FoodQuick

Henri Branken | March 3, 2023

Customer

+ orderNumber : String
+ name : String
+ contactNumber : String
+ address : String
+ city : String
+ email : String

+ setOrderNumber() : String
+ setName() : String
+ setContactNumber() : String
+ setAddress() : String
+ setCity() : String
+ setEmail() : String
+ toString() : String

Restaurant

+ name : String
+ city : String
+ contactNumber : String
+ mealAndQty : Map<String, Integer>
+ mealAndPrice : Map<String, Float>
+ specialInstr : String
+ total : float

+ setName() : String
+ setCity() : String
+ setContactNumber() : String
+ setMealAndQuantity() : Map<String, Integer>
+ setMealAndPrice() : Map<String, Float>
+ setSpecialInstr() : String
+ setTotal(mapQty : Map<String, Integer>, mapPrice : Map<String, Float>) : float
+ toString() : String

`QuickFood1.main(Str[] args)`

Create one **`Customer`** object and one **`Restaurant`** object

Invoke **`getDriver(cityToMatch:String)`** to find the closest driver with the smallest load.
Return **`chosenDriver:String`**

Invoke **`printInvoice(cObj:Customer, rObj:Restaurant, driver:String)`** to populate text variable **`inv`**.
`Return inv:String`

Create **`File`** object **`"src/data/invoice.txt"`**.
Create **FileWriter** object **`FileWriter("src/data/invoice.txt")`**

Write contents of **`inv`** to **`"src/data/invoice.txt"`**