

```

1 package ch.publiccept;
2
3 import java.util.ArrayList;
4
5 /**
6  * GroceryList
7  *
8  * public:
9  * addGroceryItem(String item)
10 * printGroceryList()
11 * modifyGroceryItem(String newItem)
12 * removeGroceryItem(String itemToRemove)
13 * isItemInList()
14 *
15 * getters:
16 * getGroceryList()
17 *
18 * private:
19 * modifyGroceryItem(int i, String newItem)
20 * removeGroceryItem(int i)
21 * findItem(String searchItem)
22 *
23 * @author created by Urs Albisser, on 2020-01-28
24 * @version 0.1
25 */
26 public class GroceryList {
27
28
29
30     // == fields ==
31     // ArrayList<String> groceryList = new ArrayList<String>();
32     // explicit type argument ArrayList<String>() can be replaced with ArrayList<>()
33     // ArrayList<>() -> () brackets = call empty constructor of the ArrayList class.
34     private ArrayList<String> groceryList = new ArrayList<>();
35
36
37
38     // == public methods ==
39     /**
40      * addGroceryItem()
41      * Add a new item to the ArrayList.
42      * @param item New item String to add
43      */
44     public void addGroceryItem(String item) {
45
46         /*
47          * ArrayList.add(item)
48          */
49         groceryList.add(item);
50     }
51
52
53     /**
54      * printGroceryList()
55      * Print the whole ArrayList to the console.
56      */
57     public void printGroceryList() {
58         /*
59          * ArrayList.size()
60          */
61         System.out.println("You have " + groceryList.size() + " items in your grocery list.");
62
63         for(int i=0; i<groceryList.size(); i++)
64             /*
65              * ArrayList.get(index)
66              */
67             System.out.println((i+1) + ". " + groceryList.get(i));
68     }
69
70
71     /**
72      * modifyGroceryItem()
73      * @param newItem
74      */
75     public void modifyGroceryItem(String currentItem, String newItem) {
76         int i = findItem(currentItem);
77         if (i>=0) {
78             modifyGroceryItem(i, newItem);
79             System.out.println("Grocery item " + currentItem + " has been modified to " + newItem);
80         } else if (i<0) {
81             System.out.println("Item " + newItem + " not found.");
82         }
83     }
84
85
86     public void removeGroceryItem(String itemToRemove) {
87         int i = findItem(itemToRemove);
88         if (i>=0) {
89             removeGroceryItem(i);
90             System.out.println("Grocery item " + itemToRemove + " has been removed.");
91         } else if (i<0) {
92             System.out.println("Item " + itemToRemove + " not found.");
93         }
94     }
95
96
97     /**

```

```

98     * isItemInList()
99     * Find item in the ArrayList
100    * @param searchItem Item String to search for
101    * @return true if found, otherwise false
102    */
103    public boolean isItemInList(String searchItem) {
104        // complicated
105        int i = findItem(searchItem);
106        if(i >= 0) {
107            return true;
108        }
109        return false;
110
111        /*
112         * ArrayList.contains(item);
113         */
114        // best practice
115        return groceryList.contains(searchItem);
116    }
117
118
119
120    // == getters ==
121    /**
122     * getGroceryList()
123     * @return groceryList ArrayList<String> Object
124     */
125    public ArrayList<String> getGroceryList() {
126        return groceryList;
127    }
128
129
130
131    // == private methods ==
132    /**
133     * modifyGroceryItem()
134     * Modifies the item at index i.
135     * @param i index
136     * @param newItem New String to replace the former item
137     */
138    private void modifyGroceryItem(int i, String newItem) {
139        /*
140         * ArrayList.set(index, item)
141         */
142        groceryList.set(i, newItem); // i = index
143        System.out.println("Grocery item " + (i + 1) + " modified to " + newItem);
144    }
145
146
147    /**
148     * removeGroceryItem()
149     * Removes the item with index i from the ArrayList.
150     * @param i index
151     */
152    private void removeGroceryItem(int i) {
153        /*
154         * ArrayList.remove(index)
155         */
156        groceryList.remove(i);
157    }
158
159
160    /**
161     * findItem()
162     * Find item in the ArrayList
163     * @param searchItem Item String to search for
164     * @return returns the item index if found, otherwise -1 is returned
165     */
166    private int findItem(String searchItem) {
167        /*
168         * ArrayList.indexOf(item)
169         */
170        // Returns the index of the first occurrence of the specified element in this list, or
171        // returns -1 if this list does not contain the element.
172        return groceryList.indexOf(searchItem);
173    }
174 }
175

```