Implementation Manual

A screenshot of a menu

AI-generated content may be incorrect.

* Public class Café with public method start. Private methods are openMenu, printReceipt, getRewardLevel, and getItemPrice.
* HashMap called orderList with String key and Integer value.
* The start method starts the program. It uses VBox, HBox, Text, Image and ImageViewer, and Button to create the welcome window.
  + The button uses the event handler to call the openMenu method.
* openMenu uses VBox, HBox, Text, Image and ImageViewer, and Button to create the menu window.
  + Each drink has a button below it to add it to the order. The button uses the event handler. We use the put method for our HashMap as well as getOrDefault. We also add 1 at the end to ensure that each time the button is pressed, the quantity increases.
    - Ex. (button.setOnAction(e -> {orderList.put("Hot Black Coffee", orderList.getOrDefault("Hot Black Coffee", 0) + 1);});)
* printReceipt uses VBox, Text, Image and ImageViewer to create the receipt window.
  + A for loop iterates through the HashMap using the entryset() and get() methods.
    - Ex. (String itemName = entry.getKey(); )
  + A double variable called price calls the getItemPrice method in order to get the price for each drink.
    - Ex. (double price = getItemPrice(itemName);)
  + One of the Texts calls the getRewardLevel method to print out the rank.
    - Ex. (Text extraText4 = new Text("Your Reward Level: " + getrewardLevel(totalAmount));)
* getRewardLevel takes in a double totalAmount and returns a String rank.
  + It uses if and else statements to determine the correct rank for the totalAmount.
* getItemPrice takes in a String itemName.
  + It utilizes a switch statement. Each case is the name of a drink, and it returns the price. The default returns 0.