

1 Objectivos

Pretende-se o uso e implementação de árvores binárias de pesquisa, e AVL's. Devem os alunos apresentar uma implementação destas árvores, e posteriormente, usar a própria implementação para criar uma agenda de telemóvel. Não devem os alunos sem autorização do professor fazer uso de classes do package "java.util", excepto para input/output. A agenda irá armazenar uma associação de nome a número de telefone/telemóvel (associação de 1 para muitos), e deve permitir o manuseamento habitual deste tipo de software.

2 O Trabalho

O trabalho consiste na apresentação de 3 classes Java: uma com a implementação da agenda, outra com uma implementação de ABP's e outra com a implementação de AVL's. Os nomes das classes devem respeitar o definido na tabela abaixo.

Funcionalidade	Interface	Implementação
árvores binárias de pesquisa	-	ABP.java
árvores AVL	-	AVL.java
Agenda	-	Agenda.java

Table 1: Classes a implementar

2.1 Árvores

A implementação das árvores deve seguir o que é habitual para este tipo de estruturas de dados. As implementações que apresentar devem ser "stand-alone". Deve ser possível com as vossas implementações criar ABP's de Integer, String, ou qualquer outro tipo comparável. As suas árvores devem ser iteráveis. O iterador pode iterar as árvores como pretender (em ordem, por níveis, etc), escolha e decida a ordem da iteração (pode até implementar mais do que uma). Para as árvores AVL, pode usar o "lazy deletion". Após a implementação das árvores, nas condições enunciadas, pode avançar para a implementação da agenda de telemóvel.

2.2 A agenda

A agenda de telemóvel, armazenará uma lista de contactos. Um Contacto é identificado pelo nome(ID) e por um ou mais números de telefone(um colega vosso negociou para 1 ou no máximo 2 contactos). Deverá ser possível realizar as seguintes operações básicas na agenda. A sua agenda deverá conter pelo menos uma árvore binária de pesquisa(uma ABP ou/e uma AVL). Não é permitido armazenar os contactos num array, queue, stack ,ou LinkedList. Está também proibido de usar qualquer estrutura de dados importada do java.util. Não crie menus, para tornar a sua agenda interactiva. Basta na classe Agenda criar um main com as funcionalidades pedidas para efeitos de teste. Deve nas implementações da agenda ser eficiente, e analisar a complexidade das suas operações. Implementações mais eficientes serão mais cotadas.

1. ADICIONAR : adiciona à agenda um novo contacto(ID e um número de telefone)
2. EDITAR : permite, usando o nome ou um dos números de telefone, modificar o contacto. É possível modificar o ID do contacto, algum dos números ou até adicionar um novo número ao contacto.
3. REMOVER: permite, usando o nome ou um dos números de telefone, remover o contacto da lista.
4. LISTAR: Lista o conteúdo da agenda, por ordem alfabética. A listagem deve incluir para cada contacto, o nome e os números associados ao contacto.
5. CHAMADOR: Mostra o ID na agenda associado ao número do chamador. Se não existir na agenda deve aparecer "DESCONHECIDO".

2.3 Entrega

O trabalho pode ser realizado individualmente ou em grupos de 2 alunos. Os alunos devem manter os mesmos grupos que formaram para a realização do(s) trabalho(s) anteriores. A data limite para a submissão do trabalho é dia 27 de Novembro de 2018, sendo realizada a submissão pelo moodle, nos moldes habituais. Os trabalhos serão apresentados e avaliados na aula prática de dia 4 de Dezembro. Todos os ficheiros deverão ser "zipados" e submetidos num único ficheiro com o número dos alunos que realizam o trabalho.