Universidade de Évora

Engenharia Informática

Estrutura de Dados e Algoritmos I

Agenda telefónica



Daniel Serrano 35087

Vasco Crespo 37913

Docente: Lígia Ferreira

16 de Janeiro de 2018

## Introdução

A partir da implementação de todas as estruturas de dados decidimos criar a nossa agenda a partir quase exclusivamente da estrutura de dados AVL pois consideramos que as AVL’s são a estrutura mais eficiente quando se quer pesquisar por algo, usámos também uma Double Linked List para representar os números de telemóvel de um contacto pois um contacto pode ter vários números mas apenas um nome associado e também uma queue para a resolução de um dos funcionalidades da agenda onde se pretende passar todos os contactos para essa queue temporária de modo a que sejam devolvidos todos os contactos da agenda.

## Desenvolvimento

Considerámos desde inicio que a própria agenda iria ser uma AVL pelo que temos uma classe ‘Agenda’ que contém métodos simples integrando os métodos implementados das AVL, criando e utilizando uma AVL de nome agendaAvl para resolver os problemas, sendo os métodos da agenda são os seguintes:

* public Contacto getContacto(int x):

Inserindo um número inteiro, este método irá procurar se existe algum contacto no índice da AVL desse inteiro e retornará o contacto desse índice x.

* public boolean isEmpty():

Este método utiliza o próprio método isEmpty() das AVL’s e retornará ‘true’ caso a Agenda esteja vazia e ‘false’ caso não esteja vazia.

* public String getContactoStringWithNumber(int num):

Ao inserir um certo número inteiro este método retorna o contacto onde esse número se encontra numa string.

* public boolean containsNome(String nome):

Inserindo um certo nome em forma de string verifica se esse nome existe na agenda e se sim, o método retorna true

* public boolean containsNum(int num):

Inserindo um número inteiro verifica se esse número existe na agenda, retornando true caso exista.

* public void setNumero(String nome, int i, int umNovo):

Dando um nome já existente na agenda, este método verifica o contacto que tem esse nome e troca o número associado a esse contacto pelo numNovo.

* public int sizeContacto(String nome):

Utilizando o método sizeNum() da classe Contacto que cria um contacto com um nome e com uma LinkedList como numero para poder ter vários números associados ao mesmo nome ou neste caso, ao mesmo Contacto, este método retorna o número de números que está associado ao nome inserido.

* public String showNumeros(String nome):

Este método é parecido com o método anterior só que em vez de retornar o número de números associados a um certo nome, retorna esses números em especifico. Verifica todos os contactos da agenda e retorna os números associados ao nome dado como input.

* public void setNome(String nomeAnt, String nomeNovo):

Este método percorre todos os contactos da agenda e quando encontrar o contacto que tem como nome nomeAnt, muda esse nome para o nomeNovo que inserimos.

* public void addContacto(String nome, int numero):

Este método percorre mais uma vez todos os contactos da agenda até encontrar o que foi inserido e depois irá adicionar o número que foi também inserido como input a esse respetivo contacto.

* public void insere(Contacto c):

Este método serve apenas para inserir um novo contacto na agenda através de um método de inserção que se encontra na implementação da estrutura de dados das AVL’s.

* public void remove(String name):

Neste método é utilizado um método da estrutura de dados AVL para remoção de um de certo contacto da agenda utilizando o nome do contacto que queremos remover.

* public void listarContacto():

Este método simplesmente retorna todos os contactos da agenda por ordem alfabética utilizando novamente um método das AVL’s.

Esta classe Agenda irá ser utilizada para a resolução das funcionalidades desta agenda, que são definidas na classe Agendaplementation e mais tarde usadas na classe Main que serve apenas para chamar as funções dependendo do que o utilizador quiser executar. Os métodos existentes nesta classe são:

* public void adicionarContacto():

Este método irá adicionar um contacto novo à agenda dando hipótese de associar um ou mais números para cada nome de cada contacto.

* public void listarContactos():

Este método utiliza o método listarContacto() da classe Agenda para colocar todos os contactos da agenda por ordem.

* public void editarContacto():

Neste método pretende-se aceder a todos os contactos da agenda dando a escolher qual dos contactos o utilizador deseja editar. É dado a escolher ao utilizador se quer editar o nome ou o número ou números de um certo contacto através de if’s e caso o utilizador escolha editar os números de um contacto pode escolher entre adicionar um novo número ao contacto, editar um dos números do contacto ou ainda remover um dos números do contacto sempre através de inputs. Para a alteração e adição de números ou nomes dos contactos são usados os métodos criados na classe Agenda.

* public void procurarTelefone():

Este método pretende procurar um número de telefone na agenda sendo que se encontrar o número dado devolverá o nome do contacto a quem pertence o número e caso não encontre irá dar a escolher ao utilizador mostrar a agenda completa.

* public void exportarContactos():

Este método apenas cria uma queue temporária que serve para transferir todos os contactos da agenda e devolvê-los pela mesma ordem que foram introduzidos na agenda em si, utilizando os métodos implementados na estrutura de dados queue.

* public void removerContacto():

Este método irá remover um contacto da agenda caso ele exista nela. Para isso receberá como input um nome de um contacto e verificará se este existe na agenda. Caso exista utilizará o método de remoção criado na classe Agenda para remover o contacto da agenda por completo.

## Conclusão

Com este trabalho conseguimos perceber como funcionam melhor as estruturas e como utilizá-las, principalmente as AVL’s, e conseguimos também adquirir mais conhecimentos a nível de java.