

| Manager  |
|--|
| <div>- isOccupied : boolean</div> <div>- name : String</div> <div>- requestingredients = String</div>  |
| <div>+ Manager(name : String)</div> <div>+ callWorker(workers : IWorker []): IWorker</div> <div>+ notifyLowStock(ingredient : String)</div> <div>+ sendMail(): String</div> <div>+ sendMail(amount : int): String</div> <div>+ isOccupied(): boolean</div> <div>+ checkInventory(inventory : Inventory)</div> <div>+ scanStock(inventory : Inventory, ingredient : String, amount : int)</div> |

| Bill   |
|--|
|  |
| <div>+ outputBill(orders : Dish[])</div> <div>+ outputSingleBill(table : Table, customerNum : int)</div> |

| Dish  |
|---|
| <div>- tableName : String</div> <div>- customerNum : int</div> <div>- comment : String</div>  |
| <div>+ Dish(dish : MenuItem, tableName : String, customerNum : int)</div> <div>+ addComment(comment : String)</div> <div>+ getCustomerNum(): int</div> <div>+ toString(): String</div> <div>+ getStringForBill(): String</div> <div>+ getTableName(): String</div> <div>+ getTableDifferences(): String</div> <div>+ getDishFromMenuItem(menu : MenuItem, tableName : String, customerNumber : int): Dish</div> |

| Cook  |
|---|
| <div>- isOccupied : boolean</div> <div>- name : String</div>  |
| <div>+ Cook(name: String)</div> <div>+ getName(): String</div> <div>+ isOccupied(): boolean</div> <div>+ prepareDish(dish : Dish, inventory : Inventory, screen : ServingTable): boolean</div> <div>+ dishReady(dish : Dish, screen : ServingTable)</div> <div>+ scanStock(inventory : Inventory, ingredient : String, amount : int)</div> <div>+ canBePrepared(dish : Dish, inventory : Inventory): boolean</div> <div>+ update(dish : Dish)</div> |

| DishIngredient   |
|--|
| <div>- name : String</div> <div>- amount : int</div> <div>- baseAmount : int</div> <div>- upperAmount : int</div> <div>- lowerAmount : int</div> <div>- price : double</div>   |
| <div>+ DishIngredient(name : String, baseAmount : int, amount : int, upperAmount : int, lowerAmount : int, price : double)</div> <div>+ addAmount(amount : int)</div> <div>+ getAmount(): int</div> <div>+ subtractAmount(amount : int)</div> <div>+ amountCanBeAdded(amount : int): boolean</div> <div>+ amountCanBeSubtracted(amount : int): boolean</div> <div>+ clone(): DishIngredient</div> <div>+ toString(): String</div> <div>+ getBaseAmount(): int</div> <div>+ getAdditionAmount(): int</div> <div>+ getBaseDifference(): int</div> <div>+ getPrice(): double</div> <div>+ getName(): String</div> |

| <<TableListener>>   |
|---|
|   |
| <div>+ getTableOrder(name : String): Dish</div> <div>+ getBill()</div> <div>+ returnDish(dish : MenuItem)</div> |

| <<ServerTableListener>>          |
|----------------------------------|
|                                  |
| <div>+ update(dish : Dish)</div> |

| <<InventoryListener>>                            |
|--|
|  |
| <div>+ notifyLowStock(ingredient : String)</div> |

| <<IWorker>>  |
|--|
|  |
| <div>+ isOccupied(): boolean</div> <div>+ scanStock(inventory : Inventory, ingredient: String, amount : int)</div> |

| Inventory  |
|--|
| <div>- INVENTORY : String</div> <div>- REQUESTFILE : String</div> <div>- manager : InventoryListener</div> <div>- inventory : HashMap&lt;String, InventoryIngredient&gt;</div>   |
| <div>+ addStock(ingredient : String, amount : int)</div> <div>+ hasEnoughIngredients(dishIngredients : HashMap&lt;String, Integer&gt;): boolean</div> <div>+ removeStock(ingredient : String, amount : int)</div> <div>+ writeToInventory()</div> <div>+ readInventory()</div> <div>+ getLowIngredients()</div> <div>+ getIngredientAmounts(name : String): int</div> <div>+ toString(): String</div> <div>+ setManager(manager : InventoryListener)</div> |

| InventoryIngredient  |
|--|
| <div>- name : String</div> <div>- currentQuantity : int</div> <div>- restockQuantity : int</div> <div>- lowerThreshold : int</div>   |
| <div>+ InventoryIngredient(name : String, amount : int, restockQuantity : int, lowerThreshold : int)</div> <div>+ getCurrentQuantity(): int</div> <div>+ setCurrentQuantity(currentQuantity : int)</div> <div>+ increaseQuantity(amount : int)</div> <div>+ decreaseQuantity(amount : int)</div> <div>+ getName(): String</div> <div>+ setName(name : String)</div> <div>+ getRestockQuantity(): int</div> <div>+ setRestockQuantity(restockQuantity)</div> <div>+ getLowerThreshold(): int</div> <div>+ setLowerThreshold(lowerThreshold : int)</div> |

| Menu   |
|--|
| <div>- FILENAME : String</div> <div>- menuItems : HashMap&lt;double, MenuItem&gt;</div>  |
| <div>+ Menu()</div> <div>+ createMenu()</div> <div>+ addDishToMenu(line : String)</div> <div>+ getMenuSize(): int</div> <div>+ getDish(id : double, tableName : String, customerNum : int): Dish</div> |

| MenuItem  |
|---|
| <div>- name : String</div> <div>- id : double</div> <div>- price : double</div> <div>- ingredients : HashMap&lt;String, DishIngredients&gt;</div>   |
| <div>+ MenuItem(name : String, id : double, price : double, ingredients : HashMap&lt;String, DishIngredients&gt;)</div> <div>+ getId(): double</div> <div>+ clone(): MenuItem</div> <div>+ cloneIngredients(): HashMap&lt;String, DishIngredient&gt;</div> <div>+ getPrice(): double</div> <div>+ getIngredients(): HashMap&lt;String, DishIngredient&gt;</div> <div>+ addIngredient(ingredient : String, amount : int)</div> <div>+ subtractIngredient(ingredient : String, amount : int)</div> <div>+ getIngredientAmounts(): HashMap&lt;String, Integer&gt;</div> <div>+ getDifferenceAmounts(): HashMap&lt;String, Integer&gt;</div> <div>+ getName(): String</div> |

| Restaurant   |
|--|
| <div>- inventory : Inventory</div> <div>- menu : Menu</div> <div>- workers : IWorker []</div> <div>- servers : HashMap&lt;String, Server&gt;</div> <div>- cooks : HashMap&lt;String, Cook&gt;</div> <div>- tables : HashMap&lt;String, Table&gt;</div> <div>- manager : Manager</div> <div>- servingTable : ServingTable</div>   |
| <div>+ Restaurant(menu : Menu, inventory : Inventory, servingTable : ServingTable)</div> <div>+ getServingTable(): ServingTable</div> <div>+ getServer(): Server</div> <div>+ getCook(): Cook</div> <div>+ getTable(): Table</div> <div>+ getMenu(): Menu</div> <div>+ getInventory(): Inventory</div> <div>+ addWorker(worker : IWorker)</div> <div>+ getWorkers(): IWorker []</div> <div>- generateTables(fileName : String)</div> <div>- generateWorkers(fileName : String, servingTable : ServingTable)</div> <div>+ setManager(managerName : String)</div> <div>+ getManager(): Manager</div> |

| RestaurantSimulation   |
|--|
|  |
| <div>+ main(args : String [])</div> <div>- readEvents(fileName : String, restaurant : Restaurant)</div> <div>- readServerAction(input : String [], restaurant : Restaurant)</div> <div>- readCookAction(input : String [], restaurant : Restaurant)</div> <div>- readManagerAction(input : String [], restaurant : Restaurant)</div> |

| Server   |
|--|
| <div>- name : String</div> <div>- isOccupied : boolean</div>   |
| <div>+ Server(name : String)</div> <div>+ getName(): String</div> <div>+ passOrder(table : Table, servingTable : ServingTable)</div> <div>+ serveDish(index : int, restaurant : Restaurant)</div> <div>+ rejectDish(index : int, restaurant : Restaurant)</div> <div>+ returnDish(index : int, tableID, restaurant : Restaurant, comment : String)</div> <div>+ addOrder(table : Table, dish : Dish, servingTable : ServingTable)</div> <div>+ seatCustomer(tableID : String, restaurant : Restaurant)</div> <div>+ generateTableBill(table : Table)</div> <div>+ generateSingleBill(table : Table, size : int)</div> <div>+ scanStock(inventory : Inventory, ingredient : String, amount : int)</div> <div>+ isOrderComplete(table : Table): boolean</div> <div>+ isOccupied(): boolean</div> <div>+ setOccupied(occupied : boolean)</div> <div>+ updateDish(dish : Dish)</div> |

| <<ServerListener>>                          |
|---|
|   |
| <div>+ callManager(manager : Manager)</div> |

| ServingTable  |
|---|
| <div>- dishesRejected : Dish []</div> <div>- dishesToBeCooked : Dish []</div> <div>- dishesToBeServed : Dish []</div> <div>- dishesBeingCooked : Dish []</div> <div>- servers : ServingTableListener []</div> <div>- cooks : ServingTableListener []</div>  |
| <div>ServingTable()</div> <div>+ getServers(): ServerTableListener []</div> <div>+ setServers(servers : ServingTableListener [])</div> <div>+ getCooks(): ServerTableListener []</div> <div>+ setCooks(cooks : ServerTableListener [])</div> <div>+ getDishesRejected(): Dish []</div> <div>+ getDishesToBeCooked(): Dish []</div> <div>+ getDishesBeingCooked(): Dish []</div> <div>+ getDishesToBeServed(): Dish []</div> <div>+ addToBeCooked(order : Dish [])</div> <div>+ addToBeCooked(order : Dish)</div> <div>+ returnDish(dish : Dish, comment : String)</div> <div>+ addToBeServed(dish : Dish)</div> <div>+ notifyServers(dish : Dish)</div> <div>+ notifyCooks(dish : Dish)</div> <div>+ getDishesToBeServed(index : int): Dish</div> <div>+ addServer(server : ServingTableListener)</div> <div>+ addServers(servers : ServingTableListener [])</div> <div>+ addCook(cook : ServingTableListener)</div> <div>+ addCooks(cooks : ServingTableListeners [])</div> <div>+ toString(): String</div> <div>+ getRejectedDish(): Dish</div> <div>+ getDishesToBeCooked(index : int): Dish</div> <div>+ setDishToServe(index : int)</div> <div>+ setDishToCooking(index : int): Dish</div> <div>+ serveDish(index : int): Dish</div> |

| Table   |
|---|
| <div>- tableName : String</div> <div>- tableSize : int</div> <div>- isOccupied : boolean</div> <div>- tableOrder : Dish []</div> <div>- orderCompleted : boolean</div> <div>- cookedOrder : Dish []</div> <div>- numberOfDishesServed : int</div>   |
| <div>+ Table(tableName : String, tableSize : int, isOccupied : int)</div> <div>+ Table(tableName : String, tableSize : int)</div> <div>+ isOccupied(): boolean</div> <div>+ setOccupied(occupied : boolean)</div> <div>+ setTableOrder(tableOrder : Dish [])</div> <div>+ addSingleOrder(dish : Dish)</div> <div>+ isOrderCompleted(): boolean</div> <div>+ setOrderCompleted(orderCompleted : boolean)</div> <div>+ getCookedOrder(): Dish []</div> <div>+ setCookedOrder(Dish [])</div> <div>+ getTableOrder(): Dish []</div> <div>+ dishesServed()</div> <div>+ removeDish(dish : Dish)</div> <div>+ setNumberOfDishesServed(numberOfDishesServed : int)</div> <div>+ numberOfDishesInOrder(): int</div> <div>+ getNumberOfDishesServed(): int</div> <div>+ getTableName(): String</div> <div>+ getCustomerOrder(): Dish []</div> <div>+ getDish(index : int): Dish</div> <div>+ getName(): String</div> <div>+ clearTable()</div> |