

T1 - Listas Encadeadas

Objetivo

O objetivo deste trabalho é implementar uma aplicação de lista telefônica em “modo console”, com o objetivo de revisar como funciona a implementação de estruturas lineares não-contíguas em memória. A implementação deste trabalho deve levar em consideração a possibilidade de utilizar-se diversos tipos de dados diferentes como conteúdo da lista encadeada.

Tarefas Obrigatórias

1. Implementar uma aplicação de lista telefônica.
2. A aplicação deve, ao ser inicializada ler um arquivo do disco com os nomes e telefones dos contatos da agenda.
3. Deve ser permitido ao usuário inserir e remover novos contatos.
4. Novos contatos devem ser gravados sempre no final do arquivo de dados.
5. Para navegar na lista de contatos, o usuário pode:
 1. Digitar a primeira letra do nome.
 2. Avançar para o próximo contato.
 3. Voltar para o contato anterior.
6. A lista de contatos deve ser exibida em ordem alfabética para o usuário.
7. Não podem ser utilizadas classes do pacote `java.util`, com exceção da classe `java.util.Scanner`.
8. O arquivo de dados deve conter pelo menos 20 registros.
9. O sistema não deve impor um número máximo de registros, no entanto, deve se proteger de problemas como “falta de memória”.
10. Divida as classes em pacotes lógicos (estruturas, modelo, programa, aplicações, etc).

Data de Entrega

13 de Março de 2014.

Dicas

Você pode utilizar todo o código visto em aula para listas encadeadas.

Não esqueça que a entrega do trabalho será realizada através do seu repositório no Github. O repositório deve ter o mesmo nome do repositório criado no Exercício 1 (E1), disponível no Portal do Aluno.

Pode ser necessário trabalhar com duas ou mais listas encadeadas, principalmente para resolver os “pontos extras”.

Objetivos de Aprendizado

O principal objetivo deste trabalho é revisar como se implementa uma lista encadeada de nodos. Entre os objetivos deste trabalho também se incluem:

- Trabalhar as relações entre classes
- Gerenciamento de registros em arquivos
- Implementar estruturas reutilizáveis.
- Trabalhar dentro de limitações de uma ferramenta ou projeto.

Avaliação

Será avaliada a conclusão das tarefas **obrigatórias** do trabalho.

Conceitos melhores podem ser obtidos demonstrando qualidade no código, nos registros de **commit**, e na **documentação** do código.

Para obter o conceito máximo, todos os “pontos extras” devem ser implementados.

Pontos Extras

1. Implemente um programa de geração de nomes e telefones para a criação de arquivos de dados grandes.
2. O arquivo de dados não pode sofrer alteração que não seja a inclusão de novos elementos no final do arquivo.
3. A implementação de lista encadeada permite a utilização da mesma em outras aplicações, com outras classes como chave e/ou informação.

Observações

Comente o código. Utilize o javadoc para manter o código documentado.

Utilize **comentários nos “commits”** realizados para manter o histórico de alterações. Um programa deste tamanho não deve ser feito em “um único commit”. **Commits parciais** permitem que “correções de rumo” sejam feitas a tempo.