

Exercício: TADs e Listas

Objetivos: Exercitar os conceitos de TADs e Listas.

Data da Entrega: 28/03/2017

OBS 1: Exercício Individual.

OBS 2: A entrega da lista deverá ser executada utilizando-se um repositório de código. O estudante deverá criar uma conta no bitbucket e dar acesso os docentes (monteiro@dc.ufc.br, javam@dc.ufc.br e jose.macedo@dc.ufc.br). Crie um projeto para cada lista de exercícios.

OBS 3: Esta lista deverá ser solucionada utilizando-se a linguagem Scala.

NOME: _____ MATRÍCULA: _____

Questão 1

Implemente um TAD para representar uma lista encadeada capaz de armazenar valores inteiros. Esse TAD deverá conter funções para:

1. Criar uma lista vazia;
2. Inserir elemento no início;
3. Imprimir os valores armazenados na lista;
4. Imprimir os valores armazenados na lista usando recursão;
5. Imprimir os valores armazenados na lista em ordem reversa (da cauda para a cabeça da lista);
6. Verificar se a lista está vazia (retorna 1 se vazia ou 0 se não vazia);
7. Recuperar/Buscar um determinado elemento da lista;
8. Remover um determinado elemento da lista;
9. Remover um determinado elemento da lista usando recursão;
10. Liberar a lista;

Implemente um programa que cria uma variável do TAD lista encadeada, criado anteriormente, e utiliza cada uma de suas funções.

Questão 2

Implemente um TAD para representar uma lista encadeada ordenada capaz de armazenar valores inteiros. Esse TAD deverá conter funções para:

1. Criar uma lista vazia;
2. Inserir elemento;
3. Imprimir os valores armazenados na lista;
4. Imprimir os valores armazenados na lista usando recursão;
5. Imprimir os valores armazenados na lista em ordem reversa;
6. Verificar se a lista está vazia (retorna 1 se vazia ou 0 se não vazia);
7. Recuperar/Buscar um determinado elemento da lista;
8. Remover um determinado elemento da lista;
9. Remover um determinado elemento da lista usando recursão;
10. Liberar a lista;
11. Verificar se duas listas são iguais;

Implemente um programa que cria uma variável do TAD lista encadeada ordenada, criado anteriormente, e utiliza cada uma de suas funções.

Questão 3

Implemente um TAD para representar uma lista duplamente encadeada e ordenada capaz de armazenar valores inteiros. Esse TAD deverá conter funções para:

1. Criar uma lista vazia;
2. Inserir elemento;
3. Imprimir os valores armazenados na lista;
4. Imprimir os valores armazenados na lista usando recursão;
5. Imprimir os valores armazenados na lista em ordem reversa;
6. Verificar se a lista está vazia (retorna 1 se vazia ou 0 se não vazia);
7. Recuperar/Buscar um determinado elemento da lista;
8. Remover um determinado elemento da lista;
9. Remover um determinado elemento da lista usando recursão;
10. Liberar a lista;
11. Verificar se duas listas são iguais;

Implemente um programa que cria uma variável do TAD lista duplamente encadeada e ordenada, criado anteriormente, e utiliza cada uma de suas funções.

Questão 4

Implemente um TAD para representar uma lista encadeada circular (simplesmente encadeada) capaz de armazenar valores inteiros. Esse TAD deverá conter funções para:

1. Criar uma lista vazia;
2. Inserir elemento;
3. Imprimir os valores armazenados na lista;
4. Imprimir os valores armazenados na lista usando recursão;
5. Verificar se a lista está vazia (retorna 1 se vazia ou 0 se não vazia);
6. Recuperar/Buscar um determinado elemento da lista;
7. Remover um determinado elemento da lista;
8. Remover um determinado elemento da lista usando recursão;
9. Liberar a lista;

Implemente um programa que cria uma variável do TAD lista encadeada circular (simplesmente encadeada), criado anteriormente, e utiliza cada uma de suas funções.

Questão 5

Implemente um TAD para representar uma lista encadeada circular (duplamente encadeada) capaz de armazenar valores inteiros. Esse TAD deverá conter funções para:

1. Criar uma lista vazia;
2. Inserir elemento;
3. Imprimir os valores armazenados na lista;
4. Imprimir os valores armazenados na lista usando recursão;
5. Verificar se a lista está vazia (retorna 1 se vazia ou 0 se não vazia);
6. Recuperar/Buscar um determinado elemento da lista;
7. Remover um determinado elemento da lista;
8. Remover um determinado elemento da lista usando recursão;
9. Liberar a lista;

Implemente um programa que cria uma variável do TAD lista encadeada circular (duplamente encadeada), criado anteriormente, e utiliza cada uma de suas funções.

Questão 6

Implemente TADs para:

1. Representar uma conta bancária, onde cada conta bancária possui um número e um saldo, além de funções para efetuar crédito e efetuar débito.
2. Representar uma conta poupança, onde cada conta poupança possui um número e um saldo, além de funções para efetuar crédito, efetuar débito e render juros.
3. Representar uma conta fidelidade, onde cada conta fidelidade possui um número, um saldo e um bônus, além de funções para efetuar crédito, efetuar débito e render bônus. Em uma conta fidelidade, a cada operação de crédito realizado, 1% do valor de creditado é acumulado ao bônus da conta. A função render bônus deve adicionar o valor do bônus no saldo da conta e, em seguida, zerar o valor do bônus.

Implemente um TAD para representar uma lista encadeada ordenada capaz de armazenar contas bancárias, contas poupanças ou contas bônus . Esse TAD deverá conter funções para:

1. Criar uma lista vazia;
2. Inserir elemento no início;
3. Imprimir os valores armazenados na lista;
4. Imprimir os valores armazenados na lista usando recursão;
5. Imprimir os valores armazenados na lista em ordem reversa (da cauda para a cabeça da lista);
6. Verificar se a lista está vazia (retorna 1 se vazia ou 0 se não vazia);
7. Recuperar/Buscar um determinado elemento da lista;
8. Remover um determinado elemento da lista;
9. Remover um determinado elemento da lista usando recursão;
10. Liberar a lista;

Implemente um programa que cria uma variável do TAD lista encadeada, criado anteriormente, e permite ao usuário realizar as seguintes operações:

1. Inserir uma conta bancária;
2. Inserir uma conta poupança;
3. Inserir uma conta fidelidade;
4. Realizar crédito em uma determinada conta;
5. Realizar débito em uma determinada conta;
6. Consultar o saldo de uma conta;
7. Consultar o bônus de uma conta fidelidade;
8. Realizar uma transferência entre duas contas;
9. Render juros de uma conta poupança;
10. Render bônus de uma conta fidelidade;
11. Remover uma conta;
12. Imprimir número e saldo de todas as contas cadastradas;