

# Report Workshop #1: Code Standards

## Introduction

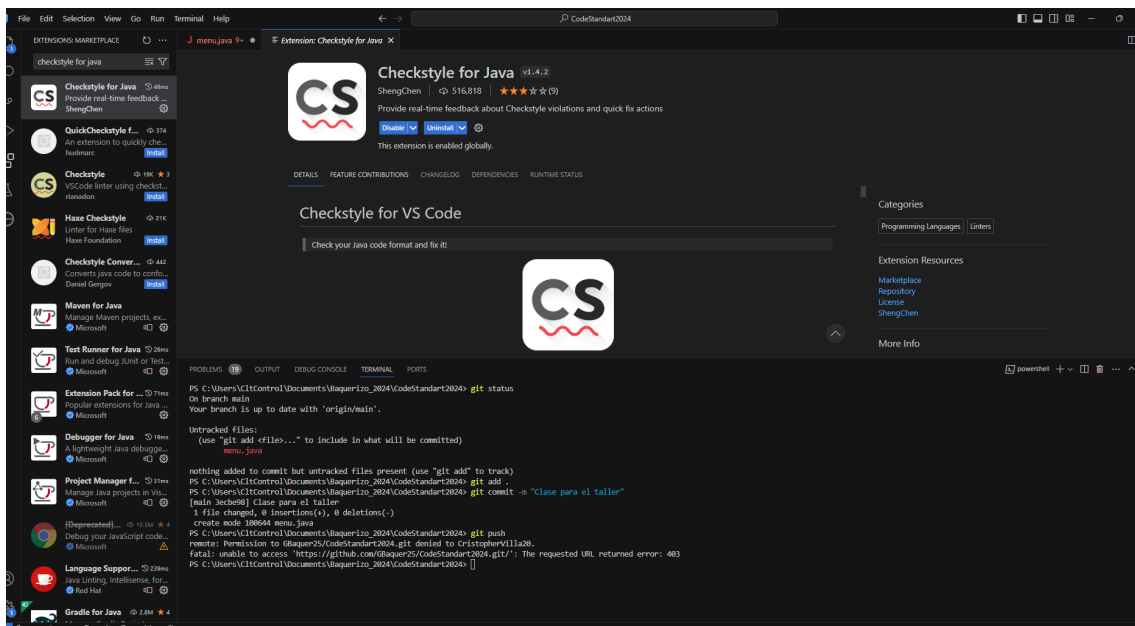
### Selected Language: Java

I chosen java because I understand it better.

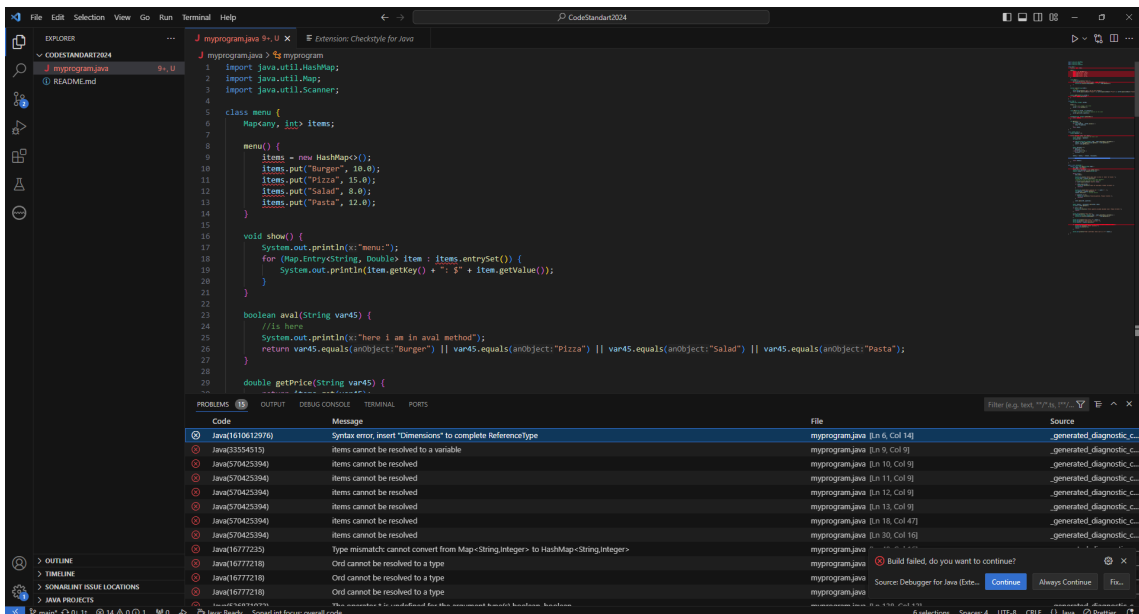
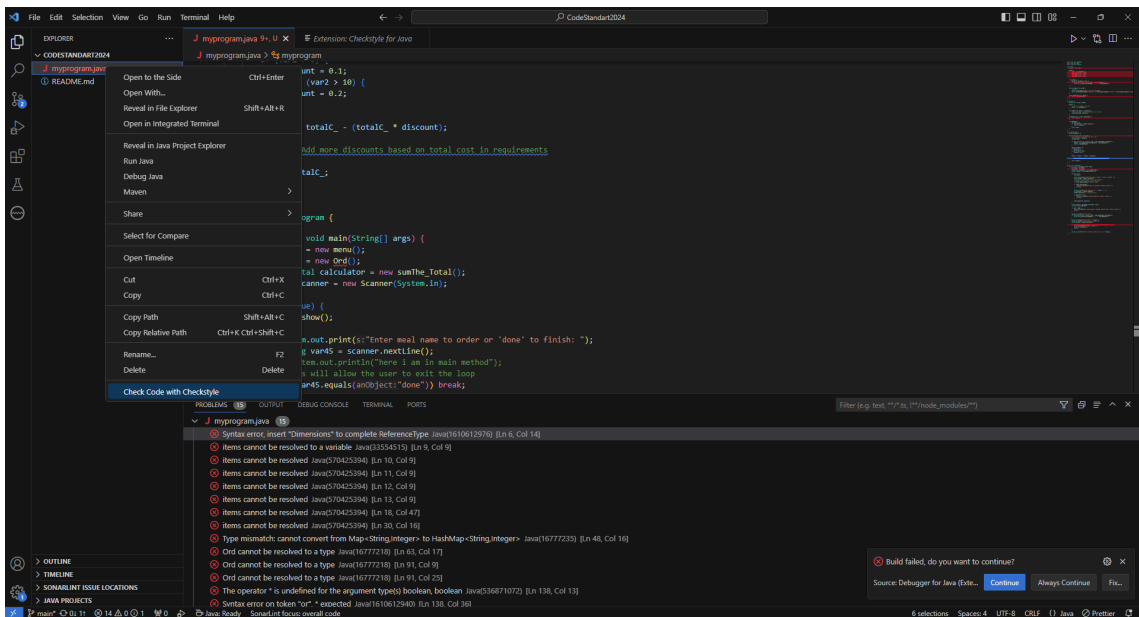
### Selected Tool: CheckStyle.

I chosen it because I heard about it before.

### Steps:



### Applying checkstyle in the code:



# Correcting errors:

Before:

1) The type in the Map is not correct. They will be change in String and Double types.

```

class menu {
    Map<any, int> items;

    menu() {
        items = new HashMap<>();
        items.put("Burger", 10.0);
        items.put("Pizza", 15.0);
        items.put("Salad", 8.0);
        items.put("Pasta", 12.0);
    }
}

```

2) The name of the class is incorrect

```

8 public class myprogram {
    Run | Debug
9     public static void main(String[] args) {
10         menu menu = new menu();
11         Ord order = new Ord();
12         sumThe_Total calculator = new sumThe_Total();
13         Scanner scanner = new Scanner(System.in);
14

```

```

15 double calc(Ord order, menu menu) {
16     // my function to calculate the total cost
17     double totalC_ = baseCost;
18     int var2 = 0;
19
20     for (Map.Entry<String, Integer> item : order.getvar45s().entrySet()) {
21         totalC_ += menu.getPrice(item.getKey()) * item.getValue();
22         var2 += item.getValue();
23     }
24
25     double discount = 0;
26     if (var2 > 5) {
27         discount = 0.1;
28     } else if (var2 > 10) {
29         discount = 0.2;
30     }
31 }

```

3) The type HashMap is not compatible, it will be change for Map.

```

32 HashMap<String, Integer> getvar45s() {
33     return var45s;
34 }

```

4) The conditional is incorrect, the correct is &&.

```
if (!confirm.equals(anObject:"yes") or !confirm.equals(anObject:"YES")) {  
    System.out.println(x:"Order canceled.");  
    System.out.println(-1);  
    return;  
}
```

5) This is a note, no error.

```
    // TODO: Add more discounts based on total cost in requirements  
  
    return totalC_;  
}  
  
public class myprogram {  
    public static void main(String[] args) {  
        // TODO: Add more discounts based on total cost in requirements  
    }  
}
```

After

1)

```
class menu {  
    Map<String, Double> items;  
  
    menu() {  
        items = new HashMap<>();  
        items.put("Burger", 10.0);  
        items.put("Pizza", 15.0);  
        items.put("Salad", 8.0);  
        items.put("Pasta", 12.0);  
    }  
}
```

2)

```
public class myprogram {  
    Run | Debug  
    public static void main(String[] args) {  
        menu menu = new menu();  
        Order order = new Order();  
        sumThe_Total calculator = new sumThe_Total();  
        Scanner scanner = new Scanner(System.in);  
  
        while (true) {  
            menu.show();  
        }  
    }  
}
```

```

double calc([Order order, menu menu]) {
    // my function to calculate the total cost
    double totalC_ = baseCost;
    int var2 = 0;

    for (Map.Entry<String, Integer> item : order.getvar45s().entrySet()) {
        totalC_ += menu.getPrice(item.getKey()) * item.getValue();
        var2 += item.getValue();
    }

    double discount = 0;
    if (var2 > 5) {
        discount = 0.1;
    }
}

```

3)

```

Map<String, Integer> getvar45s() {
    return var45s;
}

```

4)

```

if (!confirm.equals(anObject:"yes") && !confirm.equals(anObject:"YES")) {
    System.out.println(x:"Order canceled.");
    System.out.println(-1);
    return;
}

```

## Requirements

- 1) The system should display a menu with various dining options and their corresponding prices.

```

→ CodeStandart2024 git:(main) cd /home/daniela/Documents/Tallere
deStandart2024 ; /usr/bin/env /usr/lib/jvm/java-11-openjdk-amd64/b
er/workspaceStorage/27fc87a6fdbdb1d4466868549f3f2e/redhat.java/j
gram
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: 

```

Users can select multiple meals to order and specify the quantity for each meal.

```
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: Pasta
here i am in aval method
Enter quantity for Pasta: 3
```

2. Meal Quantity Validation o Validate that the quantity entered for each meal is a positive integer greater than zero or prompt users to re-enter quantities if invalid quantities are entered.

```
Enter meal name to order or 'done' to finish: Pasta
here i am in aval method
Enter quantity for Pasta: -1
Invalid quantity. Please re-enter.
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: 
```

```
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: 0
here i am in aval method
meal not available. Please re-select.
```

### 3. Cost Calculation

o The base cost for a dining experience is \$5, applied to every order.

```
Enter meal name to order or 'done' to finish: Pasta
here i am in aval method
Enter quantity for Pasta: 3
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: done
Your Ord:
Pasta: 3
Total Cost: $41.0
Confirm order (yes/no): 
```

- o Apply a discount of 10% if the total quantity of meals ordered is more than 5.

```
Enter quantity for Pasta: 7
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: done
Your Ord:
Pasta: 7
Total Cost: $80.1
Confirm order (yes/no): yes
Order confirmed. Total cost is: $80.1
```

- o Apply a discount of 20% if the total quantity of meals ordered is more than 10.

```
Enter meal name to order or 'done' to finish: Pasta
here i am in aval method
Enter quantity for Pasta: 11
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: done
Your Ord:
Pasta: 11
Total Cost: $123.3
Confirm order (yes/no): yes
Order confirmed. Total cost is: $123.3
```

#### 4. Special Offer Discount

- o Apply special discounts based on total cost conditions.

```
double discountTotal = 0;
if (totalC_ > 50.00 && totalC_ < 100.00) {
    discountTotal = 10.00;
} else if (totalC_ > 100.00) {
    discountTotal = 25.00;
}
totalC_ = totalC_ - discountTotal;
```

- o Apply a \$10 discount if the total cost exceeds \$50.



```

Enter meal name to order or 'done' to finish: Pasta
here i am in aval method
Enter quantity for Pasta: 7
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: done
Your Ord:
Pasta: 7
Total Cost: $70.1
Confirm order (yes/no): yes
Order confirmed. Total cost is: $70.1

```

Applied the discount of 10 dollars.

o Apply a \$25 discount if the total cost exceeds \$100.

```

Enter meal name to order or 'done' to finish: Pasta
here i am in aval method
Enter quantity for Pasta: 11
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: done
Entro en mayor a 100 y el total hasta ahora es123.3
Your Ord:
Pasta: 11
Total Cost: $98.3
Confirm order (yes/no): yes
Order confirmed. Total cost is: $98.3

```

## 5. Meal Availability

o Validate that the selected meals are available on the menu.

o Display an error message and prompt users to re-select meals if an unavailable meal is selected.

```

Enter meal name to order or 'done' to finish: 1000
here i am in aval method
meal not available. Please re-select.
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: █

```

## 6. Maximum Order Quantity

- o Handle orders with a maximum quantity of 100 meals.
- o Display an error message and prompt users to re-enter quantities if attempting to order more than 100 meals.

```

Enter quantity for Pasta: 3
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: Pasta
here i am in aval method
Enter quantity for Pasta: 99
Error: Maximum order quantity is 100 meals for Pasta. Your order has been increased to the remaining available quantity.
You add 97
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: Salad
here i am in aval method
Enter quantity for Salad: 4
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: done
Your Ord:
Pasta: 100
Salad: 4
Total Cost: $1088.3
Confirm order (yes/no): yes
Order confirmed. Total cost is: $1088.3

```

## 7. User Confirmation

- o Before finalizing the order, display selected meals, quantities, and total cost for user confirmation.

```
Enter meal name to order or 'done' to finish: done
Your Ord:
Pasta: 8
Total Cost: $80.9
Confirm order (yes/no): yes
Order confirmed. Total cost is: $80.9
```

## 8. Output

o Provide the total cost of the dining experience as a positive integer if the order is confirmed and valid.

```
Confirm order (yes/no): yes
Order confirmed. Total cost is: $80.9
```

o Return a value of -1 if the input data is not valid or if the order is canceled.

```
Salad: $8.0
Enter meal name to order or 'done' to finish: done
Your Ord:
Pasta: 3
Total Cost: $41.0
Confirm order (yes/no): no
Order canceled.
-1
```

```
menu:
Burger: $10.0
Pizza: $15.0
Pasta: $12.0
Salad: $8.0
Enter meal name to order or 'done' to finish: Pasta
here i am in aval method
Enter quantity for Pasta: hola
Invalid input. Please enter a number.
-1
```

## 9. Error Handling

o Handle errors gracefully and provide clear error messages for invalid input or calculation errors.

Use -1 output only as a last resort for invalid input, validating every user-entered input.

**Conclusions:**

- I have any problems for coding every requirement, the factor time is important to consider.
- This workshop teach me about the importance of the correct tool for coding standards. If we use it, the corrections are faster in the code.

**Recommendations:**

- To do quickly the job use tools for your questions, read and pass to code, no kill too much time read and proof your ideas.
- Experiment and check your checkstyle, maybe origin any new error and you have to correct it.