar o TAD Fila usando array conforme o que está nos slides.
   R: // ---------- Implementação Java
7. Implementar e testar o TAD Fila usando LSE conforme o fragmento de código que está nos slides.
   R: // ---------- Implementação Java

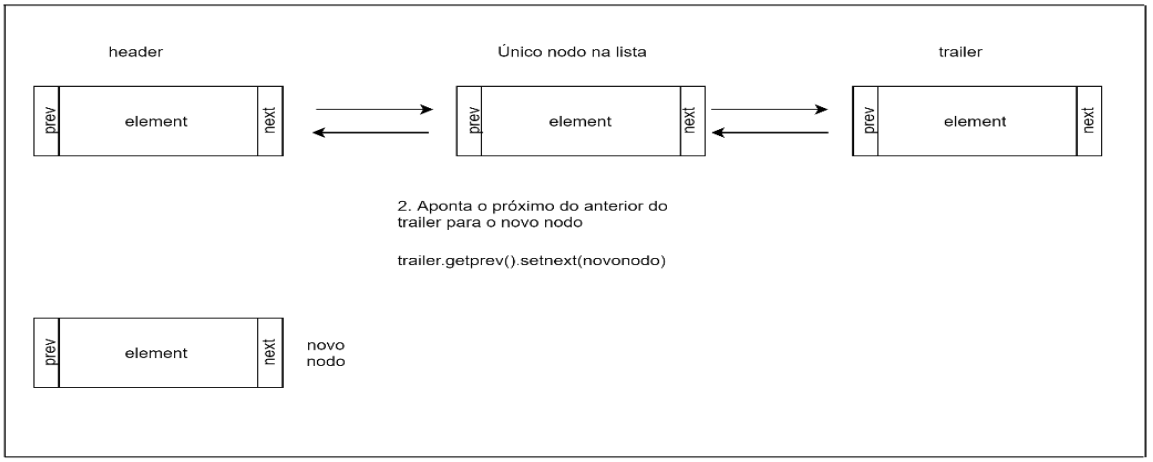
8. Implemente a solução do problema de Josephus conforme os slides.
   R: // ---------- Implementação Java

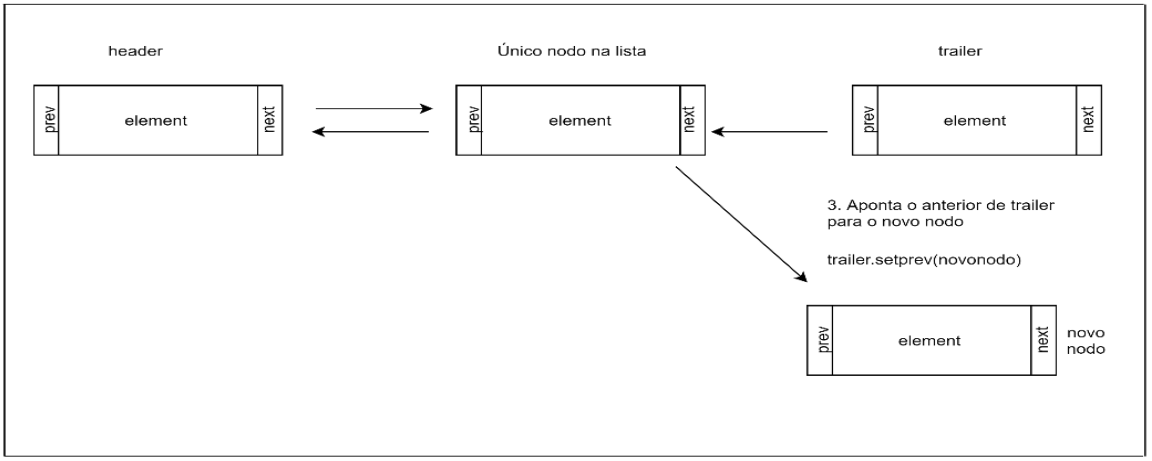
9. Implementar e testar a lista de nodos original conforme o que está nos slides.
   R: // ---------- Implementação Java
10. Desenhe figuras demonstrando cada um dos passos principais dos métodos addBefore(p, e), addFirst(e) e addLast(e) do TAD lista de nodos.

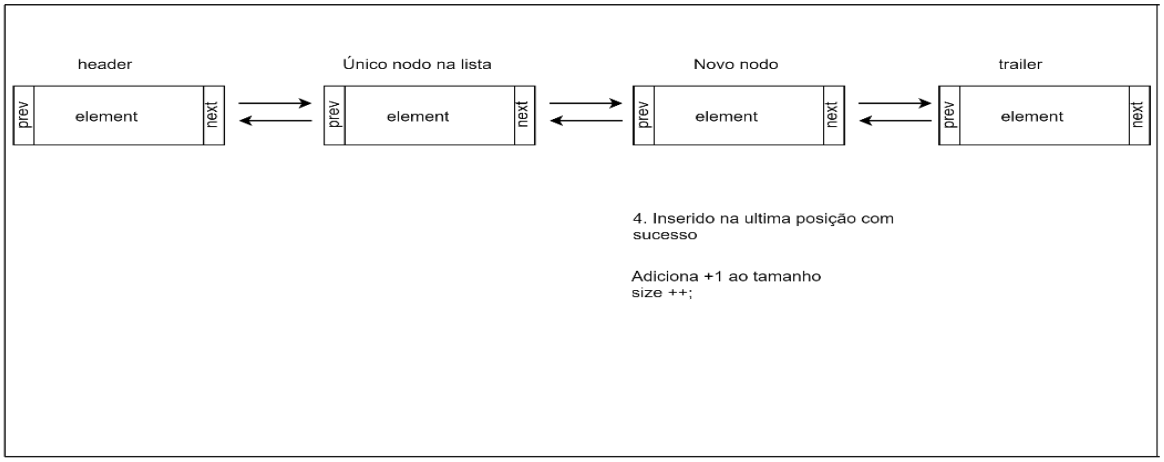
addLast(e)

addLast(e)

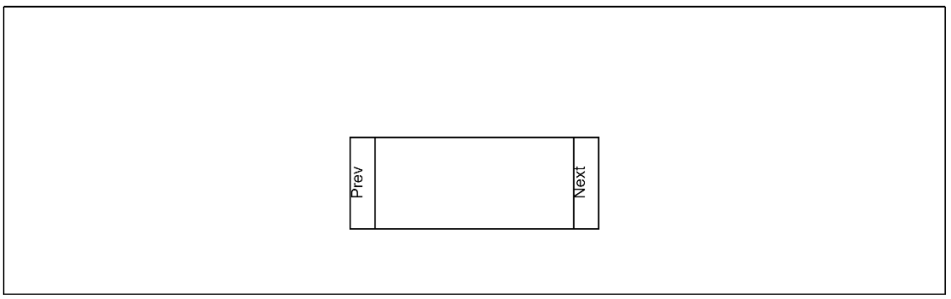


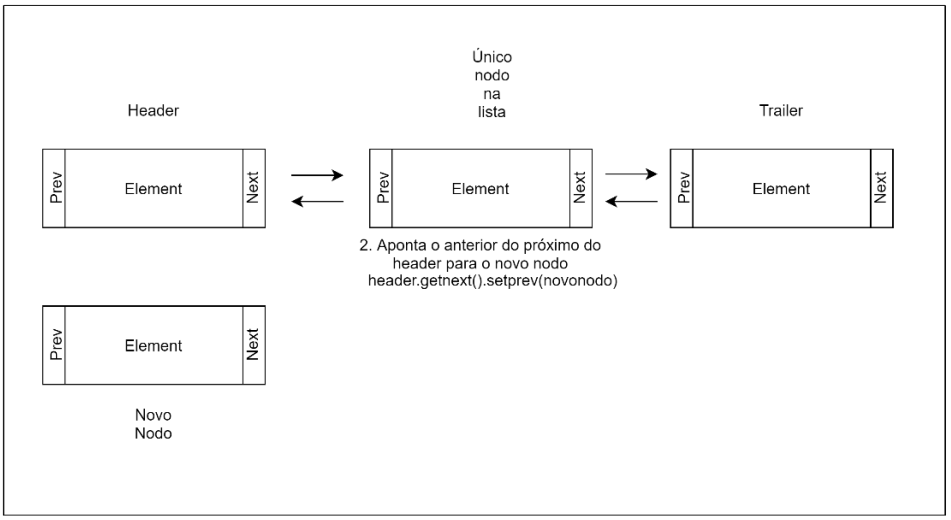


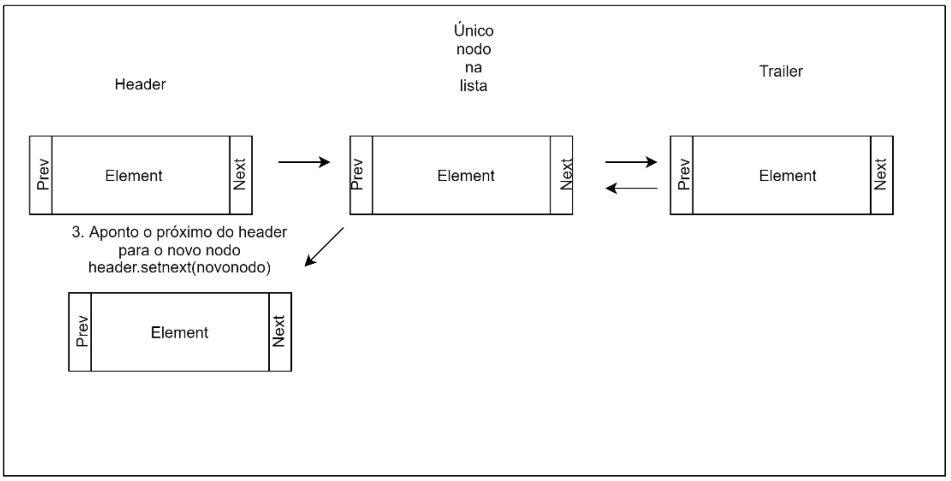


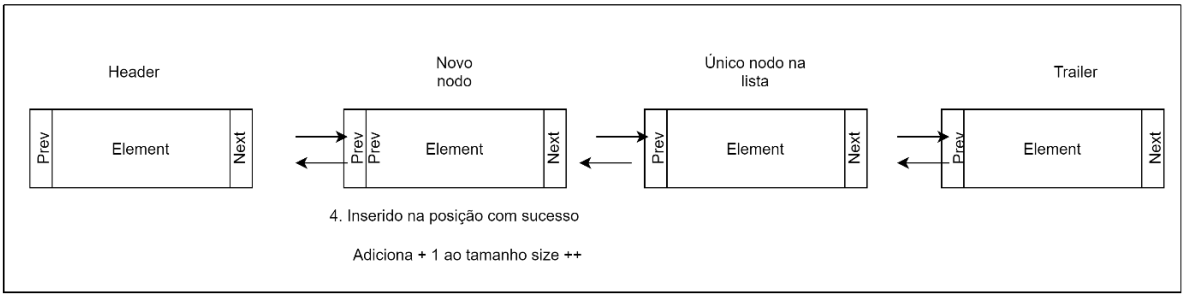


addFirst(e)

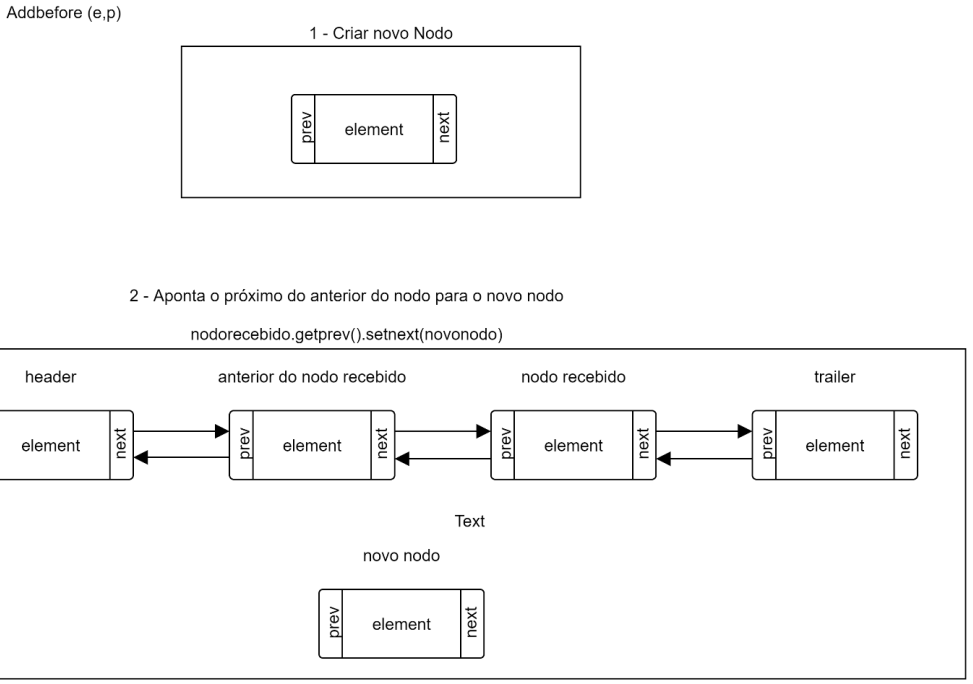

Prev
Next


Header
Único 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
Único nodo na lista
Trailer
Prev
Element
Next
3. Aponto o próximo do header para o novo nodo
header.setnext(novonodo)


Header
Novo nodo
Único nodo na lista
Trailer
Prev
Element
Next
4. Inserido na posição com sucesso
Adiciona + 1 ao tamanho size ++

Addbefore(e,p)



11. Implemente um método não recursivo adicional à implementação do TAD lista de nodos para inverter uma lista de nodos (não esqueça de incluir os testes).
    R: // ---------- Implementação Java

12. Implemente um novo método, makeFirst(p), adicional à implementação do TAD lista de nodos que move o elemento na posição p para a primeira posição, mantendo a ordem relativa dos demais elementos inalterados (não esqueça de incluir os testes).
    R: // ---------- Implementação Java

13. A implementação de NodePositionList não faz verificações de erro para testar se uma dada posição p é realmente membro dessa lista em particular. 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. Descreva como alterar a implementação de NodePositionList de uma forma eficiente que impeça esses maus usos.
    R: Para isso será criado o método "checkPosition(Position<E> p)" que verifica se a posição é valida para a lista, e cria um DNode se a posição for válida. No método checkPosition que verifica se a posição informada é válida, crie um contador para marcar quantas vezes a posição se repete dentro daquela lista após isso. crie um laço que percorrerá a lista e compara se os nodos dentro dela tem tal posição informada semelhante, para cada vez que alguma posição for igual a posição do nodo soma-se um ao contador, se o laço chegar ao final e o contador tiver registrado mais de 0 vezes que a posição se repetiu então a posição é válida. Agora caso o laço chegue a sua última repetição e o contador for igual a zero então a classe de exceção será chamada,indicando que a posição não pertence a lista.
    // ---------- Implementação Java