Name: IBRAHIM ALSHAYEA, ID: 202176470, Date; 4/28/2024

Classes: Entry<T> and HashTable<T>

1- Entry<T> Class:

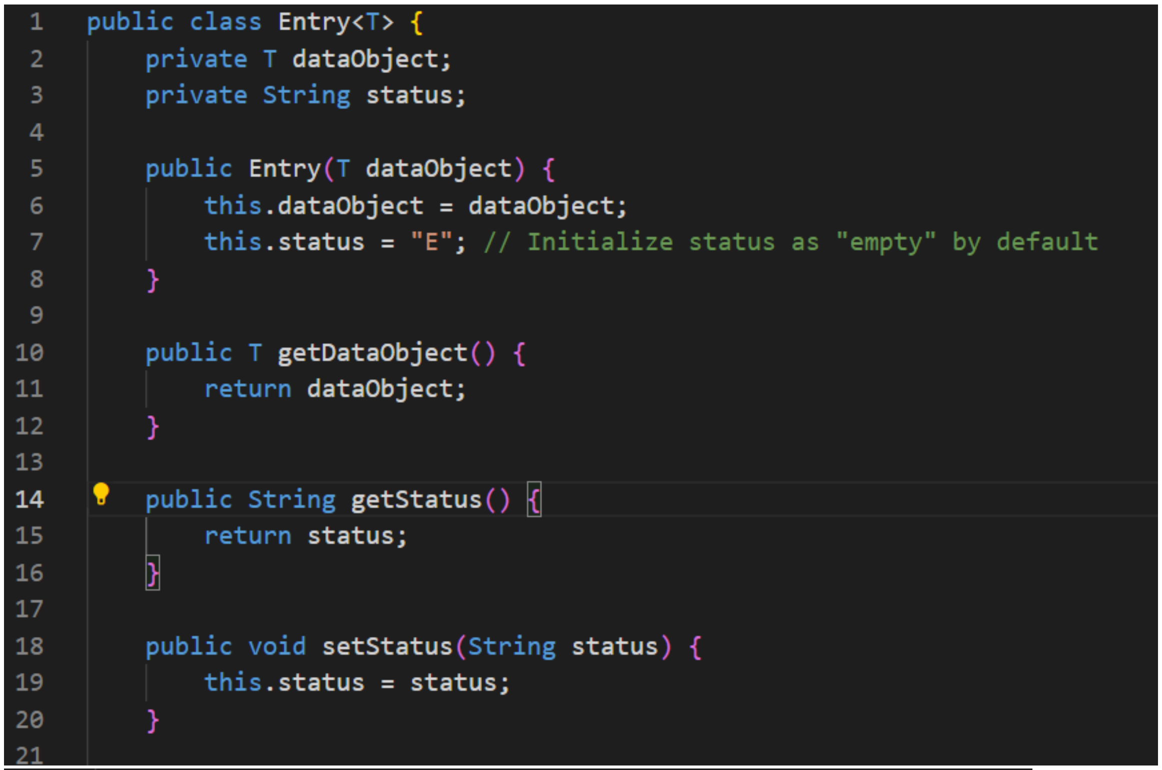
Represents an entry in a hashtable.

Contains two fields: dataObject of type T to hold the data, and status of type String to represent the status of the entry.

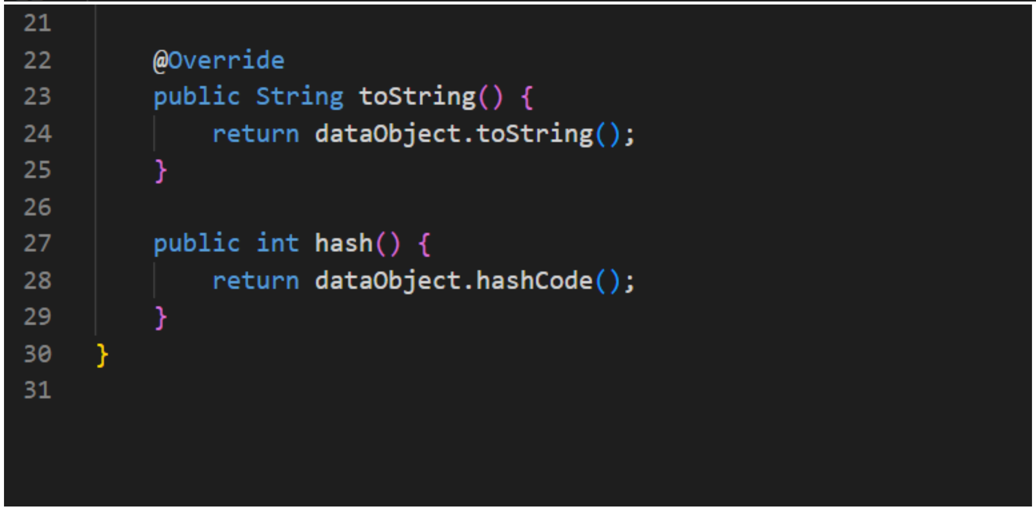
Provides methods to get and set the dataObject and status.

Includes a toString() method to return a string representation of the dataObject.

Entry class Code:



```java
public class Entry<T> {
    private T dataObject;
    private String status;

    public Entry(T dataObject) {
        this.dataObject = dataObject;
        this.status = "E"; // Initialize status as "empty" by default
    }

    public T getDataObject() {
        return dataObject;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

```



```java

    @Override
    public String toString() {
        return dataObject.toString();
    }

    public int hash() {
        return dataObject.hashCode();
    }
}

```

2- HashTable<T> Class:

Represents a hashtable data structure.

Uses an array of Entry<T> objects.

Provides methods for insertion, deletion, and finding elements in the hashtable.

Uses linear probing for collision resolution.

Includes a toString() method to return a string representation of the contents of the hashtable.

Main Method:

Tests the functionality of the HashTable class.

Creates a HashTable instance with a specified size.

Inserts elements into the hashtable and prints it after each insertion.

Performs find operations to search for elements in the hashtable.

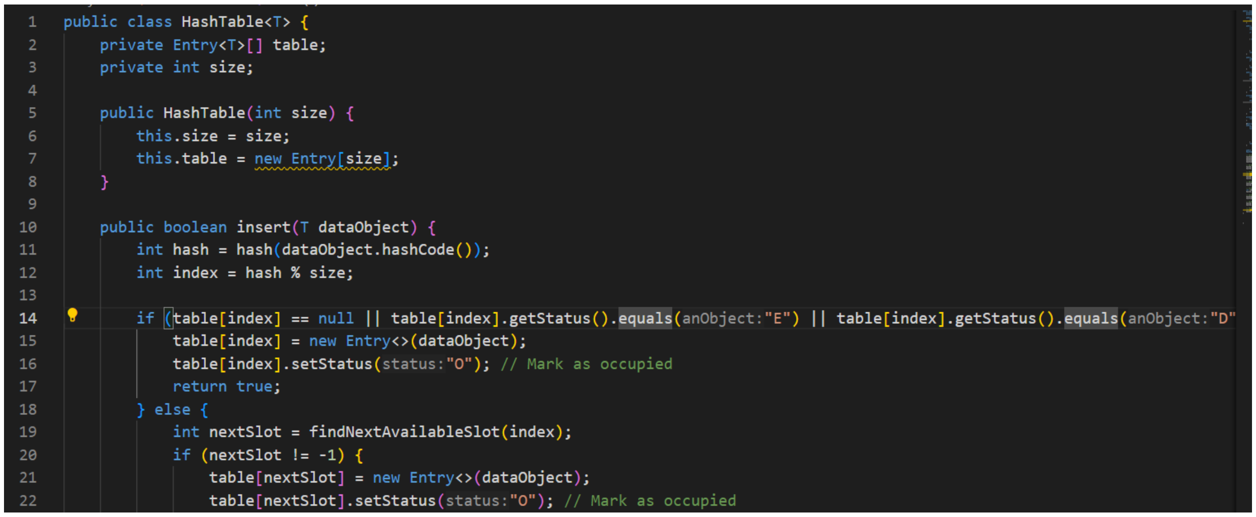
Deletes an element from the hashtable and prints it after deletion.

Inserts additional elements and prints the hashtable again.

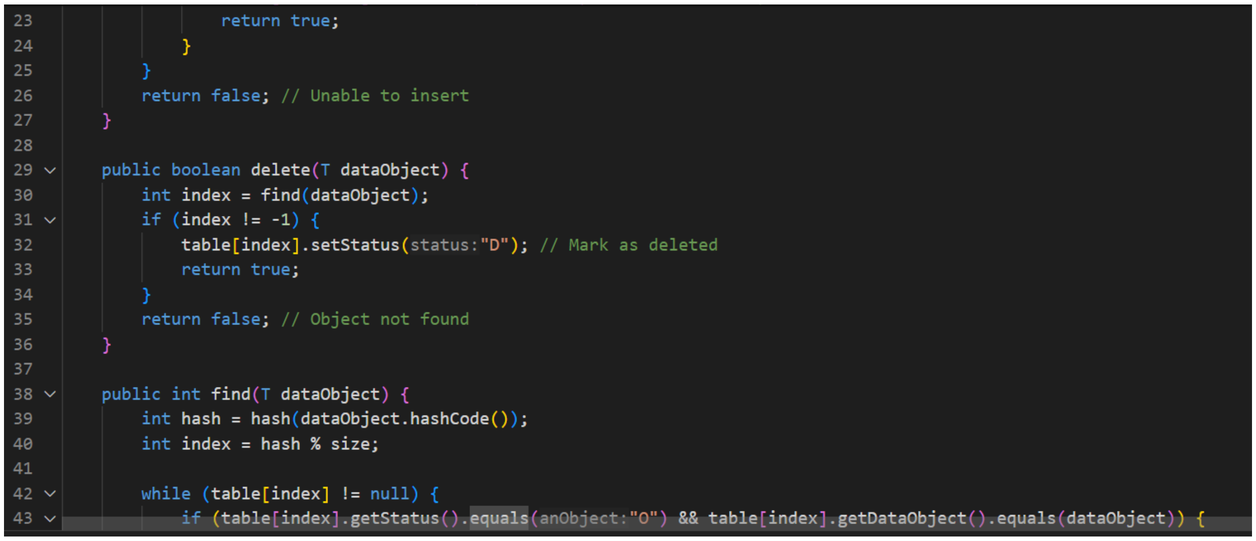
Performs a find operation for an element that was previously deleted.

Prints the final state of the hashtable after all operations.

HashTable Code is too long so I will provide the half of it, the rest is in the attached java file:

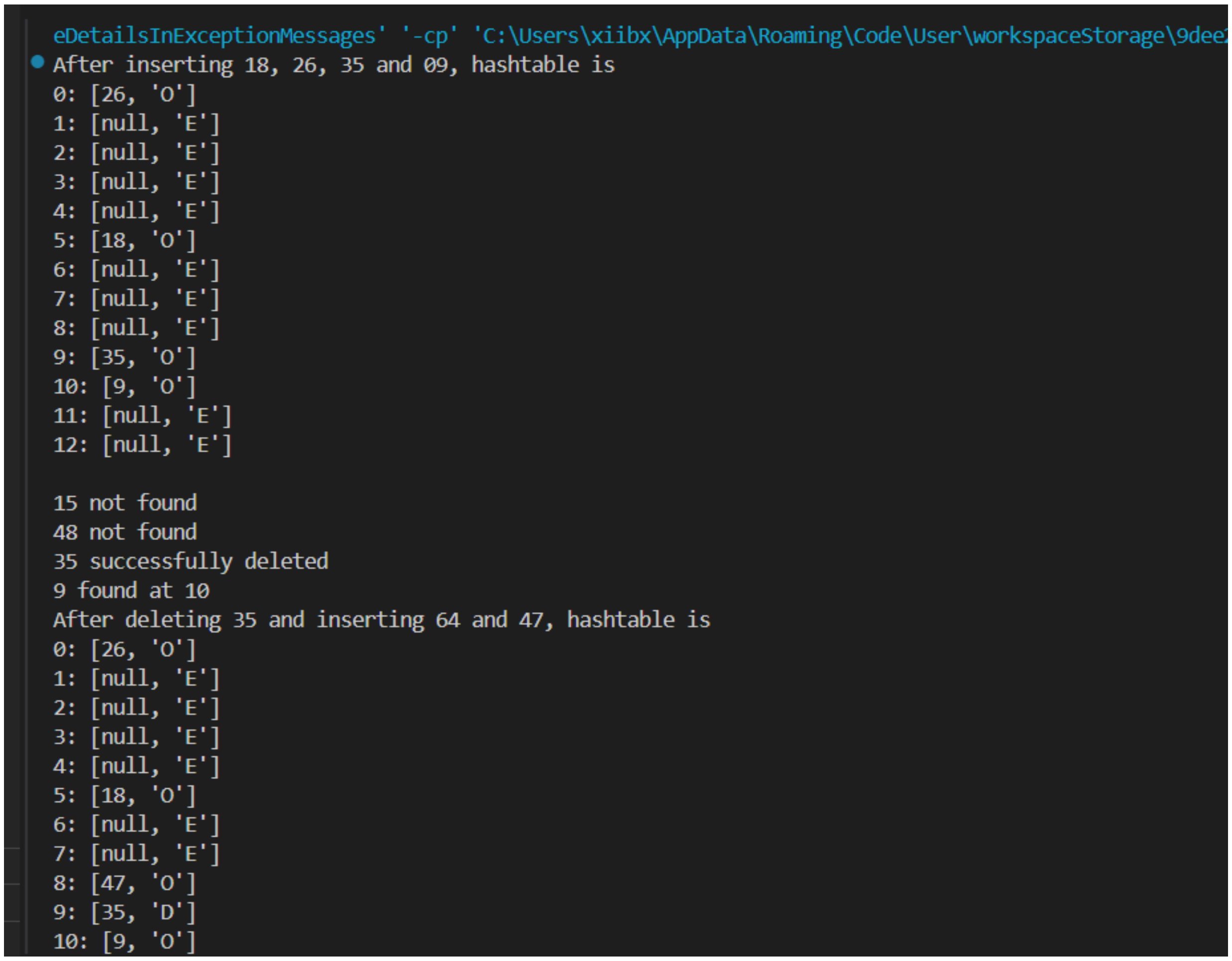


```java
public class HashTable<T> {
    private Entry<T>[] table;
    private int size;

    public HashTable(int size) {
        this.size = size;
        this.table = new Entry[size];
    }

    public boolean insert(T dataObject) {
        int hash = hash(dataObject.hashCode());
        int index = hash % size;

        if (table[index] == null || table[index].getStatus().equals(anObject:"E") || table[index].getStatus().equals(anObject:"D"
            table[index] = new Entry<>(dataObject);
            table[index].setStatus(status:"O"); // Mark as occupied
            return true;
        } else {
            int nextSlot = findNextAvailableSlot(index);
            if (nextSlot != -1) {
                table[nextSlot] = new Entry<>(dataObject);
                table[nextSlot].setStatus(status:"O"); // Mark as occupied
```



```java
                return true;
            }
        }
        return false; // Unable to insert
    }

    public boolean delete(T dataObject) {
        int index = find(dataObject);
        if (index != -1) {
            table[index].setStatus(status:"D"); // Mark as deleted
            return true;
        }
        return false; // Object not found
    }

    public int find(T dataObject) {
        int hash = hash(dataObject.hashCode());
        int index = hash % size;

        while (table[index] != null) {
            if (table[index].getStatus().equals(anObject:"O") && table[index].getDataObject().equals(dataObject)) {
```

Output:



```
eDetailsInExceptionMessages' '-cp' 'C:\Users\xiibx\AppData\Roaming\Code\User\workspaceStorage\9dee
After inserting 18, 26, 35 and 09, hashtable is
0: [26, 'O']
1: [null, 'E']
2: [null, 'E']
3: [null, 'E']
4: [null, 'E']
5: [18, 'O']
6: [null, 'E']
7: [null, 'E']
8: [null, 'E']
9: [35, 'O']
10: [9, 'O']
11: [null, 'E']
12: [null, 'E']

15 not found
48 not found
35 successfully deleted
9 found at 10
After deleting 35 and inserting 64 and 47, hashtable is
0: [26, 'O']
1: [null, 'E']
2: [null, 'E']
3: [null, 'E']
4: [null, 'E']
5: [18, 'O']
6: [null, 'E']
7: [null, 'E']
8: [47, 'O']
9: [35, 'D']
10: [9, 'O']
```