Marco Martinez
CISP 430
3/8/19
Assignment 1

CLASS DIAGRAM

Classes

```
      StringSlot
      GenericSlot
      GenericItemType
      ListEntry
      JList
      Bucket
      HashTable
      Main
```

Associations

```
      StringSlot(1) --- inherits --- (1)GenericSlot
      GenericSlot(1) --- inherits --- (1)GenericItemType
      ListEntry(1) --- includes --- (1)GenericItemType
      JList(1) --- contains --- (m)ListEntry
      Bucket(1) --- inherits --- (1)JList
      HashTable(1) --- contains --- (m)Bucket
      Main(1) --- uses --- (1)HashTable
```

StringSlot Class Attributes

```
      CONSTANT DEFINITIONS
            (+) int MAXBUCKETS

      INSTANCE VARIABLES
            (-) String key
            (-) String data

      CLASS CONSTRUCTORS
            (+) StringSlot()
            (+) StringSlot(String newData)
            (+) StringSlot(StringSlot newSlot)

      CHANGE STATE SERVICES
            (+) void setKey(String newKey)
            (+) void setData(String newData)

      READ STATE SERVICES
            (+) boolean isLess(GenericItemType git)
            (+) boolean isEqual(GenericItemType git)
            (+) boolean isGreater(GenericItemType git)
            (+) int determineIndex()
            (+) String getKey()
            (+) String getData()
            (+) String toString()
```

GenericSlot Class Attributes

```
      READ STATE SERVICES
            (+) abstract int determineIndex()
```

GenericItemType Class Attributes

```
      READ STATE SERVICES
            (+) abstract boolean isLess(GenericItemType git)
            (+) abstract boolean isEqual(GenericItemType git)
            (+) abstract boolean isGreater(GenericItemType git)
```

ListEntry Class Attributes

```
      INSTANCE VARIABLES
```

```
        (-) GenericItemType data
        (-) ListEntry next
        (-) ListEntry prev
```

CLASS CONSTRUCTORS

```
        (+) ListEntry()
        (+) ListEntry(GenericDataItem)
        (+) ListEntry(ListEntry)
```

CHANGE STATE SERVICES

```
        (+) void setData(GenericItemType)
        (+) void setNext(ListEntry)
        (+) void setPrev(ListEntry)
```

READ STATE SERVICES

```
        (+) GenericItemType getData()
        (+) ListEntry getNext()
        (+) ListEntry getPrev()
```

JList Class Attributes

INSTANCE VARIABLES

```
        (-) ListEntry head
        (-) ListEntry tail
        (-) ListEntry currentIteration
        (-) int totalCount
        (-) int currentCount
```

CLASS CONSTRUCTORS

```
        (+) JList()
        (+) JList(GenericItemType)
        (+) JList(ListEntry)
        (+) JList(JList)
        (+) JList(Stack)
        (+) JList(Queue)
        (+) JList(PriorityQueue)
```

CHANGE STATE SERVICES

```
        (+) void init()
        (+) void add_fromHead(GenericItemType)
        (+) void add_fromHead(ListEntry)
        (+) void add_fromMid(GenericItemType)
        (+) void add_fromMid(ListEntry)
        (+) void add_fromTail(GenericItemType)
        (+) void add_fromTail(ListEntry)
        (+) void bubbleSort_ascending()
        (+) void bubbleSort_descending()
        (+) GenericItemType linearSearch(GenericItemType)
        (+) GenericItemType linearSearch(ListEntry)
        (-) ListEntry lSearch(GenericItemType)
        (+) void remove(GenericItemType)
        (+) void remove(ListEntry)
        (-) void delete(GenericItemType)
        (+) void reverseList()
```

READ STATE SERVICES

```
        (+) boolean isFull()
        (+) boolean isEmpty()
        (+) int getCount()
        (+) ListEntry getStart()
        (+) ListEntry getEnd()
        (+) void Iterator_initialize()
        (+) boolean Iterator_hasNext()
        (+) GenericItemType Iterator_iterate()
```

Bucket Class Attributes

```
        CLASS CONSTRUCTORS
                (+) Bucket()
                (+) Bucket(GenericItemType data)
                (+) Bucket(ListEntry le)
                (+) Bucket(JList l)
                (+) Bucket(Bucket b)

        READ STATE SERVICES
                (+) int searchLocation(GenericSlot key)
                (+) int searchLocation(GenericItemType key)
                (+) int searchLocation(ListEntry key)
                (-) int keyLocation(GenericItemType key)
```

HashTable Class Attributes

```
        CONSTANT DEFINITION
                (+) int MAXBUCKETS

        INSTANCE VARIABLES
                (-) Bucket[] ht
                (-) int index

        CLASS CONSTRUCTORS
                (+) HashTable()
                (+) HashTable(GenericSlot gs)
                (+) HashTable(HashTable ht)



        CHANGE STATE SERVICES
                (+) void initialize()
                (+) void insertIntoHT(GenericSlot data)
                (+) GenericItemType searchHT(GenericSlot data)
                (+) void deleteFromHT(GenericSlot data)

        READ STATE SERVICES
                (+) void Iterator_initialize()
                (+) boolean Iterator_hasNext()
                (+) Bucket Iterator_getNext()
                (+) int findLocation(GenericSlot key)
                (+) int getIndex()
                (+) Bucket[] getHashTable()
                (+) Bucket getHashTable(int index)
                (+) int getMax()
```

JAVA SOURCE CODE

```
/**
 @author       Marco Martinez
 @fileName     StringSlot.java
 @version      2.0
 @description  This is a record of StringSlot.
 @date         2/1/2019

 Program Change Log
 ==========================
 Name       Date      Description
 Marco      2/1       Create baseline for Slot.java
 Marco      2/7       Finalize Slot.java
 Marco      3/7       Redesign for reuse
 */

public class StringSlot extends GenericSlot{
   // CONSTANT DEFINITIONS
   public final int MAXBUCKETS = 20;
```

```java
// INSTANCE VARIABLE DECLARATIONS
private String  key,
        data;

// CLASS CONSTRUCTORS
// (+) StringSlot()
public StringSlot() {
    this.key = null;
    this.data = null;
}

// (+) StringSlot(String newData)
public StringSlot(String newData) {
    if (newData != null) {
        this.key = newData.substring(0,9);
        this.data = newData.substring(9);
    } else {
        this.key = null;
        this.data = null;
    }
}

// (+) StringSlot(StringSlot newSlot)
public StringSlot(StringSlot newSlot) {
    this.key = newSlot.key;
    this.data = newSlot.data;
}

// CHANGE STATE SERVICES
// (+) void setKey(String newKey)
public void setKey(String newKey) {
    if (newKey != null)
        this.key = newKey.substring(0,9);
    else
        this.key = null;
}

// (+) void setData(String newData)
public void setData(String newData) {
    if (newData != null)
        this.data = newData;
    else
        this.data = null;
}

// READ STATE SERVICES
// (+) boolean isLess(GenericItemType git)
public boolean isLess(GenericItemType git) { return ( this.key.compareTo(((StringSlot) git).getKey()) < 0); }

// (+) boolean isEqual(GenericItemType git)
public boolean isEqual(GenericItemType git) { return ( this.key.compareTo(((StringSlot) git).getKey()) == 0); }

// (+) boolean isGreater(GenericItemType git)
public boolean isGreater(GenericItemType git) {
    return ( this.key.compareTo(((StringSlot) git).getKey()) > 0);
}

// (+) int determineIndex()
public int determineIndex() {
    byte[] temp = key.getBytes();
    return ((int)temp[1] + (int)temp[3] + (int)temp[5]) % MAXBUCKETS;
}

// (+) String getKey()
public String getKey() { return this.key; }

// (+) String getData()
```

```
    public String getData() { return this.data; }

    // (+) String toString()
    public String toString()
    {
        return this.key + this.data;
    }
}
/**
 @author       Marco Martinez
 @fileName     GenericSlot.java
 @version      1.0
 @description  GenericSlot.
 @date         3/6/2018

 Program Change Log
 ==========================
 Name     Date      Description
 Marco    2/20      Create baseline for GenericSlot.
 */

public abstract class GenericSlot extends GenericItemType{
    // (+) abstract int determineIndex()
    public abstract int determineIndex();
}
/**
    @author       Marco Martinez
    @fileName     GenericItemType.java
    @version      1.0
    @description  Used in Container class as the "only" data type.
    @date         12/18/2018

    Program Change Log
    ==========================
    Name     Date     Description
    Marco    12/18    Create baseline for GenericItemType.
 */

public abstract class GenericItemType {

    // (+) abstract boolean isLess(GenericItemType git)
    public abstract boolean isLess(GenericItemType git);

    // (+) abstract boolean isEqual(GenericItemType git)
    public abstract boolean isEqual(GenericItemType git);

    // (+) abstract boolean isGreater(GenericItemType git)
    public abstract boolean isGreater(GenericItemType git);
}
/**
 @author       Marco Martinez
 @fileName     ListEntry.java
 @version      1.0
 @description  Used in List Container with references to next and previous for bidirectional.
 @date         2/20/2018

 Program Change Log
 ==========================
 Name     Date     Description
 Marco    2/20    Create baseline for ListEntry.
 */

public class ListEntry {
    // (+) INSTANCE VARIABLE DECLARATION
    GenericItemType data;
    ListEntry       next,
                    prev;
    // CLASS CONSTRUCTORS
    // (+) ListEntry()
    public ListEntry() {
        this.data = null;
        this.next = null;
        this.prev = null;
    }

    // (+) ListEntry(GenericItemType data)
    public ListEntry(GenericItemType data) {
        this.data = data;
        this.next = null;
        this.prev = null;
    }

    // (+) ListEntry(ListEntry le)
    public ListEntry(ListEntry le) {
        this.data = le.getData();
        this.next = le.getNext();
        this.prev = le.getPrev();
    }

    // CHANGE STATE SERVICES
    // (+) void setData(GenericItemType data)
    public void setData(GenericItemType data) {
        this.data = data;
    }

    // (+) void setNext(ListEntry next)
    public void setNext(ListEntry next) {
        if (next != null)
            this.next = next;
        else
            this.next = null;
    }

    // (+) void setPrev(ListEntry prev)
    public void setPrev(ListEntry prev) {
        if (prev != null)
            this.prev = prev;
        else
            this.prev = null;
    }

    // READ STATE SERVICES
    // (+) GenericItemType getData()
    public GenericItemType getData() {
        return this.data;
    }

    // (+) ListEntry getNext()
    public ListEntry getNext() {
        return this.next;
    }

    // (+) ListEntry getPrev()
    public ListEntry getPrev() {
        return this.prev;
    }
}
/**
 @author        Marco Martinez
 @fileName      JList.java
 @version       1.0
 @description   Used as pointer based container with "standard" functionality.
 @date          2/20/2018

 Program Change Log
```

```java
===========================
Name     Date     Description
Marco    2/20    Create baseline for JList.
*/

public class JList {
   // INSTANCE VARIABLE DECLARATIONS
   ListEntry   head,
               tail,
               currentIteration;
   int          totalCount,
               currentCount;

   // CLASS CONSTRUCTORS
   // (+) JList()
   public JList() {
      this.head = this.tail = this.currentIteration = null;
      this.currentCount = this.totalCount = 0;
   }

   // (+) JList(GenericItemType data)
   public JList(GenericItemType data) {
      if (data != null) {
         this.head = new ListEntry(data);
         this.head.setNext(null);
         this.head.setPrev(null);
         this.currentIteration = null;
         this.tail = this.head;
         this.totalCount = 1;
         this.currentCount = 0;

      } else {
         this.head = this.tail = this.currentIteration = null;
         this.currentCount = this.totalCount = 0;
      }
   }

   // (+) JList(ListEntry le)
   public JList(ListEntry le) {
      if (le.getData() != null) {
         this.totalCount = 1;
         this.currentCount = 0;
         this.head = this.tail = this.currentIteration = le;
         while (this.currentIteration.getNext() != null) {
            this.currentIteration = this.currentIteration.getNext();
            this.totalCount++;
         }
         this.tail = this.currentIteration;
      } else {
         this.head = this.tail = this.currentIteration = null;
         this.currentCount = this.totalCount = 0;
      }
   }

   // (+) JList(JList l)
   public JList(JList l) {
      this.head = l.getStart();
      this.tail = l.tail;
      this.totalCount = l.getCount();
   }

   // CHANGE STATE SERVICES
   // (+) void init()
   public void init() {
      this.head = this.tail = this.currentIteration = null;
      this.currentCount = this.totalCount = 0;
   }
```

```java
// (+) void add_fromHead(GenericItemType git)
public void add_fromHead(GenericItemType git) {
    if (git != null) {
        if (this.isFull()) {
            this.head.setPrev(new ListEntry(git));
            this.head.getPrev().setNext(this.head);
            this.head = this.head.getPrev();
        } else if (this.isEmpty()) {
            this.head = this.tail = new ListEntry(git);
            this.head.setPrev(null);
            this.head.setNext(null);
        }
        this.totalCount++;
    }
}

// (+) void add_fromMid(GenericItemType git)
public void add_fromMid(GenericItemType git) {
    if (git != null) {
        if (this.isFull()) {
            int mid = this.totalCount / 2;
            this.currentIteration = head;
            for (int i = 0; i < mid-1; i++) {
                this.currentIteration = this.currentIteration.getNext();
            }
            ListEntry temp = this.currentIteration;
            this.currentIteration = new ListEntry(git);
            this.currentIteration.setPrev(temp);
            this.currentIteration.setNext(temp.getNext());
            temp.setNext(this.currentIteration);
            temp = this.currentIteration.getNext();
            temp.setPrev(this.currentIteration);
        } else if (this.isEmpty()) {
            this.head = this.tail = new ListEntry(git);
            this.head.setNext(null);
            this.head.setPrev(null);
        }
        this.totalCount++;
    }

}

// (+) void add_fromTail(GenericItemType git)
public void add_fromTail(GenericItemType git) {
    if (git != null) {
        if (this.isFull()) {
            this.tail.setNext(new ListEntry(git));
            this.tail.getNext().setPrev(this.tail);
            this.tail = this.tail.getNext();
        } else if (this.isEmpty()) {
            this.head = this.tail = new ListEntry(git);
            this.head.setPrev(null);
            this.head.setNext(null);
        }
        this.totalCount++;
    }
}

// (+) void add_fromHead(ListEntry le)
public void add_fromHead(ListEntry le) {
    if (le.getData() != null) {
        JList listTemp = new JList(le);
        if (this.isFull()) {
            this.head.setPrev(listTemp.getEnd());
            this.head.getPrev().setNext(this.head);
            this.head = listTemp.getStart();
            this.totalCount += listTemp.getCount();
        } else if (this.isEmpty()) {
                this.head = listTemp.getStart();
                this.tail = listTemp.getEnd();
                this.totalCount = listTemp.getCount();
            }
        }
    }

    // (+) void add_fromMid(ListEntry le)
    public void add_fromMid(ListEntry le) {
        if (le.getData() != null) {
            JList listTemp = new JList(le);
            if (this.isFull()) {
                int mid = this.totalCount / 2;
                this.currentIteration = head;
                for (int i = 0; i < mid-1; i++) {
                    this.currentIteration = this.currentIteration.getNext();
                }
                ListEntry temp = this.currentIteration.getNext();
                this.currentIteration.setNext(listTemp.getStart());
                this.currentIteration.getNext().setPrev(this.currentIteration);
                temp.setPrev(listTemp.getEnd());
                temp.getPrev().setNext(temp);
                this.totalCount += listTemp.getCount();
            } else if (this.isEmpty()) {
                this.head = listTemp.getStart();
                this.tail = listTemp.getEnd();
                this.totalCount = listTemp.getCount();
            }
        }

    }

    // (+) void add_fromTail(ListEntry le)
    public void add_fromTail(ListEntry le) {
        if (le.getData() != null) {
            JList temp = new JList(le);
            if (this.isFull()) {
                this.tail.setNext(temp.getStart());
                this.tail.getNext().setPrev(this.tail);
                this.tail = temp.getEnd();
                this.totalCount += temp.getCount();
            } else if (this.isEmpty()) {
                this.head = temp.getStart();
                this.tail = temp.getEnd();
                this.totalCount = temp.getCount();
            }
        }
    }

    // (+) void bubbleSort_ascending()
    public void bubbleSort_ascending() {
        this.currentIteration = this.head;

        for (int outer = 0; outer < this.totalCount; outer++) {
            for (int inner = 0; inner < this.totalCount-1; inner++) {
                if (this.currentIteration.getData().isGreater(this.currentIteration.getNext().getData())) {
                    GenericItemType temp = this.currentIteration.getData();
                    this.currentIteration.setData(this.currentIteration.getNext().getData());
                    this.currentIteration.getNext().setData(temp);
                }
                this.currentIteration = this.currentIteration.getNext();
            }
            this.currentIteration = this.head;
        }
    }

    // (+) void bubbleSort_descending()
    public void bubbleSort_descending() {
```

```java
    this.currentIteration = this.head;

    for (int outer = 0; outer < this.totalCount; outer++) {
        for (int inner = 0; inner < this.totalCount-1; inner++) {
            if (this.currentIteration.getData().isLess(this.currentIteration.getNext().getData())) {
                GenericItemType temp = this.currentIteration.getData();
                this.currentIteration.setData(this.currentIteration.getNext().getData());
                this.currentIteration.getNext().setData(temp);
            }
            this.currentIteration = this.currentIteration.getNext();
        }
        this.currentIteration = this.head;
    }
}

// (+) GenericItemType linearSearch(GenericItemType key)
public GenericItemType linearSearch(GenericItemType key) { return new ListEntry(this.lSearch(key)).getData(); }

// (+) GenericItemType linearSearch(ListEntry key)
public GenericItemType linearSearch(ListEntry key) { return new ListEntry(this.lSearch(key.getData())).getData(); }

//  (-) ListEntry lSearch(GenericItemType key)
private ListEntry lSearch(GenericItemType key) {
    this.currentCount = 0;
    this.currentIteration = this.head;
    for (int i = 0; i < this.totalCount; i++) {
        if (this.currentIteration.getData().isEqual(key)) {
            return this.currentIteration;
        }
        this.currentIteration = this.currentIteration.getNext();
        this.currentCount++;
    }
    this.currentCount = 0;
    return new ListEntry();

}

// (+) void remove(GenericItemType key)
public void remove(GenericItemType key) { this.delete(key); }

// (+) void remove(ListEntry key)
public void remove(ListEntry key) { this.delete(key.getData()); }

//  (-) void delete(GenericItemType key)
private void delete(GenericItemType key) {
    this.currentIteration = this.lSearch(key);
    if (this.currentIteration != null) {
        this.currentIteration.setData(this.tail.getData());
        this.tail = this.tail.getPrev();
        this.tail.setNext(null);
        this.totalCount--;
    }
    bubbleSort_ascending();
}

// (+) void reverseList()
public void reverseList() {
    JList temp = new JList();
    this.currentIteration = this.tail;
    for (int i = 0; i < this.totalCount; i++) {
        temp.add_fromTail(this.currentIteration.getData());
        this.currentIteration = this.currentIteration.getPrev();
    }
    this.head = temp.getStart();
    this.tail = temp.getEnd();
    this.totalCount = temp.getCount();
}
```

```java
    // READ STATE SERVICES
    // (+) boolean isFull()
    public boolean isFull() { return this.head != null; }

    // (+) boolean isEmpty()
    public boolean isEmpty() { return this.head == null; }

    // (+) int getCount()
    public int getCount() { return this.totalCount; }

    // (+) ListEntry getStart()
    public ListEntry getStart() { return this.head; }

    // (+) ListEntry getEnd()
    public ListEntry getEnd() { return this.tail; }

    // (+) void Iterator_initialize()
    public void Iterator_initialize() {
        this.currentCount = 0;
        this.currentIteration = this.head;
    }

    // (+) boolean Iterator_hasNext()
    public boolean Iterator_hasNext() {
        if (this.currentCount != 0) {
            if (this.currentIteration.getNext() != null)
                return true;
            else
                return false;
        } else if (this.isFull())
            return true;
        return false;
    }

    // (+) GenericItemType Iterator_iterate()
    public GenericItemType Iterator_iterate() {
        if (this.currentCount < this.totalCount) {
            if (this.currentCount != 0)
                this.currentIteration = this.currentIteration.getNext();
            else {
                this.currentIteration = this.head;
            }
            this.currentCount++;
            return this.currentIteration.getData();
        }
        return new ListEntry().getData();
    }
}
/**
    @author       Marco Martinez
    @fileName     Bucket.java
    @version      2.0
    @description  This is a record of Bucket.
    @date         2/1/2019

    Program Change Log
    ==========================
    Name      Date     Description
    Marco     2/1      Create baseline for Bucket.java
    Marco     2/7      Finalize Bucket.java
    Marco     3/7      Redesign for reuse
 */


public class Bucket extends JList {
    // CLASS CONSTRUCTORS
    // (+) Bucket()
    public Bucket() {
        super();
    }

    // (+) Bucket(GenericItemType data)
    public Bucket(GenericItemType data) {
        super(data);
    }

    // (+) Bucket(ListEntry le)
    public Bucket(ListEntry le) {
        super(le);
    }

    // (+) Bucket(JList l)
    public Bucket(JList l) {
        super(l);
    }

    // (+) Bucket(Bucket b)
    public Bucket(Bucket b) {
        super(b);
    }

    // READ STATE SERVICES
    // (+) int searchLocation(GenericSlot key)
    public int searchLocation(GenericSlot key) {
        return keyLocation(key);
    }

    // (+) int searchLocation(GenericItemType key)
    public int searchLocation(GenericItemType key) {
        return keyLocation(key);
    }

    // (+) int searchLocation(ListEntry key)
    public int searchLocation(ListEntry key) {
        return keyLocation(key.getData());
    }

    //  (-) int keyLocation(GenericItemType key)
    private int keyLocation(GenericItemType key) {
        this.currentCount = 0;
        this.currentIteration = this.head;
        for (int i = 0; i < this.totalCount; i++) {
            if (this.currentIteration.getData().isEqual(key)) {
                return i;
            }
            this.currentIteration = this.currentIteration.getNext();
            this.currentCount++;
        }
        this.currentCount = 0;
        return -1;

    }
}
/**
    @author       Marco Martinez
    @fileName     HashTable.java
    @version      2.0
    @description  Complete redesign with JList reuse.
    @date         2/1/2019

    Program Change Log
    ==========================
    Name     Date     Description
    Marco    2/1      Create baseline for HashTable.java
 */
```

```java
public class HashTable {
  // CONSTANT DEFINITIONS
  public final int MAXBUCKETS = 20;

  // INSTANCE VARIABLE DECLARATIONS
  private Bucket[] ht = new Bucket[MAXBUCKETS];
  private int index;

  // CLASS CONSTRUCTORS
  // (+) HashTable()
  public HashTable() {
    for (int i = 0; i < MAXBUCKETS; i++)
      this.ht[i] = new Bucket();
  }

  // (+) HashTable(GenericSlot gs)
  public HashTable(GenericSlot gs) {
    for (int i = 0; i < MAXBUCKETS; i++)
      this.ht[i] = new Bucket();
    this.ht[gs.determineIndex()] = new Bucket(gs);
  }

  // (+) HashTable(HashTable ht)
  public HashTable(HashTable ht) {
    this.ht = ht.getHashTable();
  }

  // CHANGE STATE SERVICES
  // (+) void initialize()
  public void initialize() {
    for (int i = 0; i < MAXBUCKETS; i++)
      this.ht[i] = new Bucket();
    this.index = 0;
  }

  // (+) void insertIntoHT(GenericSlot data)
  public void insertIntoHT(GenericSlot data) {
    int hashIndex = data.determineIndex();
    if (hashIndex < MAXBUCKETS)
      this.ht[hashIndex].add_fromTail(data);
  }

  // (+) GenericItemType searchHT(GenericSlot data)
  public GenericItemType searchHT(GenericSlot data) {
    int hashIndex = data.determineIndex();
    if (hashIndex < MAXBUCKETS)
      return ht[hashIndex].linearSearch(data);
    return null;
  }

  // (+) void deleteFromHT(GenericSlot data)
  public void deleteFromHT(GenericSlot data) {
    int hashIndex = data.determineIndex();
    if (hashIndex < MAXBUCKETS)
      this.ht[hashIndex].remove(data);
  }

  // READ STATE SERVICES
  // (+) void Iterator_initialize()
  public void Iterator_initialize() {
   this.index = 0;
 }

  // (+) boolean Iterator_hasNext()
  public boolean Iterator_hasNext() {
   return this.index < MAXBUCKETS;
 }
```

```java
   // (+) Bucket Iterator_getNext()
   public Bucket Iterator_iterate() {
     return new Bucket(this.ht[this.index++]);
   }

   // (+) int findLocation(GenericSlot key)
   public int findLocation(GenericSlot key) {
       if (ht[key.determineIndex()].linearSearch(key) != null)
           return this.ht[key.determineIndex()].searchLocation(key);
       return -1;
    }

   // (+) int getIndex()
   public int getIndex() {
     return this.index;
   }

   // (+) Bucket[] getHashTable()
   public Bucket[] getHashTable() {
     return this.ht;
   }

   // (+) Bucket getHashTable(int index)
   public Bucket getHashTable(int index) {
       if (index < MAXBUCKETS)
           return this.ht[index];
       return new Bucket();
    }

   // (+) int getMax()
   public int getMax() {
     return this.MAXBUCKETS;
   }
}

/**
   @author       Marco Martinez
   @fileName      Main.java
   @version      1.0
   @description  First assignment testing.
   @date         2/1/2019

   Program Change Log
   ==========================
   Name     Date     Description
   Marco    2/1      Test for ability to read in and write out text files.
 */

// LIBRARIES
import java.io.*;

public class Main
{
    public static void main(String[] args)
    {
        try {
            HashTable ht = new HashTable();
            InputStream in = new FileInputStream("DATA.dat");
            InputStream isIn = new FileInputStream("SEARCH.dat");
            InputStream load = new FileInputStream("SAVE.dat");
            OutputStream save = new FileOutputStream("SAVE.dat");
            OutputStream preRestore = new FileOutputStream("PRERESTORE.txt");
            OutputStream postRestore = new FileOutputStream("POSTRESTORE.txt");
            OutputStream searchResults = new FileOutputStream("SEARCHRESULTS.txt");
            OutputStream efficiencyResults = new FileOutputStream("EFFICIENCYRESULTS.txt");

            ht.initialize();
            readData(ht,in);
            in.close();
            reportContentsOfHT(ht,preRestore);
            saveState(ht,save);
            save.close();
            loadState(ht,load);
            load.close();
            reportContentsOfHT(ht,postRestore);
            outputSearchResults(ht,isIn,searchResults);
            isIn.close();
            outputEfficiency(ht,efficiencyResults);
            efficiencyResults.close();
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }

    // METHODS
    // (+) static void reportContentsOfHT(HashTable ht,OutputStream out)
    public static void reportContentsOfHT(HashTable ht,OutputStream out) {
        try {
            int i = 1;
            String file = new String("");
            byte[] buffer = new byte[4096];
            ht.Iterator_initialize();
            while (ht.Iterator_hasNext()) {
                Bucket temp = ht.Iterator_iterate();
                file += "Bucket " + Integer.toString(i) + ":\n";
                file += "----------------------------------\n";
                temp.Iterator_initialize();
                while (temp.Iterator_hasNext()) {
                    file += "    " + ((StringSlot)temp.Iterator_iterate()).toString() + "\n";
                }
                file += "\n";
                i++;
            }
            buffer = file.getBytes();
            out.write(buffer);
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }

    // (+) static String[] convertInToString(InputStream in)
    public static String[] convertInToString(InputStream in) {
        try {
            byte[] data = new byte[4096];
            String file;

            in.read(data);
            file = new String(data, "UTF-8");
            return file.split("\\r?\\n");

        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
            return new String[0];
        }
    }

    // (+) static void outputSearchResults(HashTable ht,InputStream isIn, OutputStream searchResults)
    public static void outputSearchResults(HashTable ht,InputStream isIn, OutputStream searchResults) {
        try {
            String[] search = convertInToString(isIn);
            String file = new String("");
            byte[] buffer = new byte[4096];

            file += "    Search Key          Bucket/Position         Record\n";
            file += "    ------------------------------------------------------\n";
            for (int i = 0; i < search.length - 1; i++) {
                if (ht.searchHT(new StringSlot(search[i])) != null) {
                    file += String.format("%-28s","   " + search[i]) + String.format("%-19s",new
StringSlot(search[i]).determineIndex()+1 + "/" + (ht.findLocation(new StringSlot(search[i]))+1)) + String.format("%-
10s",ht.searchHT(new StringSlot(search[i])).toString() + "\n");
                } else {
                    file += "   " + search[i] + "                                was not found.\n";
                }
            }

            buffer = file.getBytes();
            searchResults.write(buffer);
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }

    // (+) static void outputEfficiency(HashTable ht,OutputStream efficiencyResults)
    public static void outputEfficiency(HashTable ht,OutputStream efficiencyResults) {
        try {
            String file = new String("");
            byte[] buffer = new byte[4096];
            int totalCollisions = 0;
            int totalCollidedBuckets = 0;

            file += "   Program Collision Report\n" +
                    "   ---------------------------\n";
            ht.Iterator_initialize();
            while (ht.Iterator_hasNext()) {
                Bucket buc = ht.Iterator_iterate();
                file += "   Bucket " + ht.getIndex() + " has a length of " + buc.getCount() + ".\n";
                if (buc.getCount() > 1) {
                    totalCollisions += buc.getCount();
                    totalCollisions--;
                    totalCollidedBuckets++;
                }
            }
            file += "\n";
            file += "   There were a total of " + totalCollisions + " collisions within this hashing algorithm.\n";
            file += "   The average length per collision was " + ((float)totalCollisions/totalCollidedBuckets) + ".\n";

            buffer = file.getBytes();
            efficiencyResults.write(buffer);
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }

    // (+) static void saveState(HashTable ht, OutputStream out)
    public static void saveState(HashTable ht, OutputStream out) {
        try {
            String file = new String();
            Bucket listTemp;
            byte[] buffer = new byte[4096];
            ht.Iterator_initialize();
            StringSlot temp;
            while (ht.Iterator_hasNext()) {
                listTemp = ht.Iterator_iterate();
                listTemp.Iterator_initialize();
                while (listTemp.Iterator_hasNext()) {
                    temp = (StringSlot) listTemp.Iterator_iterate();
                    file += temp.getKey() + temp.getData() + "\n";
                }
                file += "EOBUCKET\n";
            }
            buffer = file.getBytes();
            out.write(buffer);
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }

    // (+) static void readData(HashTable ht, InputStream in)
    public static void readData(HashTable ht, InputStream in) {
        try {
            byte[] data = new byte[4096];
            String file;
            String[] lines;
            in.read(data);
            file = new String(data, "UTF-8");
            lines = file.split("\\r?\\n");

            for (int i = 0; i < lines.length-1; i++) {
                ht.insertIntoHT(new StringSlot(lines[i]));
            }
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }

    // (+) static void loadState(HashTable ht, InputStream in)
    public static void loadState(HashTable ht, InputStream in) {
        try {
            byte[] data = new byte[4096];
            String file;
            String[] lines;
            in.read(data);
            int count = 0;
            file = new String(data, "UTF-8");
            lines = file.split("\\r?\\n");

            for (int i = 0; i < ht.getMax(); i++) {
                ht = new HashTable();
                while (!lines[count].equals("EOBUCKET")) {
                    if (!lines[count].substring(0,3).equals("null")) {
                        ht.getHashTable(i).add_fromTail(new StringSlot(lines[count]));
                    } else {
                        ht.getHashTable(i).add_fromTail(new StringSlot());
                    }
                    count++;
                }
            }
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }
}
```

REPORTING FILES

PRERESTORE.txt

```
   Bucket 1:
   ----------------------------------
   GENOA SYSTTRIMBLE SAN JOSE, CA

   Bucket 2:
   ----------------------------------
   TATUNG CO.EL PR. LONG BEACH CA
   CORE INTERFEDERA BOCA RATON FL
   CURITS INCUNIO PETERBOROUGH NH

   Bucket 3:
   ----------------------------------
```

```
SIGMA DESIUNIVER A SAN JOSE CA
TAXAN CORPCITY OF INDUSTRY, CA
ORCHID CORWESTINGHO FREMONT CA
MICROWAY CTEMPOHOUSE LONDON UK

Bucket 4:
----------------------------------
MICRODESIGUNVIE WINTER PARK FL
PARADISE STAYLOR S BRISBANE CA
INTERLUDE.RICHMOND HOUSTON, TX
TALLTREE SSNTONIO PALO ALTO CA
PROMETHEUSFREMONT S FREMONT CA
FUNK SOFTW3RD ST CAMBRIDGE, MA

Bucket 5:
----------------------------------
AST RESEARALTON AV IRVINE   CA
DAC SW INCSPRING VAL DALLAS TX
COMPUADD CTECH BLVD. AUSTIN TX

Bucket 6:
----------------------------------
MAYNARD ELSEMOR CASSELBERRY FL

Bucket 7:
----------------------------------
SUMMIT TECBABSON WELLESLEY, MA
MICROGRAFXGREEN RICHARDSON, TX
BUSSIN TOL128 AVE BELLEVUE, WA
DYSAN CORPPAT H SANTA CLARA CA

Bucket 8:
----------------------------------
ROSESOFT CUNIVE WAY SEATTLE WA
OKIDATA COA MOUNTAIN LAUREL NJ

Bucket 9:
----------------------------------
TECMAR INTCOCHRAN RD. SOLON OH
QUANTUM SWSTAFFO OTTAWA CANADA

Bucket 10:
----------------------------------
PRINCETON.EWING S PRINCETON NJ
HEWLETT PABERNARD SAN DIEGO CA
SPECTRUM SWOLFE R SUNNYVALE CA

Bucket 11:
----------------------------------
MAXELL CO.OXFORD  MOONACHIE NJ
SOURCE TELPOBOX 1305 MCLEAN VA
QUBIE CORPCALLE S CAMARILLO CA

Bucket 12:
----------------------------------
KAMERMAN LCIRRUS BEAVERTON, OR
```

```
   EPSON AMERBEDA STR TORRANCE CA
   DIGITAL REGARDEN C MONTEREY CA

   Bucket 13:
   ----------------------------------
   HERCULES CNINTH ST BERKELEY CA
   MICROMART.CAMPUS D NORCROSS GE
   BORTHER I.THENALA DR IRVINE CA

   Bucket 14:
   ----------------------------------
   EXPRESS SYREMING SCHAUMBURG IL
   IOMEGA CORWESTA SOUTH ROY UTAH
   BORLAND I.SCOTTS V. DR S.V. CA

   Bucket 15:
   ----------------------------------

   Bucket 16:
   ----------------------------------
   GRAPHIC CO5TH AVE. WALTHAM, MA
   MICROPRO IBOX 57135 HAYWARD CA
   PROSOFT COBELLAI HOLLEYWOOD CA
   HONEYWELL.BAKER COSTA MESA, CA

   Bucket 17:
   ----------------------------------
   SSISOFTWARCENTER ST. OREM UTAH
   EVEREX SYSMILMONT FREMONT,  CA

   Bucket 18:
   ----------------------------------
   LOGIQUEST.MONTRE QUEBEC CANADA
   IBM CORPORBOCA RATON,  FLORIDA

   Bucket 19:
   ----------------------------------
   QUADRAM COLOACH AV NORCROSS GE
   MICROSTUF,H.W. PKWY ROSWELL GE
   NCR COORPOBOWLING DR DAYTON OH
   EMERSON COSTAN ST SANTA ANA CA
   AMDEK COR.MAINE GROVE VALLY IL

   Bucket 20:
   ----------------------------------
   VEN-TAL INWALSH SANTA CLARA CA
   VICTOR CORSCOTTS VALLEY, CALIF
   INTEL COOR5 ST MOUNTAINVIEW CA
   CHANNELS IKI ST TORONTO CANADA
```

POSTRESTORE.txt

```
   Bucket 1:
   ----------------------------------
   GENOA SYSTTRIMBLE SAN JOSE, CA

   Bucket 2:
```

```
----------------------------------
TATUNG CO.EL PR. LONG BEACH CA
CORE INTERFEDERA BOCA RATON FL
CURITS INCUNIO PETERBOROUGH NH

Bucket 3:
----------------------------------
SIGMA DESIUNIVER A SAN JOSE CA
TAXAN CORPCITY OF INDUSTRY, CA
ORCHID CORWESTINGHO FREMONT CA
MICROWAY CTEMPOHOUSE LONDON UK

Bucket 4:
----------------------------------
MICRODESIGUNVIE WINTER PARK FL
PARADISE STAYLOR S BRISBANE CA
INTERLUDE.RICHMOND HOUSTON, TX
TALLTREE SSNTONIO PALO ALTO CA
PROMETHEUSFREMONT S FREMONT CA
FUNK SOFTW3RD ST CAMBRIDGE, MA

Bucket 5:
----------------------------------
AST RESEARALTON AV IRVINE   CA
DAC SW INCSPRING VAL DALLAS TX
COMPUADD CTECH BLVD. AUSTIN TX

Bucket 6:
----------------------------------
MAYNARD ELSEMOR CASSELBERRY FL

Bucket 7:
----------------------------------
SUMMIT TECBABSON WELLESLEY, MA
MICROGRAFXGREEN RICHARDSON, TX
BUSSIN TOL128 AVE BELLEVUE, WA
DYSAN CORPPAT H SANTA CLARA CA

Bucket 8:
----------------------------------
ROSESOFT CUNIVE WAY SEATTLE WA
OKIDATA COA MOUNTAIN LAUREL NJ

Bucket 9:
----------------------------------
TECMAR INTCOCHRAN RD. SOLON OH
QUANTUM SWSTAFFO OTTAWA CANADA

Bucket 10:
----------------------------------
PRINCETON.EWING S PRINCETON NJ
HEWLETT PABERNARD SAN DIEGO CA
SPECTRUM SWOLFE R SUNNYVALE CA

Bucket 11:
----------------------------------
```

```
MAXELL CO.OXFORD  MOONACHIE NJ
SOURCE TELPOBOX 1305 MCLEAN VA
QUBIE CORPCALLE S CAMARILLO CA

Bucket 12:
----------------------------------
KAMERMAN LCIRRUS BEAVERTON, OR
EPSON AMERBEDA STR TORRANCE CA
DIGITAL REGARDEN C MONTEREY CA

Bucket 13:
----------------------------------
HERCULES CNINTH ST BERKELEY CA
MICROMART.CAMPUS D NORCROSS GE
BORTHER I.THENALA DR IRVINE CA

Bucket 14:
----------------------------------
EXPRESS SYREMING SCHAUMBURG IL
IOMEGA CORWESTA SOUTH ROY UTAH
BORLAND I.SCOTTS V. DR S.V. CA

Bucket 15:
----------------------------------

Bucket 16:
----------------------------------
GRAPHIC CO5TH AVE. WALTHAM, MA
MICROPRO IBOX 57135 HAYWARD CA
PROSOFT COBELLAI HOLLEYWOOD CA
HONEYWELL.BAKER COSTA MESA, CA

Bucket 17:
----------------------------------
SSISOFTWARCENTER ST. OREM UTAH
EVEREX SYSMILMONT FREMONT,  CA

Bucket 18:
----------------------------------
LOGIQUEST.MONTRE QUEBEC CANADA
IBM CORPORBOCA RATON,  FLORIDA

Bucket 19:
----------------------------------
QUADRAM COLOACH AV NORCROSS GE
MICROSTUF,H.W. PKWY ROSWELL GE
NCR COORPOBOWLING DR DAYTON OH
EMERSON COSTAN ST SANTA ANA CA
AMDEK COR.MAINE GROVE VALLY IL

Bucket 20:
----------------------------------
VEN-TAL INWALSH SANTA CLARA CA
VICTOR CORSCOTTS VALLEY, CALIF
INTEL COOR5 ST MOUNTAINVIEW CA
CHANNELS IKI ST TORONTO CANADA
```

SEARCHRESULTS.txt

```
Search Key          Bucket/Position         Record
-------------------------------------------------------
KALLTREE S                                  was not found.
DAC SW INC              5/2                 DAC SW INCSPRING VAL DALLAS TX
COMPUADD C              5/3                 COMPUADD CTECH BLVD. AUSTIN TX
MICROWAY C              3/4                 MICROWAY CTEMPOHOUSE LONDON UK
TATUNG CO.              2/1                 TATUNG CO.EL PR. LONG BEACH CA
DYSAN CORP              7/4                 DYSAN CORPPAT H SANTA CLARA CA
DIGITAL RE              12/3                DIGITAL REGARDEN C MONTEREY CA
QUANTBM SW                                  was not found.
PROMETHEUS              4/5                 PROMETHEUSFREMONT S FREMONT CA
EVEREX SYS              17/2                EVEREX SYSMILMONT FREMONT,  CA
MICROMART.              13/2                MICROMART.CAMPUS D NORCROSS GE
PARADISE S              4/2                 PARADISE STAYLOR S BRISBANE CA
SPECTRUM S              10/3                SPECTRUM SWOLFE R SUNNYVALE CA
MICRODESIG              4/1                 MICRODESIGUNVIE WINTER PARK FL
MAXELL CO.              11/1                MAXELL CO.OXFORD  MOONACHIE NJ
AMDEK COR,              19/5                AMDEK COR.MAINE GROVE VALLY IL
TECMAR INT              9/1                 TECMAR INTCOCHRAN RD. SOLON OH
IBM CORPOR              18/2                IBM CORPORBOCA RATON,  FLORIDA
OHCHID COR                                  was not found.
FUN  SOFTW                                  was not found.
```

EFFICENCYRESULTS.txt

```
Program Collision Report
----------------------------
Bucket 1 has a length of 1.
Bucket 2 has a length of 3.
Bucket 3 has a length of 4.
Bucket 4 has a length of 6.
Bucket 5 has a length of 3.
Bucket 6 has a length of 1.
Bucket 7 has a length of 4.
Bucket 8 has a length of 2.
Bucket 9 has a length of 2.
Bucket 10 has a length of 3.
Bucket 11 has a length of 3.
Bucket 12 has a length of 3.
Bucket 13 has a length of 3.
Bucket 14 has a length of 3.
Bucket 15 has a length of 0.
Bucket 16 has a length of 4.
Bucket 17 has a length of 2.
Bucket 18 has a length of 2.
Bucket 19 has a length of 5.
Bucket 20 has a length of 4.

There was a total of 39 collisions within this hashing algorithm.
The average length per collision was 2.2941177.
```