

PRÁTICA 11

1) Implementar os métodos adicionais na classe ListaLigada:

a) método void Concatena(ListaLigada lista)

recebe um objeto da classe Lista Ligada, representando uma 2ª lista ligada, e concatena essa lista recebida com a lista em questão (Concatenar é acrescentar a 2ª lista no final da 1ª lista).

Por exemplo: suponha que exista um objeto lista1 da classe ListaLigada com os valores:

head(da lista1) → 10 → 7 → 15 → 2

Suponha que exista um outro objeto lista2 com os valores:

head(da lista2) → 5 → 20 → 8

Então a chamada lista1.concatena(lista2) terá como resultado:

head(da lista1) → 10 → 7 → 15 → 2 → 5 → 20 → 8

b) método bool Compara(ListaLigada lista)

recebe um objeto da classe Lista Ligada, representando uma 2ª lista ligada, e compara essa 2ª lista com a lista em questão, devolvendo true caso as listas tenham o mesmo conteúdo e false caso contrário.

c) método void Inverte()

inverte a lista encadeada, fazendo o primeiro nó ser o último, o segundo nó ser o penúltimo, e assim por diante, mas sem criar novos nós, ou seja, apenas invertendo os ponteiros.

Por exemplo: suponha que exista um objeto lista1 da classe ListaLigada com os valores:

head(da lista1) → 10 → 7 → 15 → 2

Então a chamada lista1.inverte() produzirá o resultado:

head(da lista1) → 2 → 15 → 7 → 10

Obs.: as versões recursivas abaixo deverão ter um parâmetro adicional: referência do primeiro nó da lista

d) versão recursiva do método Imprime

e) versão recursiva do método BuscaNode

f) versão recursiva do método RemoveNode

g) versão recursiva do método Tamanho

2) Testar os métodos implementados acima.

3) Criar a classe ListaLigadaOrdenada, herdeira de ListaLigada, representando uma lista encadeada que tem seus dados em ordem crescente, a partir do nó apontado pelo head.

- a) Reescrever o método InserirNode, para inserir o novo nó de forma a manter a lista em ordem crescente.
Dica: para implementar esse método, é bom ter 2 variáveis do tipo Node para percorrer a lista: ant (endereço do nó anterior) e atual (endereço do nó atual). E também uma variável bool inseriu, que indica se já inseriu o nó ou não.
- b) Reescrever o método BuscaNode, para buscar o elemento na lista ordenada (qdo encontrar um valor maior do que o procurado, já pode parar a busca)
- c) Reescrever o método RemoveNode
- d) Implementar o método ListaLigadaOrdenada Merge (ListaLigadaOrdenada lista)
Recebe um objeto da classe ListaLigadaOrdenada, representando uma 2ª lista ligada, e produz uma 3ª lista, contendo a intercalação da 1ª lista com a 2ª lista ligada.

4) Teste a classe ListaLigadaOrdenada.