

Relatório de desenvolvimento do terceiro trabalho de Algoritmos e Estruturas de Dados I

Para a realização desta atividade foi desenvolvida duas classes: “App” e “Arvore”, na linguagem Java. A classe App é a principal, onde é realizada a leitura dos pergaminhos (.txt) e a impressão do resultado. Já na classe “Arvore” é realizado todos os métodos de manipulação da árvore a partir de Nodos. Nesse relatório será possível encontrar a descrição dos métodos assim como os testes de caso.

- **Classe “App”:**

Com auxílio das bibliotecas: Path, Files e BufferedReader foi realizada a leitura dos pergaminhos.

- Caso conseguir ler o arquivo (try):
 - É instanciada, criada, uma árvore;
 - A partir disso chamamos o método da classe Arvore chamado “arquivoOrganiza” onde, de maneira mais organizada, adiciona o primeiro pai a o primeiro filho, ou seja, as duas primeiras linhas do pergaminho;
 - Enquanto haver linhas, será realizado a separação das linhas inserindo-as em um vetor auxiliar para a adição das mesmas na árvore;
 - Chama método da classe Arvore que atribui terras aos filhos;
 - Por fim, imprime o bárbaro que tiver mais terras.
- Caso não ler o arquivo (catch):
 - Retorna mensagem informando que houve um problema, e a exception.

- **Classe “Arvore”:**

A estrutura de dados “árvore” herda características das topologias em árvore, assim, os dados de uma árvore organizam-se de forma hierárquica. Para o desenvolvimento desta classe utilizei o auxílio das bibliotecas LinkedList e Queue do Java para maior praticidade na manipulação de alguns métodos.

É criada a classe “NodoArvore” para o manuseio dos nodos da árvore. Nesta classe há os atributos pai (NodoArvore), nome (String), filhos (LinkedList<NodoArvore>), terras (int) e nFilhos (int). Seu construtor recebe por parâmetro o número de terras e o nome do bárbaro a ser referenciado. Há os métodos básicos para o desenvolver dos demais métodos da árvore: “addSubtree” onde adiciona uma sub-árvore, ou seja, um filho; “getSubtree” que retorna a sub-árvore; e “getSubtreeSize” onde retorna o tamanho da sub-árvore (número de filhos).

Para métodos da árvore é realizada a criação de duas variáveis: “root” (NodoArvore) que será a raiz da nossa árvore e “nElementos” que contará quantos elementos temos em nossa árvore.

Foi-se realizado a separação dos métodos (com comentários) para melhor organização, sendo elas: métodos básicos, métodos positions, métodos estatísticos e métodos auxiliares.

Métodos Básicos	Descrição
Nivel(String nome)	Recebe por parâmetro o nome do bárbaro que deseja pesquisar, e retorna qual o nível que este bárbaro se encontra.
AddNodo(String nome, Integer terras, String pai)	Recebe por parâmetro o nome do bárbaro a ser adicionado, o número de terras que possui e o nome de seu pai. Retorna true se bárbaro foi inserido, e false caso não for.
PesquisaNodo(String nome, NodoArvore ref)	Recebe por parâmetro nome do bárbaro que deseja pesquisar e o nodo de referência para encontrá-lo (sua raiz). Este é um método que retorna um NodoArvore, sendo assim, caso a referência não existir (for nula), retorna null. Caso a referência informada for igual ao nome informado, retorna a própria

	referência, se não é criado um nodo auxiliar e este é retornado após encontrado.
Métodos Positions	Descrição
PositionsWidth()	Método para percorrer a árvore em sua "largura". Retorna uma lista com todos os bárbaros caso a raiz não for nula.
Métodos Estatísticas	Descrição
ArquivoOrganiza(String l1, String l2)	Recebe por parâmetro duas Strings, estas sendo as duas primeiras linhas do pergaminho. Este método foi desenvolvido para a organização do início do pergaminho, já que a primeira linha onde contém a quantidade de terras do pai é diferente das demais. Método não possui retorno (void).
AtribuiTerras()	Este método realiza a atribuição de terras do pai aos filhos. Caso o root não seja nulo, é realizado o cálculo para a distribuição de terras do pai aos filhos (dividindo o total de terras pela quantidade de filhos).
UltimoNivel()	Este método realiza a criação de uma lista com apenas os bárbaros do último nível. Retorna a lista com os bárbaros do último nível.
BarbaroMaisTerras()	Retorna o bárbaro que tiver mais terras do último nível.
Métodos Auxiliares	Descrição
ImprimeArvore(NodoArvore ref)	Imprime a árvore a partir da raiz como referência.
ImprimeArvore()	Auxilia a impressão da árvore. Verifica se existe (root != null), e chama método para imprimir a partir de root.
ImprimeBarbaroMaisTerras()	Retorna a impressão do bárbaro que tem mais terras e a quantidade que o mesmo tem.

Estes são os métodos utilizados para a realização do exercício. Ao longo do desenvolvimento do trabalho criei outros demais métodos que achei que pudesse me auxiliar, entretanto encontrei o atual modo que o trabalho se encontra como a melhor e mais prática alternativa.

Demais métodos criados: `getRoot()`, `getPai(String nome)`, `contains(String nome)`, `isExternal(String nome)`, `isInternal(String nome)`, `qtdTerras(String nome, String pai)` - retorna a quantidade de terras -, `getMaisFilhos(NodoArvore nodo, int nFilhos)` - Retorno pai (nodo) que possui mais filhos -, `clear()`, `isEmpty()` e `size()`. Esses códigos podem ser encontrados em um arquivo .txt chamado "Métodos_não_usados" envidado juntamente ao arquivo desse relatório. Decidiu-se separá-los para o código final ficar mais limpo.

Todos os códigos realizados foram devidamente acompanhados de comentários no intuito de descrever o melhor possível o desenvolvimento/raciocínio.

- **Testes realizados:**

- *Caso0.txt*

```
103787
Thorgestax Deldriralex 4626
Thorgestax Jocgnibardyx 3530
Jocgnibardyx Diorflimhikox 5154
Jocgnibardyx Grutohiaux 5797
Jocgnibardyx Docmangax 4471
Jocgnibardyx Delscatorflex 6180
Jocgnibardyx Cristipix 5299
Jocgnibardyx Pacmonumicynax 2056
Thorgestax Alteetoflex 6042
Thorgestax Delrenmax 6080
Delrenmax Diormancllox 4736
Delrenmax Neppanpix 2249
Delrenmax Klodrimanrix 5353
Delrenmax Mirtpliblepkox 3177
Thorgestax Frinvinulamax 3189
```

Resultado:

```
Barbaro com mais terras: Klodrimanrix com o total de 12062 terras :)
```

- *Caso1.txt*

```
263412
Floki Siefried 7654
Floki Ragnar 1234
Floki Rollo 2738
Vidar Erick 6019
Vidar Einar 6273
Vidar Gimle 689
Floki Bjorn 7485
Floki Knut 1903
Rollo Ivar 964
Rollo Mimir 873
```

Resultado:

```
Barbaro com mais terras: Ivar com o total de 28674 terras :)
```

○ Caso2.txt

564732
Brofgrenund Thasmilgrir 2341
Brofgrenund Khaertmusnus 4543
Brofgrenund Bhublorthur 6435
Brofgrenund Riglerod 34325
Khaertmusnus Arnvorbith 324
Thasmilgrir Vapeddeith 918
Thasmilgrir Arvirtholf 1244
Brofgrenund Dhalskiploll 2421
Brofgrenund Khorbifdild 2345
Brofgrenund Rerstedbrald 4563
Thasmilgrir Snutgirngrell 6422
Brofgrenund Dhuziprad 5342
Thasmilgrir Frothginnold 5343
Thasmilgrir Rhagvufvaur 7834
Khaertmusnus Preslulfrek 453
Khaertmusnus Sgedduggvaud 5321
Khaertmusnus Jopausug 2342

Resultado:

Barbaro com mais terras: Sgedduggvaud com o total de 24104 terras :)

○ Caso3.txt

1098934
Pulnozaf Lleitvurdget 12234
Pulnozaf Hyaerbasner 342
Pulnozaf Pralbodbraeld 4325
Pralbodbraeld Lleitvurdget 5453
Pralbodbraeld Dayvesolk 5432
Lleitvurdget Yultenilk 5667
Pulnozaf Hrulskulsnerd 3453
Pulnozaf Kausbrothgokr 5432
Pralbodbraeld Morthullgonn 4534
Pralbodbraeld Skatmudrol 6544

Resultado:

Barbaro com mais terras: Skatmudrol com o total de 62571 terras :)

Caso4.txt

563782
Lirgrolrund Yekkurkmaern 4534
Lirgrolrund Larngethredr 765
Lirgrolrund Satgantom 8743
Yekkurkmaern Nalugfug 3456
Yekkurkmaern Bheigforlam 5434
Yekkurkmaern Kreslafnonn 2345
Yekkurkmaern Dettaugnof 345
Yekkurkmaern Hraellkolnuk 1093
Lirgrolrund Firtmeiddot 3245
Nalugfug Vumsegnug 5432
Nalugfug Kuzgaetilr 1983

Resultado:

Barbaro com mais terras: Vumsegnug com o total de 21707 terras :)