|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/foodManagement/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Restaurant.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/Listable.html)   [**NEXT CLASS**](http://docs.google.com/foodManagement/SortedListOfImmutables.html) | [**FRAMES**](http://docs.google.com/index.html?foodManagement/Restaurant.html)    [**NO FRAMES**](http://docs.google.com/Restaurant.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **foodManagement**

Class Restaurant

java.lang.Object  
 **foodManagement.Restaurant**

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

The Restaurant has a name (String), a menu (list of Entrees), an inventory (list of Food), and an amount of cash on hand, measured in pennies (int) This class facilitates orders being placed, deliveries being made to the inventory, and entrees being added to the menu.

|  |  |
| --- | --- |
| **Constructor Summary** | |
| [**Restaurant**](http://docs.google.com/foodManagement/Restaurant.html#Restaurant(java.lang.String,%20int))(java.lang.String nameIn, int startingCash)            Standard constructor. |

|  |  |
| --- | --- |
| **Method Summary** | |
| void | [**addEntree**](http://docs.google.com/foodManagement/Restaurant.html#addEntree(foodManagement.Entree))([Entree](http://docs.google.com/foodManagement/Entree.html) entreeToAdd)            Adds an entree to the menu. |
| boolean | [**addShipmentToInventory**](http://docs.google.com/foodManagement/Restaurant.html#addShipmentToInventory(foodManagement.SortedListOfImmutables))([SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) list)            Adds the specified list of food items to the inventory. |
| boolean | [**checkIfInInventory**](http://docs.google.com/foodManagement/Restaurant.html#checkIfInInventory(foodManagement.Entree))([Entree](http://docs.google.com/foodManagement/Entree.html) entree)            Checks if the Food items contained in the specified Entree are actually contained in the restaurant's inventory. |
| int | [**getCash**](http://docs.google.com/foodManagement/Restaurant.html#getCash())()            Getter for the current amount of cash on hand |
| [SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) | [**getInventory**](http://docs.google.com/foodManagement/Restaurant.html#getInventory())()            Getter for the inventory. |
| [SortedListOfImmutables](http://docs.google.com/foodManagement/SortedListOfImmutables.html) | [**getMenu**](http://docs.google.com/foodManagement/Restaurant.html#getMenu())()            Getter for the menu. |
| java.lang.String | [**getName**](http://docs.google.com/foodManagement/Restaurant.html#getName())()            Getter for the name of the restaurant. |
| boolean | [**placeOrder**](http://docs.google.com/foodManagement/Restaurant.html#placeOrder(foodManagement.Entree))([Entree](http://docs.google.com/foodManagement/Entree.html) entree)            Removes the food items contained in the specified Entree from the inventory. |

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

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

### Restaurant

public **Restaurant**(java.lang.String nameIn,  
 int startingCash)

Standard constructor. The menu and the inventory are both initialized as empty lists. The name and cash amount are set to match the paramters.

**Parameters:**nameIn - name of the restaurantstartingCash - cash amount that the restaurant will have, measured in pennies

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

### getName

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

Getter for the name of the restaurant.

**Returns:**reference to the name of the restaurant

### getMenu

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

Getter for the menu.

**Returns:**a reference to a copy of the menu

### addEntree

public void **addEntree**([Entree](http://docs.google.com/foodManagement/Entree.html) entreeToAdd)

Adds an entree to the menu.

**Parameters:**entreeToAdd - reference to the entree to be added to the menu

### getInventory

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

Getter for the inventory.

**Returns:**a reference to a copy of the inventory

### getCash

public int **getCash**()

Getter for the current amount of cash on hand

**Returns:**the current amount of cash, measured in pennies

### checkIfInInventory

public boolean **checkIfInInventory**([Entree](http://docs.google.com/foodManagement/Entree.html) entree)

Checks if the Food items contained in the specified Entree are actually contained in the restaurant's inventory.

**Parameters:**entree - Entree that we are checking against the inventory **Returns:**true if the list of Food items contained in the Entree are all present in the inventory, false otherwise.

### addShipmentToInventory

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

Adds the specified list of food items to the inventory. If the total wholesale cost of all of the food items combined exceeds the amount of cash on hand, then NONE of the food items are added to the inventory. If the amount of cash on hand is sufficient to pay for the shipment, then the amount of cash on hand is reduced by the wholesale cost of the shipment.

**Parameters:**list - food items to be added to the inventory **Returns:**true if the food items are added; false if the food items are not added because their wholesale cost exceeds the current cash on hand

### placeOrder

public boolean **placeOrder**([Entree](http://docs.google.com/foodManagement/Entree.html) entree)

Removes the food items contained in the specified Entree from the inventory. If the inventory does not contain all of the items required for this Entree, then NOTHING is removed from the inventory. If the inventory contains all of the required items, then the amount of cash on hand is INCREASED by the retail value of the Entree.

**Parameters:**entree - Entree that has been ordered **Returns:**true if the food items are removed from the inventory; false if the food items were not removed because one or more required items were not contained in the inventory

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | [**Package**](http://docs.google.com/foodManagement/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Restaurant.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/Listable.html)   [**NEXT CLASS**](http://docs.google.com/foodManagement/SortedListOfImmutables.html) | [**FRAMES**](http://docs.google.com/index.html?foodManagement/Restaurant.html)    [**NO FRAMES**](http://docs.google.com/Restaurant.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

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