**DSA Assignment 3 Task 2**

In this assignment Task 2, Person class has been created. Person’s name is to generate HashCode, which added all ASCII code of name characters together. Attribute key represent Key. Attributes: name, phonenumber and age represent value.

I create a SingleLinkedList package to create Single Link List to store E type object (in this assignment we only store person object). The NodeLinkedList is the entry in the List and Hash Table (which I would create later). I create HashMap package to create HashTableWithChaining for E type object as required. It has add, remove, contain, getSize, toString as required.

The hashTable has an array of SingleLinkedList <Person>. It has InitialCapacity of 10. It also has a load factor of 0.75, if number of entries exceed the 75% of capacity, it will double the size of hash table (factor of 2) and re hash all elements of the hash table.

My Hash Table resolve the collision. If 2 people are in one bucket however they have 2 different keys. They will be linked by a SingleLinkedList. If a new added element has the same key with previous element, the previous element value will be changed to value of new element.

The main class is Assignment3\_Task2 Class

**Sample output**

**- Check Add Method (include tostring to see result) The Added method some collision test (my and jimmy has different key but link in same bucket)**

1. Add a person to HashTable

2. Remove a person to HashTable

3. Check a person (test contains method)

4. Get size of HashTable

5. To String method

6. Quit program

1

Please provide person nane:

my

Please provide phone number:

021

Please person's age:

21

Person Person{name=my, key=230} has been added. The current size of the map is 1 with the capacity of 10

1. Add a person to HashTable

2. Remove a person to HashTable

3. Check a person (test contains method)

4. Get size of HashTable

5. To String method

6. Quit program

1

Please provide person nane:

jimmy

Please provide phone number:

022

Please person's age:

22

Person Person{name=jimmy, key=550} has been added. The current size of the map is 2 with the capacity of 10

1. Add a person to HashTable

2. Remove a person to HashTable

3. Check a person (test contains method)

4. Get size of HashTable

5. To String method

6. Quit program

1

Please provide person nane:

duc

Please provide phone number:

023

Please person's age:

23

Person Person{name=duc, key=316} has been added. The current size of the map is 3 with the capacity of 10

1. Add a person to HashTable

2. Remove a person to HashTable

3. Check a person (test contains method)

4. Get size of HashTable

5. To String method

6. Quit program

1

Please provide person nane:

thanh

Please provide phone number:

024

Please person's age:

24

Person Person{name=thanh, key=531} has been added. The current size of the map is 4 with the capacity of 10

1. Add a person to HashTable

2. Remove a person to HashTable

3. Check a person (test contains method)

4. Get size of HashTable

5. To String method

6. Quit program

1

Please provide person nane:

tien

Please provide phone number:

025

Please person's age:

25

Person Person{name=tien, key=432} has been added. The current size of the map is 5 with the capacity of 10

1. Add a person to HashTable

2. Remove a person to HashTable

3. Check a person (test contains method)

4. Get size of HashTable

5. To String method

6. Quit program

5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Print out the whole hash table:

Bucket [0]: Person{name=my, key=230} Person{name=jimmy, key=550}

Bucket [1]: Person{name=thanh, key=531}

Bucket [2]: Person{name=tien, key=432}

Bucket [3]: Null

Bucket [4]: Null

Bucket [5]: Null

Bucket [6]: Person{name=duc, key=316}

Bucket [7]: Null

Bucket [8]: Null

Bucket [9]: Null

**- Check Remove Method**

1. Add a person to HashTable

2. Remove a person to HashTable

3. Check a person (test contains method)

4. Get size of HashTable

5. To String method

6. Quit program

2

Please provide person nane:

duc

Person Person{name=duc, key=316} has been removed. The current size of the map is 5 with the capacity of 10

Person Person{name=duc, key=316} has been removed. The current size of the map is 4 with the capacity of 10

1. Add a person to HashTable

2. Remove a person to HashTable

3. Check a person (test contains method)

4. Get size of HashTable

5. To String method

6. Quit program

5

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Print out the whole hash table:

Bucket [0]: Person{name=my, key=230} Person{name=jimmy, key=550}

Bucket [1]: Person{name=thanh, key=531}

Bucket [2]: Person{name=tien, key=432}

Bucket [3]: Null

Bucket [4]: Null

Bucket [5]: Null

Bucket [6]: Null

Bucket [7]: Null

Bucket [8]: Null

Bucket [9]: Null

-Check Person