

Lista de Exercícios 2 – Listas, Pilhas e Filas

Considerando as classes *ListaEncadeada*, *Pilha* e *Fila* (na forma vetor e encadeamento) vistas em aula, resolva os exercícios a seguir. Não se esqueça de prever as exceções que forem necessárias. A lista pode ser feita por grupos de até 3 alunos e deve ser entregue em 08/11/2002.

1) Dada uma lista encadeada de caracteres formada por uma seqüência alternada de letras e dígitos, construa um método que retorne uma lista na qual as letras são mantidas na seqüência original e os dígitos são colocados na ordem inversa. Exemplos:

A 1 E 5 T 7 W 8 G → A E T W G 8 7 5 1

3 C 9 H 4 Q 6 → C H Q 6 4 9 3

Como mostram os exemplos, **as letras devem ser mostradas primeiro, seguidas dos dígitos**. Sugestões:

- usar uma fila e uma pilha;
- supor um método *ehDigito()* *retorna booleano* que retorna verdadeiro caso um caractere seja um dígito.

2) Um SGBD executa transações de aplicações por ordem de chegada, a menos que uma aplicação tenha maior prioridade que outra (*o SGBD atende 5 aplicações, tendo cada uma um código (0-4) e um valor de prioridade*). Escreva um **método da classe SGBD** que recebe um conjunto de “n” transações (*cada transação mantém o código da aplicação, um número seqüencial (número de ordem da transação solicitada pela aplicação) e um programa*) e organiza uma fila de execução destas transações. Transações podem ser inseridas na fila enquanto ela tiver menos que 10 elementos (tamanho máximo da fila). O SGBD começa a executar as transações da fila (exclusão da fila) sempre que a fila alcançar o tamanho máximo ou quando todas as “n” transações estiverem na fila.

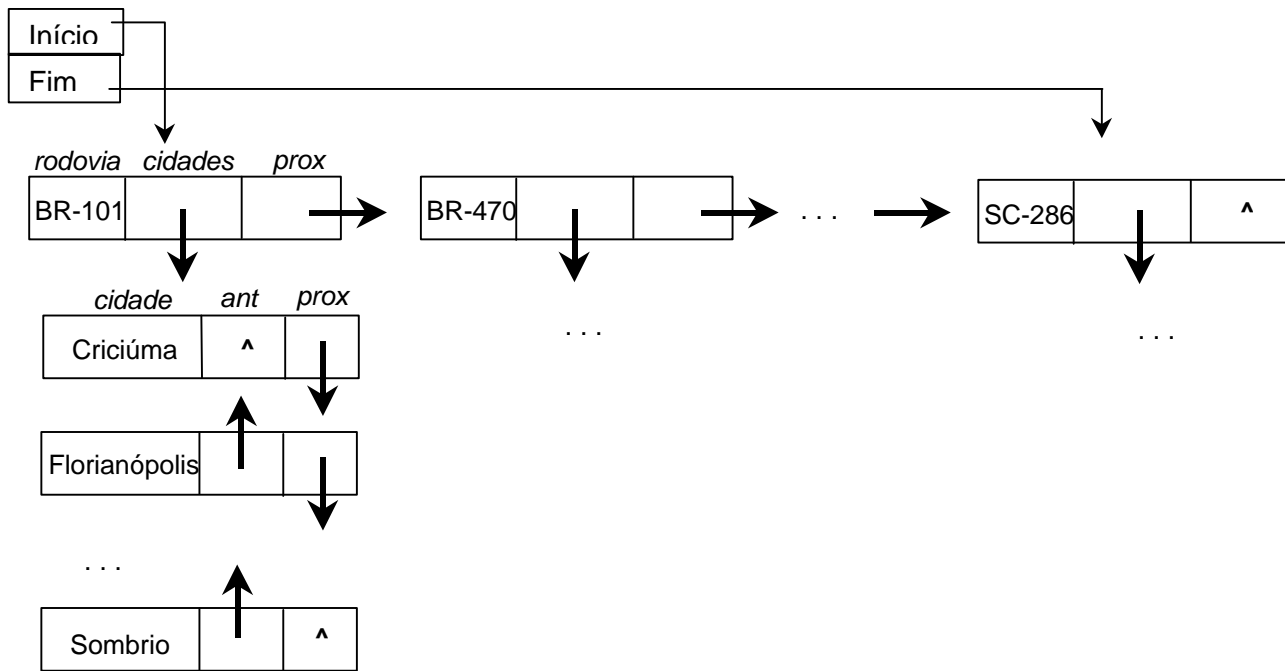
Observações:

- as prioridades das aplicações podem ser mantidas em um vetor
- o conjunto de transações de entrada pode ser uma lista
- definir as classes que julgares necessárias
- os métodos definidos para listas e filas podem ser utilizados
- supor que existe um método da classe SGBD chamado *ExecutaProgramaTransação(programa string)*

3) Construa um método que recebe uma lista encadeada de números inteiros e retorna uma lista **sem repetições**, ou seja, uma lista onde cada número apareça apenas uma vez. Exemplo:

12 5 -7 8 5 9 12 1 8 → 12 5 -7 8 9 1

4) Uma estrutura de dados mantém informações sobre rodovias do estado de SC e as cidades pelas quais uma rodovia passa. Um exemplo desta estrutura é mostrado abaixo:



Faça o seguinte:

- defina as classes que julgares necessárias para implementar esta estrutura;
- implemente, sem utilizar nenhum método visto em aula, um método *insereCidade(nomeRodovia string, nomeCidade string)* que insere uma cidade na lista de cidades de uma rodovia, mantendo sempre ordenada a lista de cidades;
- implemente um método *rodoviasCidade(nomeCidade string)* que retorna uma lista encadeada com os nomes de todas as rodovias que passam pela cidade *nomeCidade*;
- implemente um método *Cruzamento(nomeRodovia1 string, nomeRodovia2 string)* que retorna verdadeiro se as duas rodovias se cruzam em alguma cidade, ou falso, caso contrário. Considere que as listas de cidades estão ordenadas!

5) Utilizando uma pilha, escreva um método que receba um número inteiro positivo no formato decimal e **converte** este número para o **formato binário**. Exemplos:

5	→	101
13	→	1101
1	→	1

O método retorna uma lista encadeada com os dígitos que fazem parte do número em binário. Considere $x \text{ DIV } y$ e $x \text{ MOD } y$ o quociente e o resto da divisão entre dois números inteiros x e y , respectivamente, na linguagem algorítmica.