

Project2.java

```
1 import java.util.*;
2
3
4 /**
5  * The driver program for project2.
6  * This driver reads in a file and hashes them into a table.
7  *
8  * @author Jack Zhan
9  * @version 2016-04-12
10 */
11
12
13 public class Project2 {
14
15     //Storage for the items
16     public static List<Integer> Items = new ArrayList<Integer>();
17
18     /**
19      * Main entry point for the application.
20      */
21     public static void main (String args[]) {
22
23         Project2      p      = new Project2();
24         HashTable      ht     = new HashTable();
25         HashTable2      ht2    = new HashTable2();
26         String          fileName = "input.txt";
27
28         System.out.println("Entered main() method.");
29         p.readInputFile(fileName);
30
31         ht.RunHashTable(Items, 120, 1, "Linear", 1, 120);
32         ht.RunHashTable(Items, 120, 1, "Quadratic", 1, 120);
33         ht2.RunHashTable(Items, 120, 1, 120);
34
35         ht.RunHashTable(Items, 120, 1, "Linear", 1, 113);
36         ht.RunHashTable(Items, 120, 1, "Quadratic", 1, 113);
37         ht2.RunHashTable(Items, 120, 1, 113);
38
39         ht.RunHashTable(Items, 40, 3, "Linear", 1, 41);
40         ht.RunHashTable(Items, 40, 3, "Quadratic", 1, 41);
41
42         ht.RunHashTable(Items, 120, 1, "Linear", 2, 120);
43         ht.RunHashTable(Items, 120, 1, "Quadratic", 2, 120);
44         ht2.RunHashTable(Items, 120, 2, 120);
45
46         return;
47     }
48
49     /**
50      * Opens, reads, and closes the file containing items.
51      */
52
53     private void readInputFile(String fileName) {
54
55         String line = null;
56
57         System.out.println("Entered readInputFile() method.");
58
59         try {
```

Project2.java

```
61      // FileReader reads text files in the default encoding.
62      FileReader fileReader = new FileReader(fileName);
63
64      // Always wrap FileReader in BufferedReader.
65      BufferedReader bufferedReader = new BufferedReader(fileReader);
66
67      while((line = bufferedReader.readLine()) != null) {
68          //Resets everything when there is a blank line and
69          //adds the data to array lists
70          if(line.isEmpty() ){
71              }
72          else
73          {
74              //Getting order and setting up the matrixes to be
75              //added into the array list
76              Items.add(Integer.parseInt(line));
77
78          }
79      }
80      // Always close files.
81      bufferedReader.close();
82  }
83  catch(FileNotFoundException ex) {
84      System.out.println(
85          "Unable to open file '" +
86          fileName + "'");
87  }
88  catch(IOException ex) {
89      System.out.println(
90          "Error reading file '"
91          + fileName + "'");
92  }
93  return;
94  }
95
96 }
97
```