

Atividade Contínua 4

Disciplina: Estrutura de Dados

Instruções **importantes** antes de iniciar a sua atividade:

Essa atividade vem acompanhada de um avaliador automático para te auxiliar na resolução. Alcançar êxito em todos os testes **não garante que o seu código está 100% correto** ou que você **alcançará a nota máxima**, mas é um indício de que você pode estar no caminho certo. Para a atribuição de nota, **seu código passará por inspeção manual e um segundo corretor independente**, com novos dados de teste.

Caso seu código **não passe nos testes do corretor automático**, é um grande indício de que **ele está errado** e muito provavelmente sua atividade **receberá uma nota baixa**. Qualquer **modificação no código do corretor** será considerada como **fraude**, resultando na **anulação** da AC. Seu código deverá apresentar **solução original e totalmente desenvolvida por você**. Tentativas de **plágio** resultarão na **anulação** da AC.

Sobre a entrega do código:

Seu código **deverá ser entregue no Classroom** como um arquivo .py (Python). **O nome do arquivo deve conter o seu nome completo**. Por exemplo:



Você também deve informar o seu **número de RA** e **nome completo** nos campos **NUMERO_RA** e **NOME_COMPLETO** como *strings*, dentro do programa. Por exemplo:

```
## =====  
## =====  
  
# INFORME SEUS DADOS (como string nas variáveis a seguir):  
NUMERO_RA = "1234567"  
NOME_COMPLETO = "Rafael Will Macedo de Araujo"  
# OBS: Não informar seus dados corretamente implicará em nota 0.0 (zero)!
```

Códigos sem nome completo e/ou RA receberão nota ZERO!

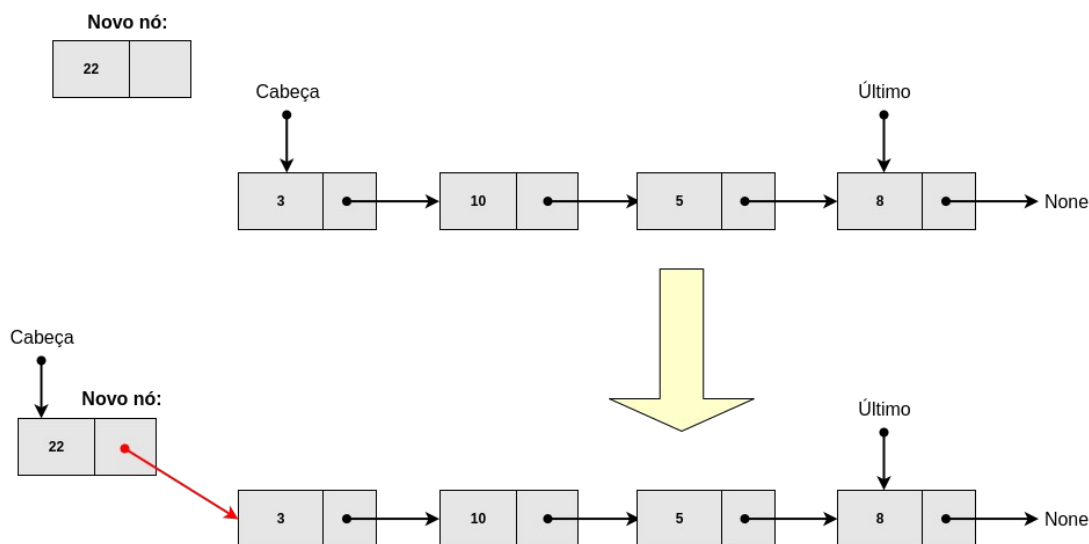
Não esqueça de Enviar a sua solução, caso contrário será considerada não entregue.

Faça o que se pede:

Dada a classe *ListaSimplificada*, um TAD que implementa algumas operações básicas de uma lista encadeada, faça o que se pede nas duas questões a seguir:

1) Complete o código do método ***adicionar_na_frente(item)***, que recebe um item qualquer, encapsula-o em um novo nó e adiciona esse nó **na frente da lista**. Você deve fazer as **alterações necessárias nos atributos: *cabeça*, *ultimo*, *tamanho***.

Por exemplo: suponha que desejamos adicionar o item 22 na frente da lista. Teríamos o seguinte (antes e depois):



2) Complete o código do método ***count(alvo)***, que recebe um valor por parâmetro, e conta quantas ocorrências desse valor ocorrem na lista. Esse método sempre devolve (retorna) um número inteiro, que é o número de ocorrências do alvo na lista.

Dica: você terá que percorrer toda a lista encadeada, a partir da cabeça, para fazer a contagem.

Observação: a classe *ListaSimplificada* já possui 3 atributos: *cabeça*, *ultimo*, *tamanho*. Também existem 2 métodos prontos (já implementados) para te ajudar a testar o seu código: *imprime_lista()*, *append(item)*.

Você não pode criar atributos ou métodos adicionais. Isto é, você só deve implementar os 2 métodos pedidos, sem criar novos atributos e nem novos métodos.