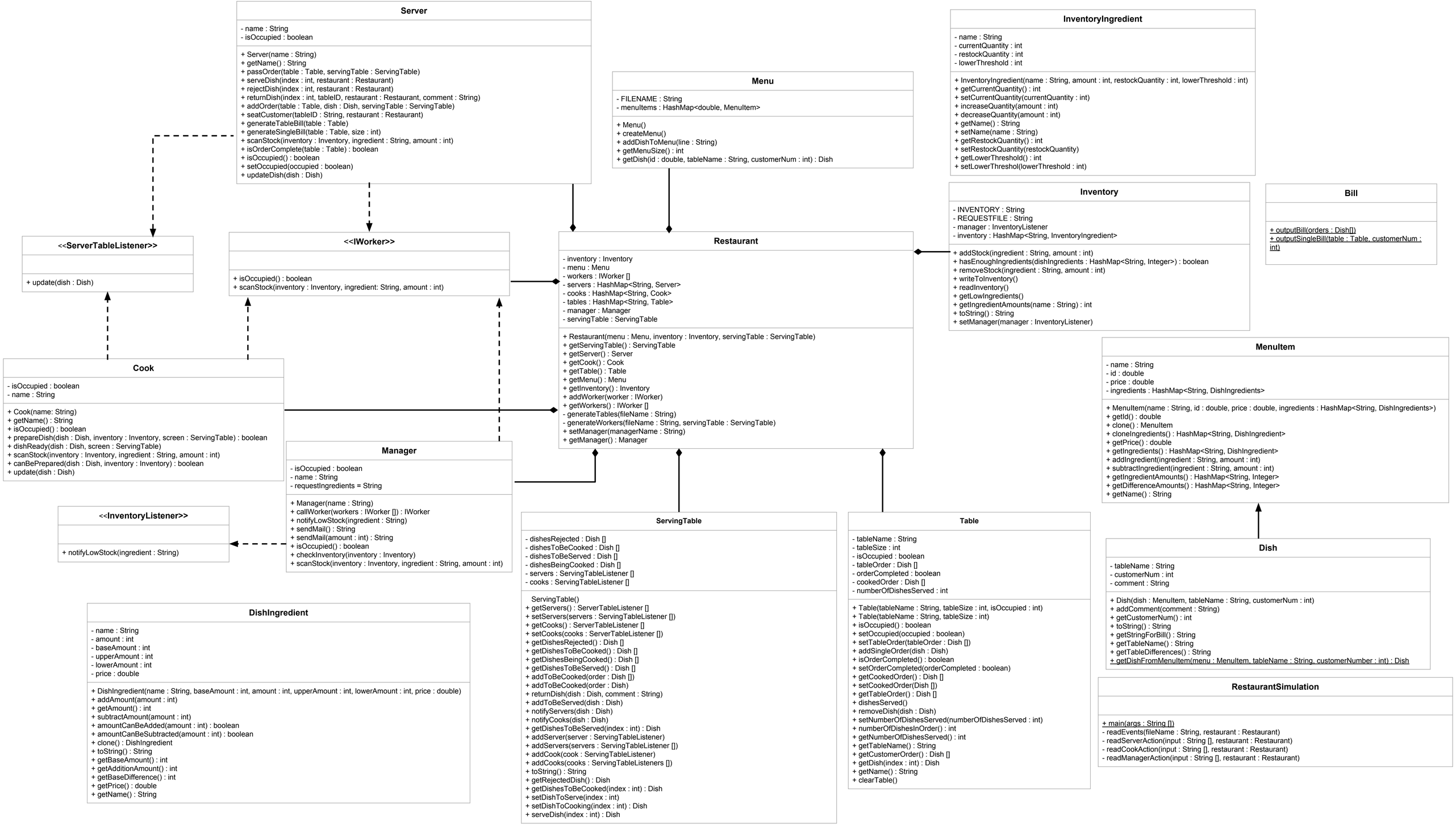
Server
- name : String
- isOccupied : boolean
+ Server(name : String)
+ getName() : String
+ passOrder(table : Table, servingTable : ServingTable)
+ serveDish(index : int, restaurant : Restaurant)
+ rejectDish(index : int, restaurant : Restaurant)
+ returnDish(index : int, tableID, restaurant : Restaurant, comment : String)
+ addOrder(table : Table, dish : Dish, servingTable : ServingTable)
+ seatCustomer(tableID : String, restaurant : Restaurant)
+ generateTableBill(table : Table)
+ generateSingleBill(table : Table, size : int)
+ scanStock(inventory : Inventory, ingredient : String, amount : int)
+ isOrderComplete(table : Table) : boolean
+ isOccupied() : boolean
+ setOccupied(occupied : boolean)
+ updateDish(dish : Dish)
Menu
- FILENAME : String
- menuItems : HashMap<double, MenuItem>
+ Menu()
+ createMenu()
+ addDishToMenu(line : String)
+ getMenuSize() : int
+ getDish(id : double, tableName : String, customerNum : int) : Dish
InventoryIngredient
- name : String
- currentQuantity : int
- restockQuantity : int
- lowerThreshold : int
+ InventoryIngredient(name : String, amount : int, restockQuantity : int, lowerThreshold : int)
+ getCurrentQuantity() : int
+ setCurrentQuantity(currentQuantity : int)
+ increaseQuantity(amount : int)
+ decreaseQuantity(amount : int)
+ getName() : String
+ setName(name : String)
+ getRestockQuantity() : int
+ setRestockQuantity(restockQuantity)
+ getLowerThreshold() : int
+ setLowerThreshold(lowerThreshold : int)
Inventory
- INVENTORY : String
- REQUESTFILE : String
- manager : InventoryListener
- inventory : HashMap<String, InventoryIngredient>
+ addStock(ingredient : String, amount : int)
+ hasEnoughIngredients(dishIngredients : HashMap<String, Integer>) : boolean
+ removeStock(ingredient : String, amount : int)
+ writeToInventory()
+ readInventory()
+ getLowIngredients()
+ getIngredientAmounts(name : String) : int
+ toString() : String
+ setManager(manager : InventoryListener)
Bill
+ outputBill(orders : Dish[])
+ outputSingleBill(table : Table, customerNum : int)
<<ServerTableListener>>
+ update(dish : Dish)
<<IWorker>>
+ isOccupied() : boolean
+ scanStock(inventory : Inventory, ingredient: String, amount : int)
Restaurant
- inventory : Inventory
- menu : Menu
- workers : IWorker []
- servers : HashMap<String, Server>
- cooks : HashMap<String, Cook>
- tables : HashMap<String, Table>
- manager : Manager
- servingTable : ServingTable
+ Restaurant(menu : Menu, inventory : Inventory, servingTable : ServingTable)
+ getServingTable() : ServingTable
+ getServer() : Server
+ getCook() : Cook
+ getTable() : Table
+ getMenu() : Menu
+ getInventory() : Inventory
+ addWorker(worker : IWorker)
+ getWorkers() : IWorker []
- generateTables(fileName : String)
- generateWorkers(fileName : String, servingTable : ServingTable)
+ setManager(managerName : String)
+ getManager() : Manager
MenuItem
- name : String
- id : double
- price : double
- ingredients : HashMap<String, DishIngredients>
+ MenuItem(name : String, id : double, price : double, ingredients : HashMap<String, DishIngredients>)
+ getId() : double
+ clone() : MenuItem
+ cloneIngredients() : HashMap<String, DishIngredient>
+ getPrice() : double
+ getIngredients() : HashMap<String, DishIngredient>
+ addIngredient(ingredient : String, amount : int)
+ subtractIngredient(ingredient : String, amount : int)
+ getIngredientAmounts() : HashMap<String, Integer>
+ getDifferenceAmounts() : HashMap<String, Integer>
+ getName() : String
Cook
- isOccupied : boolean
- name : String
+ Cook(name: String)
+ getName() : String
+ isOccupied() : boolean
+ prepareDish(dish : Dish, inventory : Inventory, screen : ServingTable) : boolean
+ dishReady(dish : Dish, screen : ServingTable)
+ scanStock(inventory : Inventory, ingredient : String, amount : int)
+ canBePrepared(dish : Dish, inventory : Inventory) : boolean
+ update(dish : Dish)
Manager
- isOccupied : boolean
- name : String
- requestIngredients = String
+ Manager(name : String)
+ callWorker(workers : IWorker []) : IWorker
+ notifyLowStock(ingredient : String)
+ sendMail() : String
+ sendMail(amount : int) : String
+ isOccupied() : boolean
+ checkInventory(inventory : Inventory)
+ scanStock(inventory : Inventory, ingredient : String, amount : int)
<<InventoryListener>>
+ notifyLowStock(ingredient : String)
ServingTable
- dishesRejected : Dish []
- dishesToBeCooked : Dish []
- dishesToBeServed : Dish []
- dishesBeingCooked : Dish []
- servers : ServingTableListener []
- cooks : ServingTableListener []
ServingTable()
+ getServers() : ServerTableListener []
+ setServers(servers : ServingTableListener [])
+ getCooks() : ServerTableListener []
+ setCooks(cooks : ServerTableListener [])
+ getDishesRejected() : Dish []
+ getDishesToBeCooked() : Dish []
+ getDishesBeingCooked() : Dish []
+ getDishesToBeServed() : Dish []
+ addToBeCooked(order : Dish [])
+ addToBeCooked(order : Dish)
+ returnDish(dish : Dish, comment : String)
+ addToBeServed(dish : Dish)
+ notifyServers(dish : Dish)
+ notifyCooks(dish : Dish)
+ getDishesToBeServed(index : int) : Dish
+ addServer(server : ServingTableListener)
+ addServers(servers : ServingTableListener [])
+ addCook(cook : ServingTableListener)
+ addCooks(cooks : ServingTableListeners [])
+ toString() : String
+ getRejectedDish() : Dish
+ getDishesToBeCooked(index : int) : Dish
+ setDishToServe(index : int)
+ setDishToCooking(index : int) : Dish
+ serveDish(index : int) : Dish
Table
- tableName : String
- tableSize : int
- isOccupied : boolean
- tableOrder : Dish []
- orderCompleted : boolean
- cookedOrder : Dish []
- numberOfDishesServed : int
+ Table(tableName : String, tableSize : int, isOccupied : int)
+ Table(tableName : String, tableSize : int)
+ isOccupied() : boolean
+ setOccupied(occupied : boolean)
+ setTableOrder(tableOrder : Dish [])
+ addSingleOrder(dish : Dish)
+ isOrderCompleted() : boolean
+ setOrderCompleted(orderCompleted : boolean)
+ getCookedOrder() : Dish []
+ setCookedOrder(Dish [])
+ getTableOrder() : Dish []
+ dishesServed()
+ removeDish(dish : Dish)
+ setNumberOfDishesServed(numberOfDishesServed : int)
+ numberOfDishesInOrder() : int
+ getNumberOfDishesServed() : int
+ getTableName() : String
+ getCustomerOrder() : Dish []
+ getDish(index : int) : Dish
+ getName() : String
+ clearTable()
Dish
- tableName : String
- customerNum : int
- comment : String
+ Dish(dish : MenuItem, tableName : String, customerNum : int)
+ addComment(comment : String)
+ getCustomerNum() : int
+ toString() : String
+ getStringForBill() : String
+ getTableName() : String
+ getTableDifferences() : String
+ getDishFromMenuItem(menu : MenuItem, tableName : String, customerNumber : int) : Dish
DishIngredient
- name : String
- amount : int
- baseAmount : int
- upperAmount : int
- lowerAmount : int
- price : double
+ DishIngredient(name : String, baseAmount : int, amount : int, upperAmount : int, lowerAmount : int, price : double)
+ addAmount(amount : int)
+ getAmount() : int
+ subtractAmount(amount : int)
+ amountCanBeAdded(amount : int) : boolean
+ amountCanBeSubtracted(amount : int) : boolean
+ clone() : DishIngredient
+ toString() : String
+ getBaseAmount() : int
+ getAdditionAmount() : int
+ getBaseDifference() : int
+ getPrice() : double
+ getName() : String
RestaurantSimulation
+ main(args : String [])
- readEvents(fileName : String, restaurant : Restaurant)
- readServerAction(input : String [], restaurant : Restaurant)
- readCookAction(input : String [], restaurant : Restaurant)
- readManagerAction(input : String [], restaurant : Restaurant)