Google olabi

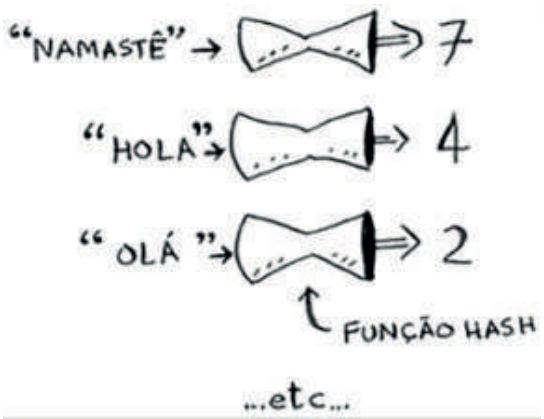
<ESTRUTURADEDADOS>
<HASHTABLES>

{ Estrutura de Dados

Quase toda linguagem de programação contém estruturas de dados baseadas em hash. A linguagem java contém estruturas de dados de tabela hash (Hashtable), mapa de hash (HashMap) e árvores baseadas na função hash. A base destas estruturas de dados é o design do valor-chave: nele, cada chave é única, mas o mesmo valor pode existir para várias chaves correspondentes a um objeto, uma vez que esta função sempre retorna o valor inteiro para o mesmo objeto. Destrinchamos todos estes conceitos e terminologias a seguir.

{ O que é Hash?

Hashing é o processo de transformar qualquer chave ou string de caracteres em outro valor/tipo, usualmente representado por um valor ou chave de comprimento fixo mais curto de forma a facilitar a localização ou uso da string original. Uma função hash gera novos valores baseados em um algoritmo matemático, também conhecido como “valor de hash”. Para evitar que a conversão do Hash volte à chave original, o valor sempre utiliza algoritmo unidirecional.



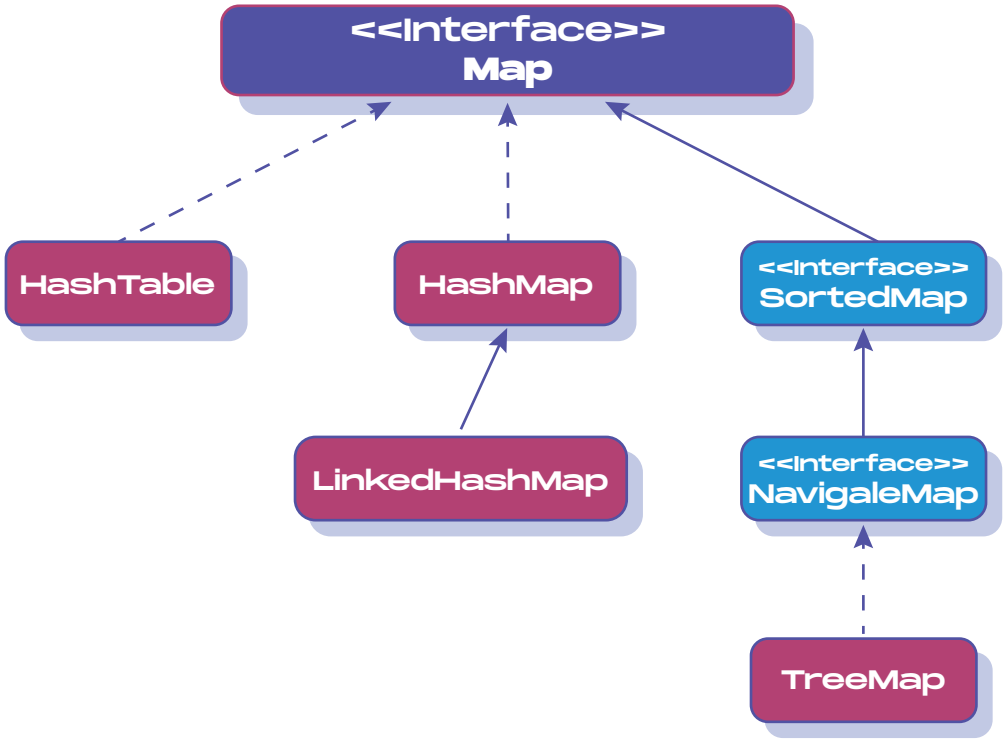
O hashing também permite o acesso a dados de maneira eficiente no que diz respeito à computação e espaço de armazenamento, já que reduz o tempo de acesso não linear de listas ordenadas, não ordenadas e de árvores estruturadas, bem como requisitos de armazenamento - frequentemente exponenciais, de acesso direto à espaços de estado de chaves grandes, ou de comprimento variável.

{ Map e HashMap

Map é a interface usada para armazenar dados em um par chave-valor, enquanto HashMap é a classe de implementação da interface Map. Java tem várias classes (TreeHashMap, LinkedHashMap) que implementam a interface Map para armazenar dados em um par de valor-chave.

A interface Map sozinha não pode ser usada para armazenar dados, mas podemos criar um objeto de suas classes de implementação e então usar a referência Map para armazenar o objeto.

<Hierarquia da interface Map>



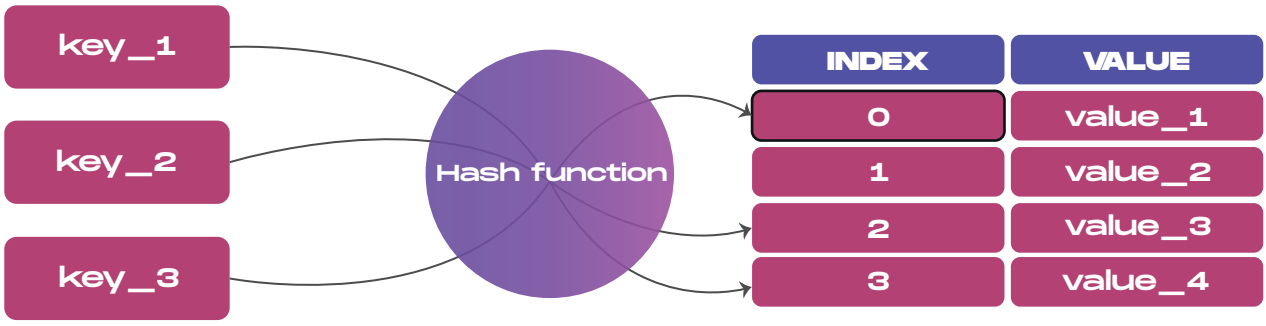
O uso mais popular para hash é a implementação de tabelas de hash (Hash Table). As funções de hash e suas tabelas de hash associadas são usadas em aplicativos de armazenamento e recuperação de dados para acessar dados em um tempo pequeno e quase constante. Estas exigem uma quantidade de espaço de armazenamento apenas uma fração maior do que o espaço total necessário para os próprios dados ou registros.

{ Hashtable

Uma tabela de hash armazena pares de chave e valor em uma lista que pode ser acessada por meio de seu índice. Como os pares de chave e valor são ilimitados, a função hash mapeará as chaves para o tamanho da tabela. Um valor de hash torna-se o índice de um elemento específico.

A classe Hashtable implementa uma tabela de hash, que mapeia chaves para valores. Ela herda a classe Dictionary e implementa a interface Map. Para um melhor entendimento, digamos que temos um dicionário, onde cada palavra tem sua definição. Além disso, precisamos obter, inserir e remover palavras do dicionário rapidamente. As palavras serão as chaves no Hashtable, pois elas são supostamente únicas, e as definições serão os valores.

<Exemplo de Tabela de Hash>



Para armazenar e recuperar com sucesso objetos de uma hashtable, os objetos usados como chaves devem implementar o método **hashCode()** para determinar qual container o par chave-valor deve mapear. Geralmente, o hashcode é um inteiro não-negativo que é igual para objetos iguais e pode ou não ser igual para objetos desiguais. Para determinar se dois objetos são iguais ou não, o hashcode faz uso do método **equal().**

Colisão de hash

É possível que dois objetos desiguais tenham o mesmo hashcode. Isto é chamado de colisão. Para resolver colisões, o hashtable utiliza um array de listas. Os pares mapeados para um único container (array index) são armazenados em uma lista e a referência da lista é armazenada no array index.

Diferença entre HashMap e Hashtable

HashMap e Hashtable são ambos usados para armazenar dados em forma de chave e valor. Ambos estão usando a técnica de hashing para armazenar chaves exclusivas. Entretanto, há muitas diferenças entre as classes HashMap e Hashtable que são dadas a seguir:

HashMap	Hashtable
HashMap não é sincronizado. Não é seguro e não pode ser compartilhado entre muitos threads sem o código de sincronização adequado.	O Hashtable é sincronizado. É segura para os threads e pode ser compartilhada com muitas threads.
Permite uma chave nula e múltiplos valores nulos.	Não permite nenhuma chave ou valor nulo.
O HashMap é uma nova classe introduzida em JDK 1.2.	O Hashtable é uma classe clássica.
É rápido	É lento
HashMap é percorrido pelo Iterator.	Hashtable é percorrido pelo Enumerator e Iterator.
O Iterator no HashMap é rápido em caso de falha.	Enumerator em Hashtable não é rápido em caso de falha.
HashMap herda a classe AbstractMap.	Hashtable herda a classe do Dicionário.

{ Métodos de Hashtable

Alguns métodos que podem ser utilizados para Hashtable estão apresentados a seguir.

Método	Descrição
clear()	Limpa a Hashtable.
clone()	Cria uma cópia.
compute()	Calcula um mapeamento para a chave especificada e seu valor mapeado atual.
containsKey()	Testa se o objeto especificado é uma chave.
containsValue()	Retorna verdadeiro se este hashtable mapeia uma ou mais chaves para este valor.
elements()	Retorna uma enumeração dos valores
get()	Retorna o valor para o qual a chave especificada é mapeada.
isEmpty()	Testa se este hashtable não contém nenhuma chave para os valores.
keys()	Retorna uma enumeração das chaves.
merge()	Se a chave especificada ainda não está associada a um valor ou está associada a um valor nulo, associa-a ao valor não-nulo dado.
remove()	Remove a chave (e seu valor correspondente).
size()	Retorna o número de chaves.

Referências:

https://www.geeksforgeeks.org/hashtable-in-java/
https://docs.oracle.com/javase/8/docs/api/java/util/Hashtable.html
https://stringfixer.com/pt/Hash_sum
https://www.javatpoint.com/difference-between-hashmap-and-hashtable

Hash:

https://pt.education-wiki.com/7864460-hashing-function-in-java
https://www.techtarget.com/searchdatamanagement/definition/hashing

Map e Hashmap:

https://www.delftstack.com/pt/howto/java/difference-between-hashmap-and-map-in-java/

Professor: Júlio César | Monitora: Thássia Lima

Google olabi

<ESTRUTURADEDADOS>
<HASHTABLES>

/exercícios/

1. Escrever um programa Java para testar se um mapa contém um mapeamento para a chave especificada.

2. Escrever um programa Java para remover todos os mapeamentos de um mapa.

3. (Hashtable) Uma empresa precisa de um programa de computador que efetue o cadastro de compradores:

* Os compradores deverão ser alocados e recuperados rapidamente da memória.

* Crie o programa para esta empresa, alocando os “Compradores” em uma hash table

* Utilizando a Api do java.

* Use sua criatividade para escolher os componentes que serão utilizados para

* Construir a Hash Table

* A chave hash deverá ser composta pelo NOME;

* Cada comprador tem os seguintes dados:Nome; RG; CPF; Telefone