Estructuras de datos

Arle Morales Ortiz
Docente
Ingeniería de software

Tablas hash

Una tabla hash, matriz asociativa, mapa hash, tabla de dispersión o tabla fragmentada es una estructura de datos que asocia llaves o claves con valores. La operación principal que soporta de manera eficiente es la búsqueda: permite el acceso a los elementos (teléfono y dirección, por ejemplo) almacenados a partir de una clave generada (usando el nombre o número de cuenta, por ejemplo). Funciona transformando la clave con una función hash en un hash, un número que identifica la posición (casilla o espacio) donde la tabla hash localiza el valor deseado.

Ejemplo básico :

Necesitamos obtener, insertar y
eliminar palabras del diccionario rápidamente.

Por lo tanto, Hashtable (o HashMap) tiene sentido.
Las palabras serán las claves en Hashtable ,
ya que se supone que son únicas. Las definiciones,
en cambio, serán los valores.

hashCo*utilizado como clave en una Hashtable , el objeto no debe violar el contrato hashCo****de()*** *para ser de() . los objetos iguales deben devolver el mismo código.*

- La posición del depósito se identifica llamando al método hashcode(). Una tabla hash contiene valores basados en la clave.
- La clase Java Hashtable contiene elementos únicos.
- La clase Java Hashtable no permite una clave o valor nulo.
- La clase Java Hashtable está sincronizada.

Ejemplo 1 código:

```java
package co.edu.cue.clases;

public class Word {
    private String name;

    public Word(String name) {
        this.name = name;
    }

}
```

Word.java | PrincipalWord.java

```java
package co.edu.cue.principal;

import co.edu.cue.clases.Word;

import java.util.Hashtable;

public class PrincipalWord {
    public static void main(String[] args) {
        Hashtable<Word, String> table = new Hashtable<>();

        Word word = new Word( name: "cat");
        table.put(word, "an animal");

        System.out.println(table.get(word));

        System.out.println("remove an entry :" + table.remove(word));

    }
}
```

Ahora podemos crear un Hashtable

agreguemos una entrada

obtener una entrada

Ejemplo 2 código:

```java
package co.edu.cue.clases;

import java.util.Hashtable;
import java.util.Map;

public class Hashtable1 {
    public static void main(String args[]) {
        Hashtable<Integer, String> hm = new Hashtable<Integer, String>();

        hm.put(100, "Ana");
        hm.put(102, "Luis");
        hm.put(101, "Santiago");
        hm.put(103, "Mafe");

        for (Map.Entry m : hm.entrySet()) {
            System.out.println(m.getKey() + " " + m.getValue());
        }
    }
}
```

Si solo necesitamos las claves en un mapa, keySet es una buena opción. Sin embargo, existe una forma más rápida de obtener tanto las claves como los valores.

El método entrySet() retorna un objeto de la interfaz interna Entry, que representa una pareja clave-valor.

Ejemplo 2B código

```java
package co.edu.cue.clases;

import java.util.Hashtable;
import java.util.Map;

public class Hashtable1 {
    public static void main(String args[]) {
        Hashtable<Integer, String> hm = new Hashtable<Integer, String>();

        hm.put(100, "Ana");
        hm.put(102, "Luis");
        hm.put(101, "Santiago");
        hm.put(103, "Mafe");

        System.out.println(hm.getOrDefault( key: 101,  defaultValue: "Not Found"));
        System.out.println(hm.getOrDefault( key: 105,  defaultValue: "Not Found"));
    }
}
```

El método getOrDefault (Object key, V defaultValue) de la interfaz Map , se utiliza para obtener el valor asignado con la clave especificada. Si no se asigna ningún valor con la clave proporcionada, se devuelve el valor predeterminado.

Ejemplo 2C código

```java
package co.edu.cue.clases;

import java.util.Hashtable;
import java.util.Map;

public class Hashtable1 {
    public static void main(String args[]) {
        Hashtable<Integer, String> hm = new Hashtable<Integer, String>();

        hm.put(100, "Ana");
        hm.put(102, "Luis");
        hm.put(101, "Santiago");
        hm.put(103, "Mafe");

        System.out.println("Initial Map: "+ hm);
        //Inserts, as the specified pair is unique
        hm.putIfAbsent(104,"Arle");
        System.out.println("Updated Map: "+ hm);
        //Returns the current value, as the specified pair already exist
        hm.putIfAbsent(101,"pepe");
        System.out.println("Updated Map: "+ hm);

    }
}
```

El método putIfAbsent (clave K, valor V) de la clase HashMap se usa para mapear la clave especificada con el valor especificado, solo si dicha clave no existe (o está mapeada como nula) en esta instancia de HashMap.

Ejemplo 3 código

```java
package co.edu.cue.principal;

import co.edu.cue.clases.Book;
import java.util.Hashtable;
import java.util.Map;

public class Principalbook {
    public static void main(String[] args) {
        //Creating map of Books
        Map<Integer, Book> map = new Hashtable<Integer,Book>();
        //Creating Books
        Book b1=new Book( id: 101, name: "Let us C", author: "Yashwant Kanetkar", publisher: "BPB", quantity: 8);
        Book b2=new Book( id: 102, name: "Data Communications & Networking", author: "Forouzan", publisher: "Mc Graw Hill", quantity: 4);
        Book b3=new Book( id: 103, name: "Operating System", author: "Galvin", publisher: "Wiley", quantity: 6);
        //Adding Books to map
        map.put(1,b1);
        map.put(2,b2);
        map.put(3,b3);
        //Traversing map
        for(Map.Entry<Integer, Book> entry:map.entrySet()){
            int key=entry.getKey();
            Book b=entry.getValue();
            System.out.println(key+" Details:");
            System.out.println(b.id+" "+b.name+" "+b.author+" "+b.publisher+" "+b.quantity);
        }
    }
}
```

Ejercicios 2

1. visite el siguiente enlace, dado el ejemplo implemente una aplicación parecida apóyese en el código.

https://www.scientecheasy.com/2020/12/hashtable-in-java.html/

2. Investigue 5 métodos de Hashtab y desarrolle en java un ejercicio básico aplicando 2 de ellos.

3. Seguir las instrucciones dadas por su profesor e implementar el archivo “código.java”.

¡Gracias!

Arle Morales Ortiz
2022

Corporación Universitaria Empresarial Alexander von Humboldt

Filial de: CÁMARA DE COMERCIO DE ARMENIA Y DEL QUINDÍO