|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/foodManagement/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SortedListOfImmutables.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | |  |
| [**PREV CLASS**](http://docs.google.com/foodManagement/Restaurant.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?foodManagement/SortedListOfImmutables.html)    [**NO FRAMES**](http://docs.google.com/SortedListOfImmutables.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **foodManagement**

Class SortedListOfImmutables

java.lang.Object  
 **foodManagement.SortedListOfImmutables**

public class **SortedListOfImmutables**extends java.lang.Object

A SortedListOfImmutables represents a sorted collection of immutable objects that implement the Listable interface. An array of references to Listable objects is used internally to represent the list. The items in the list are always kept in alphabetical order based on the names of the items. When a new item is added into the list, it is inserted into the correct position so that the list stays ordered alphabetically by name.

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**SortedListOfImmutables**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#SortedListOfImmutables())()            This constructor creates an empty list by creating an internal array of size 0. |
| [**SortedListOfImmutables**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#SortedListOfImmutables(foodManagement.SortedListOfImmutables))([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) other)            Copy constructor. |

|  |  |
| --- | --- |
| **Method Summary** | |
| void | [**add**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#add(foodManagement.Listable))([Listable](http://docs.google.com/foodManagement/Listable.html) itemToAdd)            Adds an item to the list. |
| void | [**add**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#add(foodManagement.SortedListOfImmutables))([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) listToAdd)            Adds an entire list of items to the current list, maintaining the alphabetical ordering of the list by the names of the items. |
| boolean | [**checkAvailability**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#checkAvailability(foodManagement.Listable))([Listable](http://docs.google.com/foodManagement/Listable.html) itemToFind)            Checks to see if a particular item is in the list. |
| boolean | [**checkAvailability**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#checkAvailability(foodManagement.SortedListOfImmutables))([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) listToCheck)            Checks if a list of items is contained in the current list. |
| [Listable](http://docs.google.com/foodManagement/Listable.html) | [**get**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#get(int))(int i)            Returns a reference to the item in the ith position in the list. |
| int | [**getRetailValue**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#getRetailValue())()            Returns the sum of the retail values of all items in the list. |
| int | [**getSize**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#getSize())()            Returns the number of items in the list. |
| int | [**getWholesaleCost**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#getWholesaleCost())()            Returns the sum of the wholesale costs of all items in the list. |
| void | [**remove**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#remove(foodManagement.Listable))([Listable](http://docs.google.com/foodManagement/Listable.html) itemToRemove)            Removes an item from the list. |
| void | [**remove**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#remove(foodManagement.SortedListOfImmutables))([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) listToRemove)            Removes an entire list of items from the current list. |
| java.lang.String | [**toString**](http://docs.google.com/foodManagement/SortedListOfImmutables.html#toString())() |

|  |
| --- |
| **Methods inherited from class java.lang.Object** |
| equals, getClass, hashCode, notify, notifyAll, wait, wait, wait |

|  |
| --- |
| **Constructor Detail** |

### SortedListOfImmutables

public **SortedListOfImmutables**()

This constructor creates an empty list by creating an internal array of size 0. (Note that this is NOT the same thing as setting the internal instance variable to null.)

### SortedListOfImmutables

public **SortedListOfImmutables**([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) other)

Copy constructor. The current object will become a copy of the list that the parameter refers to. The copy must be made in such a way that future changes to either of these two lists will not affect the other. In other words, after this constructor runs, adding or removing things from one of the lists must not have any effect on the other list.

**Parameters:**other - the list that is to be copied

|  |
| --- |
| **Method Detail** |

### getSize

public int **getSize**()

Returns the number of items in the list.

**Returns:**number of items in the list

### get

public [Listable](http://docs.google.com/foodManagement/Listable.html) **get**(int i)

Returns a reference to the item in the ith position in the list. (Indexing is 0-based, so the first element is element 0).

**Parameters:**i - index of item requested **Returns:**reference to the ith item in the list

### add

public void **add**([Listable](http://docs.google.com/foodManagement/Listable.html) itemToAdd)

Adds an item to the list. This method assumes that the list is already sorted in alphabetical order based on the names of the items in the list. The new item will be inserted into the list in the appropriate place so that the list will remain alphabetized by names. In order to accomodate the new item, the internal array must be re-sized so that it is one unit larger than it was before the call to this method.

**Parameters:**itemToAdd - refers to a Listable item to be added to this list

### add

public void **add**([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) listToAdd)

Adds an entire list of items to the current list, maintaining the alphabetical ordering of the list by the names of the items.

**Parameters:**listToAdd - a list of items that are to be added to the current object

### remove

public void **remove**([Listable](http://docs.google.com/foodManagement/Listable.html) itemToRemove)

Removes an item from the list. If the list contains the same item that the parameter refers to, it will be removed from the list. If the item appears in the list more than once, just one instance will be removed. If the item does not appear on the list, then this method does nothing.

**Parameters:**itemToRemove - refers to the item that is to be removed from the list

### remove

public void **remove**([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) listToRemove)

Removes an entire list of items from the current list. Any items in the parameter that appear in the current list are removed; any items in the parameter that do not appear in the current list are ignored.

**Parameters:**listToRemove - list of items that are to be removed from this list

### getWholesaleCost

public int **getWholesaleCost**()

Returns the sum of the wholesale costs of all items in the list.

**Returns:**sum of the wholesale costs of all items in the list

### getRetailValue

public int **getRetailValue**()

Returns the sum of the retail values of all items in the list.

**Returns:**sum of the retail values of all items in the list

### checkAvailability

public boolean **checkAvailability**([Listable](http://docs.google.com/foodManagement/Listable.html) itemToFind)

Checks to see if a particular item is in the list.

**Parameters:**itemToFind - item to look for **Returns:**true if the item is found in the list, false otherwise

### checkAvailability

public boolean **checkAvailability**([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) listToCheck)

Checks if a list of items is contained in the current list. (In other words, this method will return true if ALL of the items in the parameter are contained in the current list. If anything is missing, then the return value will be false.)

**Parameters:**listToCheck - list of items that may or may not be a subset of the current list **Returns:**true if the parameter is a subset of the current list; false otherwise

### toString

public java.lang.String **toString**()

**Overrides:**toString in class java.lang.Object

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/foodManagement/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SortedListOfImmutables.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | |  |
| [**PREV CLASS**](http://docs.google.com/foodManagement/Restaurant.html)   NEXT CLASS | [**FRAMES**](http://docs.google.com/index.html?foodManagement/SortedListOfImmutables.html)    [**NO FRAMES**](http://docs.google.com/SortedListOfImmutables.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[Web Accessibility](https://www.umd.edu/web-accessibility)