**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

Below are sample out put to check all features :add, remove, check (contains), get size, to String , Hash Table extend by a factor of 2 when the the number of element / capacity exceed 0.75

**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**

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

3

Please provide person nane:

jimmy

This person: jimmy in the hash table

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

3

Please provide person nane:

hdashduihd

This person: hdashduihd NOT in the hash table

* **Check get size**

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

4

Size of the hash table:4

* **Check toString 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

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 extend Hash table (keep adding until it exceed 0.75). In this case, the initial capacity is 10. So after we add 4 people , the total number is 8, which make the ratio exceed 0.75. The table size will double from capacity of 10 to 20 and all table will be rehashed**

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:

addPerson1

Please provide phone number:

031

Please person's age:

31

Person Person{name=addPerson1, key=977} 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

1

Please provide person nane:

addPerson2

Please provide phone number:

032

Please person's age:

32

Person Person{name=addPerson2, key=978} has been added. The current size of the map is 6 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:

addPerson3

Please provide phone number:

033

Please person's age:

33

Person Person{name=addPerson3, key=979} has been added. The current size of the map is 7 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]: Person{name=addPerson1, key=977}

Bucket [8]: Person{name=addPerson2, key=978}

Bucket [9]: Person{name=addPerson3, key=979}

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

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:

addPerson4

Please provide phone number:

034

Please person's age:

34

Person Person{name=addPerson4, key=980} has been added. The current size of the map is 8 with the capacity of 10

===========================================================================================================================================================

The load factor has exceed 0.75. Hash Table are exapanded by a factor of 2 . The old capacity is: 10 The new capacity is: 20

All elements will be rehashed as follow:

Person Person{name=my, key=230} has been added. The current size of the map is 1 with the capacity of 20

Person Person{name=jimmy, key=550} has been added. The current size of the map is 2 with the capacity of 20

Person Person{name=addPerson4, key=980} has been added. The current size of the map is 3 with the capacity of 20

Person Person{name=thanh, key=531} has been added. The current size of the map is 4 with the capacity of 20

Person Person{name=tien, key=432} has been added. The current size of the map is 5 with the capacity of 20

Person Person{name=addPerson1, key=977} has been added. The current size of the map is 6 with the capacity of 20

Person Person{name=addPerson2, key=978} has been added. The current size of the map is 7 with the capacity of 20

Person Person{name=addPerson3, key=979} has been added. The current size of the map is 8 with the capacity of 20

===========================================================================================================================================================

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=addPerson4, key=980}

Bucket [1]: Null

Bucket [2]: Null

Bucket [3]: Null

Bucket [4]: Null

Bucket [5]: Null

Bucket [6]: Null

Bucket [7]: Null

Bucket [8]: Null

Bucket [9]: Null

Bucket [10]: Person{name=my, key=230} Person{name=jimmy, key=550}

Bucket [11]: Person{name=thanh, key=531}

Bucket [12]: Person{name=tien, key=432}

Bucket [13]: Null

Bucket [14]: Null

Bucket [15]: Null

Bucket [16]: Null

Bucket [17]: Person{name=addPerson1, key=977}

Bucket [18]: Person{name=addPerson2, key=978}

Bucket [19]: Person{name=addPerson3, key=979}

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