



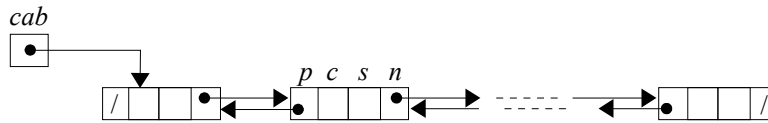
-
- 1) Questões sobre conceitos:
- O que é uma lista?
 - O que caracteriza uma pilha?
 - O que caracteriza uma fila?
 - É correto dizer que uma pilha e uma fila são listas com restrições? Explique.
- 2) Considerando as classes Lista, Pilha e Fila e os seus métodos:
- listaVazia()** – retorna verdadeiro se a lista estiver vazia e falso, caso contrário;
 - listaCheia()** – retorna verdadeiro se a lista estiver cheia (falta de memória) e falso, caso contrário;
 - inserir(D)** – insere o elemento D na lista de forma ordenada;
 - incluir(D, P)** – insere o elemento D na posição P da lista;
 - retirar(P)** – retorna o elemento da posição P da lista, eliminando-o da mesma;
 - excluir(D)** – retorna a posição do elemento D na lista ou 0 (zero), caso não exista D, eliminando-o da mesma;
 - consultar(P)** – retorna o elemento da posição P da lista;
 - buscar(D)** – retorna a posição do elemento D na lista ou 0 (zero), caso não exista D;
 - pilhaVazia()** – retorna verdadeiro se a pilha estiver vazia e falso, caso contrário;
 - pilhaCheia()** – retorna verdadeiro se a pilha estiver cheia (falta de memória) e falso, caso contrário;
 - empilhar(D)** – insere o elemento D no topo da pilha;
 - desempilhar()** – retorna o elemento do topo da pilha, eliminando-o da mesma;
 - verTopo()** – retorna o elemento do topo da pilha;
 - filaVazia()** – retorna verdadeiro se a fila estiver vazia e falso, caso contrário;
 - filaCheia()** – retorna verdadeiro se a fila estiver cheia (falta de memória) e falso, caso contrário;
 - enfileirar(D)** – insere o elemento D no fim da fila;
 - desenfileirar()** – retorna o elemento da frente da fila, eliminando-o da mesma;
 - verFrente()** – retorna o elemento da frente da fila;
- considerando, ainda, que cada objeto ao ser criado (declarado) é uma estrutura vazia, implemente a partir somente dos métodos descritos, trechos de programa para realizar as seguintes operações:
- inverter uma pilha;
 - inverter uma lista;
 - inverter uma fila;
 - fazer a união (de conjuntos) entre duas listas;
 - fazer a intersecção (de conjuntos) entre duas listas;
 - fazer a diferença (de conjuntos) entre duas listas;
 - fazer a união (de conjuntos) entre duas pilhas;
 - fazer a intersecção (de conjuntos) entre duas pilhas;
 - fazer a diferença (de conjuntos) entre duas pilhas;
 - fazer a união (de conjuntos) entre duas filas;
 - fazer a intersecção (de conjuntos) entre duas filas;
 - fazer a diferença (de conjuntos) entre duas filas;
 - concatenar duas pilhas;
 - concatenar duas filas;
 - concatenar duas listas;
 - intercalar duas pilhas;
 - intercalar duas filas;
 - intercalar duas listas.
- OBS:
- Considere as listas não ordenadas
 - Procure manter, quando possível, a mesma ordem das estruturas originais.
- 3) Considerando o estabelecido na questão anterior e, ainda, que todas as estruturas armazenam tipo inteiro, como ficarão os conteúdos da lista *list*, da pilha *stack* e da fila *queue*, inicialmente vazias, se forem executadas as seguintes sequências de instruções:
- list.inserir(5);**
 - list.inserir(2);**
 - list.inserir(10);**

```

stack.empilhar(list.excluir(10));
stack.empilhar(list.excluir(20));
queue.enfileirar(list.excluir(10));
queue.enfileirar(stack.desempilhar());
list.inserir(queue.desenfileirar());

```

- 4) Observe o esquema a seguir:



Considere que *cab*, atributo privado da classe *ListaDupla*, aponta para um nó bobo, e que cada nó é um objeto da classe *Celula*, onde seus atributos são públicos e estão indicados na figura, sendo que *c* é do tipo *int* e *s* é do tipo *float*. Desenvolva um método retorne a quantidade de valores iguais a um determinado valor *float* passado como argumento.

- 5) Considerando a questão anterior, desenvolva um método que retorne a quantidade de elementos (células) pertencentes à lista.
- 6) Ainda referente ao esquema da questão 4, desenvolva um método para inserir uma nova célula no fim da lista.
- 7) Existe algum limite de tamanho para uma lista encadeada? Se sim, qual é esse limite?
- 8) Cite uma vantagem e uma desvantagem da lista encadeada em relação à lista implementada em vetor.
- 9) Cite aplicações para as estruturas pilha, fila e lista.
- 10) Considere uma estrutura encadeada que armazena as cartas de um baralho comum sem coringa. Se cada carta é um objeto da classe *Carta*, faça as declarações dos atributos necessários a esta classe, considerando as informações essenciais que tratam do valor, naipe, cor e posição (frente/verso). Se achar necessário, pode acrescentar outros atributos como, por exemplo, o símbolo correspondente ao valor da carta. Para cada tipo declarado, coloque um comentário esclarecendo a escolha do mesmo.
- 11) Refaça as questões 4, 5 e 6 considerando a lista sem o nó bobo.
- 12) Refaça as questões 4, 5 e 6 considerando a lista circular com o nó bobo.
- 13) Refaça as questões 4, 5 e 6 considerando a lista circular sem o nó bobo.