

Hello, Peter

Thank you for your detailed information about the problem. With your sample application you help us to find and fix your problems.

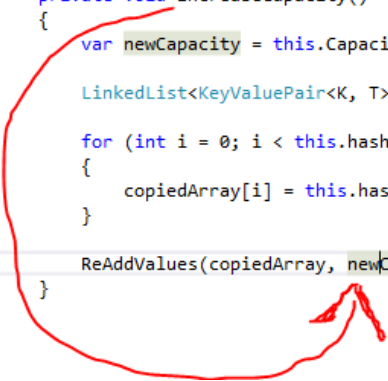
- There are some compilation error in Remove method, that doesn't let you to compare objects with operators "==" and "!=". You can use Equals method to compare objects. We will fix it.
- **Poor:** The main problem is in **IncreaseCapacity** method in **HashTable** class. **When you pass parameters to ReAddValues method the second parameter should be new capacity, not old.** So you don't increase the capacity of hash table.

```
private void IncreaseCapacity()
{
    var newCapacity = this.Capacity * 2;

    LinkedList<KeyValuePair<K, T>>[] copiedArray = new LinkedList<KeyValuePair<K, T>>[newCapacity];

    for (int i = 0; i < this.hashTable.Length; i++)
    {
        copiedArray[i] = this.hashTable[i];
    }

    ReAddValues(copiedArray, newCapacity);
}
```



- **Better:** I will give you one advanced advice. If you want increase performance of hash table, you do not need to increasing capacity of hash table. You can just remove **IncreaseCapacity** and **ReAddValues** methods. I am commented them.

```
public void Add(K key, T value)
{
    int hashedKey = this.GenerateIndex(key);

    //if (this.Count > ((this.Capacity / 100.0) * 75))
    //{
    //    this.IncreaseCapacity();
    //}

    if (this.hashTable[hashedKey] == null)
    {
        this.hashTable[hashedKey] = new LinkedList<KeyValuePair<K, T>>();
        this.hashTable[hashedKey].AddFirst(new KeyValuePair<K, T>(key, value));
    }
    else
    {
        this.hashTable[hashedKey].AddAfter(this.hashTable[hashedKey].Last, new KeyValuePair<K, T>(key, value));
    }

    this.Count++;
}

//private void IncreaseCapacity()
//{
//    var newCapacity = this.Capacity * 2;
//    LinkedList<KeyValuePair<K, T>>[] copiedArray = new LinkedList<KeyValuePair<K, T>>[newCapacity];
//    for (int i = 0; i < this.hashTable.Length; i++)
//    {
//        copiedArray[i] = this.hashTable[i];
//    }
//    ReAddValues(copiedArray, newCapacity);
//}

//private void ReAddValues(LinkedList<KeyValuePair<K, T>>[] copiedArray, int capacity)
//{
//    this.Capacity = capacity;
//    this.hashTable = new LinkedList<KeyValuePair<K, T>>[this.Capacity];
//    this.Count = 0;
//    for (int i = 0; i < copiedArray.Length; i++)
//    {
//        if (copiedArray[i] != null)
//        {
//            LinkedList<KeyValuePair<K, T>> list = copiedArray[i];
//            foreach (var keyValuePair in list)
//            {
//                this.Add(keyValuePair.Key, keyValuePair.Value);
//            }
//        }
//    }
//}
//}
```

I will send you back the project with fixed bugs. I hope this information helps you. Thank you for the question. Have a nice day.

Best regards,

Vladimir Georgiev