

Mapas e tabelas de dispersão

Lista06

Informações

- Trabalho em *duplas*;
- Entrega pelo Moodle em `<http://trab.dc.unifil.br/moodle/>`.

Para fazer esta lista, parta do projeto `TabelasDispersao`, incluso no pacote deste roteiro.

1. A implementação inicial da tabela de dispersão do pacote desta atividade é funcional, porém muito simples. Seu principal problema é não verificar inserções que causem colisões de dispersão. Portanto, implemente as seguintes políticas para tratamento de colisões:
 - (a) Encadeamento separado: utilização de listas encadeadas em cada posição da tabela.
 - (b) Expansão: sempre que a tabela atingir 100% de ocupação em relação a `tabela.length` **ou** alguma lista encadeada ficar maior que 3 elementos, a tabela dobra de tamanho. Lembre-se de reinserir os elementos na nova tabela utilizando a nova função de dispersão.
2. O tratamento de colisões da atividade anterior é totalmente funcional e é o mais simples de ser implementado, porém não é o que apresenta melhor desempenho, em especial quanto a acessos e ocupação de memória. Nesta atividade, pesquise e estude em [1, 2] sobre a técnica de **endereçamento aberto** e implemente-a.
3. Acesse a documentação de `Object` do Java 8 [3] e leia as especificações dos métodos `equals` e `hashCode`. Escreva um resumo sobre ambos, constando a relação entre eles e as propriedades de igualdade de objetos.
4. Leia a documentação e tutorial de uso da TAD *Mapa* [4] e utilize a classe `Map` para programar a classe `HistogramaCaracteres`, que recebe uma `String` como parâmetro de construção e conta a quantidade de cada caractere presente na `String`, ignorando os espaços em branco. A classe escreve essa relação computada através do método `toString`. Considere o seguinte código ilustrando a utilização `HistogramaCaracteres`:

```
1 String msg = "Vê se tira notas boas!";
2 HistogramaCaracteres hc = new HistogramaCaracteres(msg);
3 System.out.println("Resultado do histograma:\n" + hc.toString());
```

O código acima gera a seguinte saída no terminal:

Terminal de saída

```
Resultado do histograma:
a: 3
b: 1
e: 1
```

ê: 1
i: 1
n: 1
o: 2
s: 3
t: 2
V: 1
!: 1

5. Apresente a análise de complexidade assintótica para todos os algoritmos implementados nesse exercício.

Referências

- [1] CORMEN, Thomas H. et al. *Algoritmos: teoria e prática*. Rio de Janeiro: Campus, 2012. 926 p. ISBN 978-85-352-3699-6.
- [2] GOODRICH, Michael T.; TAMASSIA, Roberto. *Estruturas de dados e algoritmos em Java*. 4. ed. Porto Alegre: Bookman, 2007. 600 p. ISBN 9788560031504.
- [3] ORACLE. *Object (Java Platform SE 8)*. Disponível em: <<https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html>>.
- [4] ORACLE. *The Map Interface (The Java Tutorials > Collections > Interfaces)*. Disponível em: <<https://docs.oracle.com/javase/tutorial/collections/interfaces/map.html>>.