**CafeX**

(Technical exercise for Scala position)

Date: 04/06/2017

Name: Myilvahanan Duraipandian

Git Repo: <https://github.com/myildurai/CafeX>

Contents

[Purpose: 3](#_Toc484374982)

[Brief: 3](#_Toc484374983)

[Checkin 1: 4](#_Toc484374984)

[Checkin 2: 4](#_Toc484374985)

[Checkin 3 (Step 1): 5](#_Toc484374986)

[Checkin 4 (Step 1): 5](#_Toc484374987)

[Checkin 5 (Step 2): 6](#_Toc484374988)

[Checkin 6 (Step 2): 6](#_Toc484374989)

# Purpose:

This document is to explain the process of how the exercise was developed and checked in. The requirement given was …

Basic exercise

● Use TDD

● Use Git and tag commits (e.g. after step 1)

Complete the steps in order. Don’t read ahead.

At each step build the simplest possible solution which meets our requirement.

Tag a git commit after each step so that your approach is clear.

Scenario

Cafe X menu consists of the following items:

● Cola - Cold - 50p

● Coffee - Hot - £1.00

● Cheese Sandwich - Cold - £2.00

● Steak Sandwich - Hot - £4.50

Customers don’t know how much to tip and staff need tips to survive!

Step 1 : Standard Bill

Pass in a list of purchased items that produces a total bill.

e.g. [“Cola”, “Coffee”, “Cheese Sandwich”] returns 3.5

Step 2: Service Charge

Add support for a service charge :

● When all purchased items are drinks no service charge is applied

● When purchased items include any food apply a service charge of 10% to the total bill

(rounded to 2 decimal places)

● When purchased items include any hot food apply a service charge of 20% to the total bill

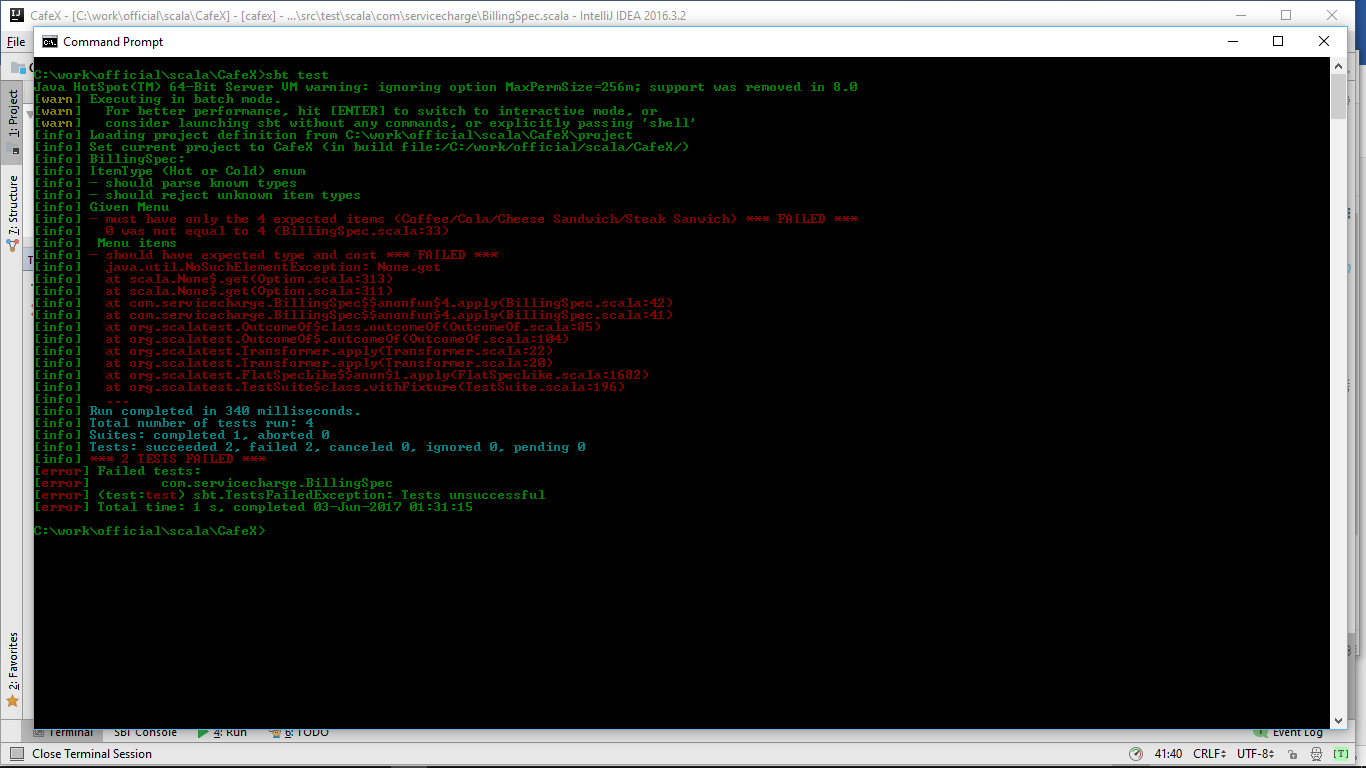
with a maximum £20 service charge

# Brief:

The basis of TDD is to develop tests first with minimal components required without the actual implementation. Execute the test and record the failure first before the actual business components are developed. The business components are then developed and amended until the tests defined succeeds. The business component developed is quite straight forward and below are the order of Github checkin followed.

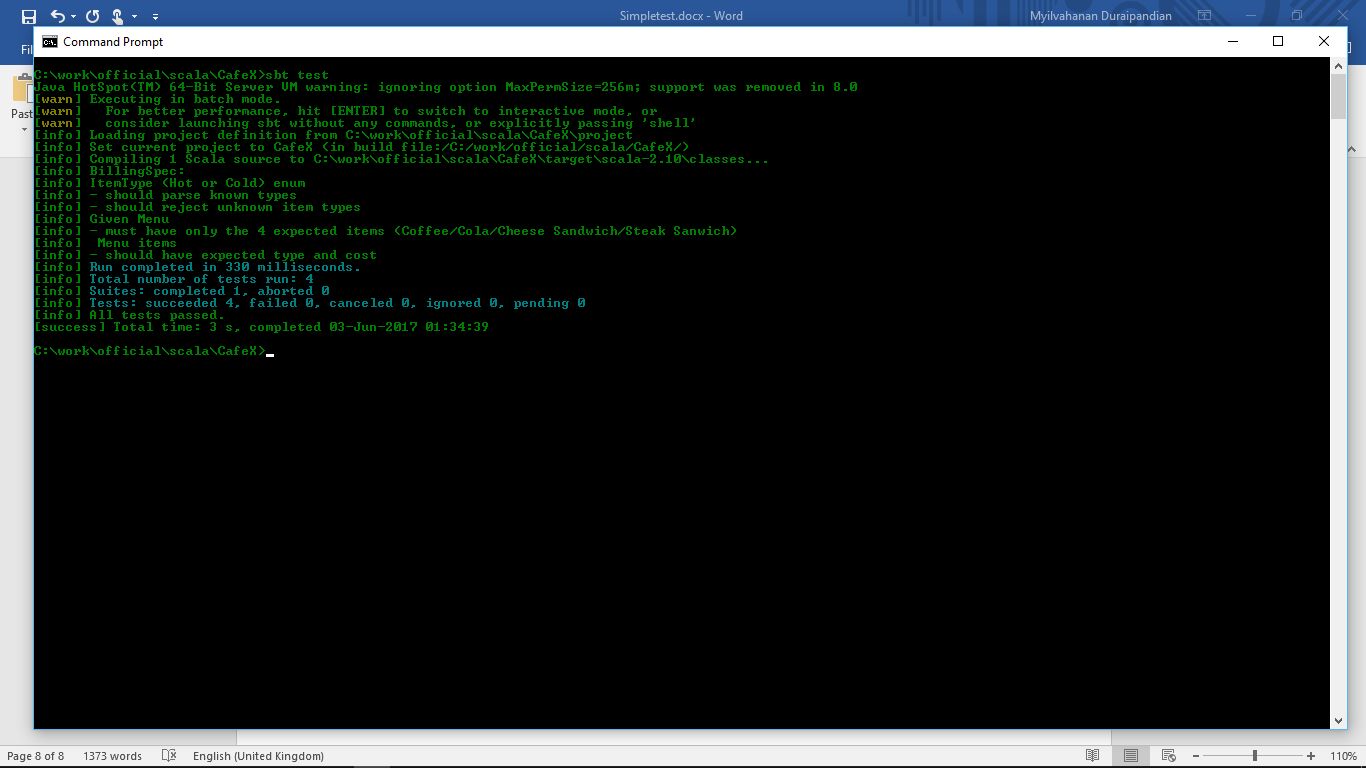
# Checkin 1:

Tests written were to check the content of the menu. The fundamental test was to check if it has the menu items with the cost and type of items. The basic menu components were implemented so as to compile the tests. Having not implemented the Menu, it has failed below in SBT test.



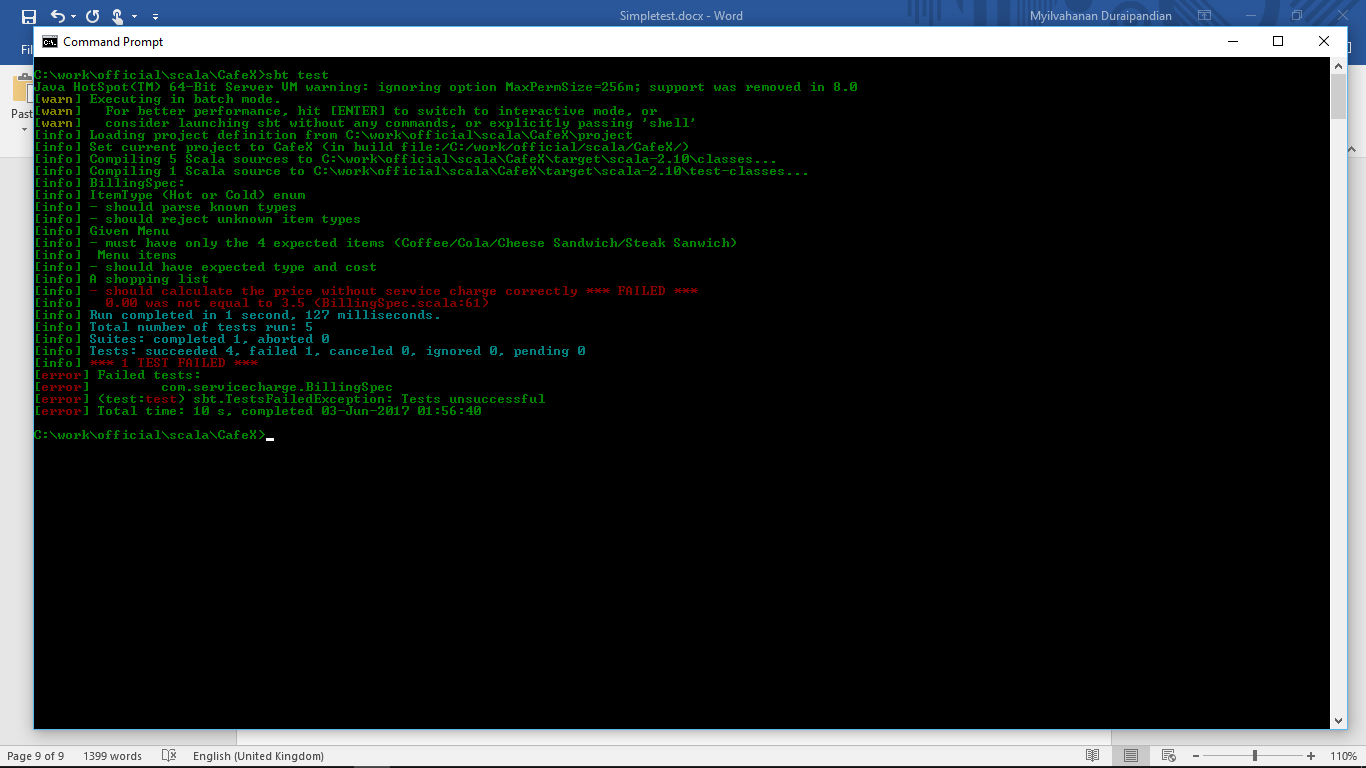
# Checkin 2:

As Menu object was implemented to the requirements provided, the below is the result of test where all tests have succeeded. As the menu has succeeded, its time now for the development of billing object for the step 1 to be implemented.



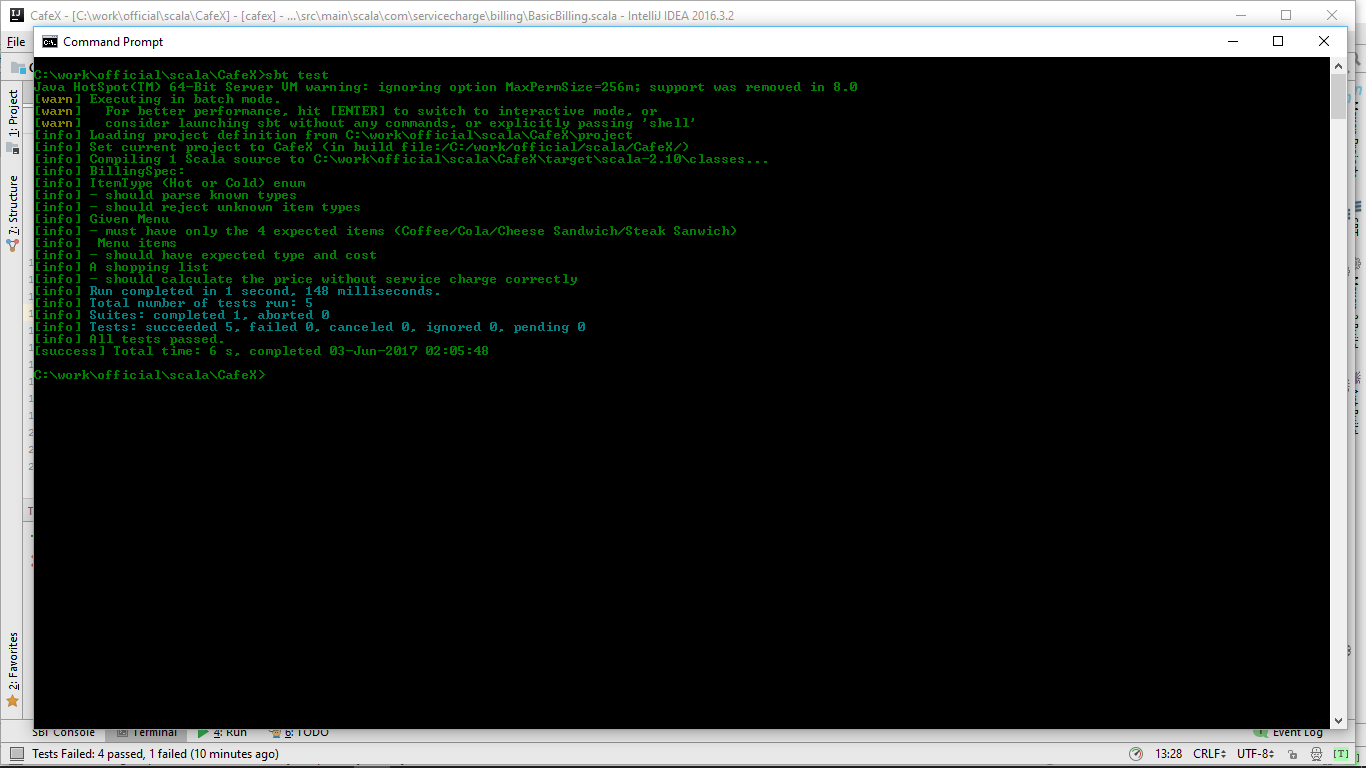
# Checkin 3 (Step 1):

Basic billing is the business component that does the billing without the service charges. As usual the tests are developed and is left to fail as below. The basic billing object is not implemented and returns 0 for any shopping. The below test failure is because the expected 3.5 is not the value the basic billing object returns. Now its time for the implementation of basic billing object.



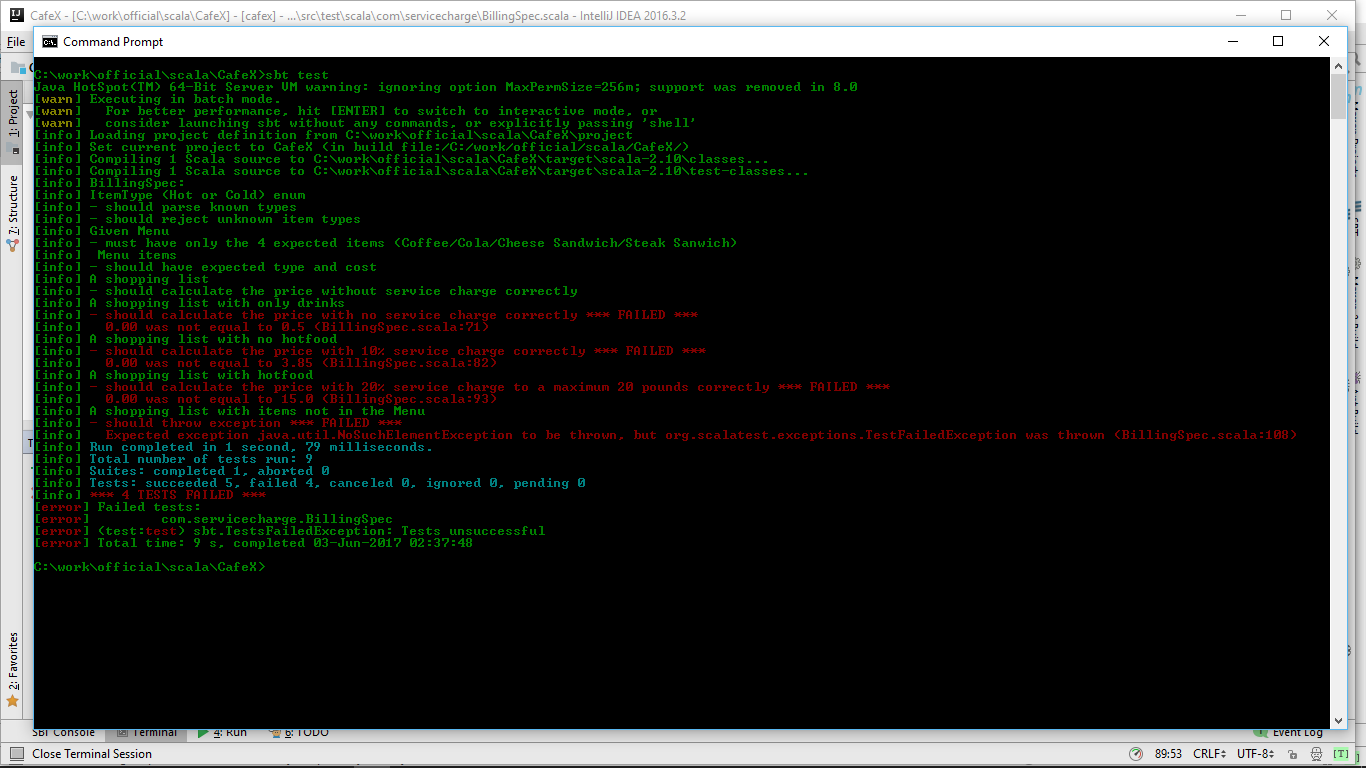
# Checkin 4 (Step 1):

The Basic billing business object is implemented and the test now succeeds.



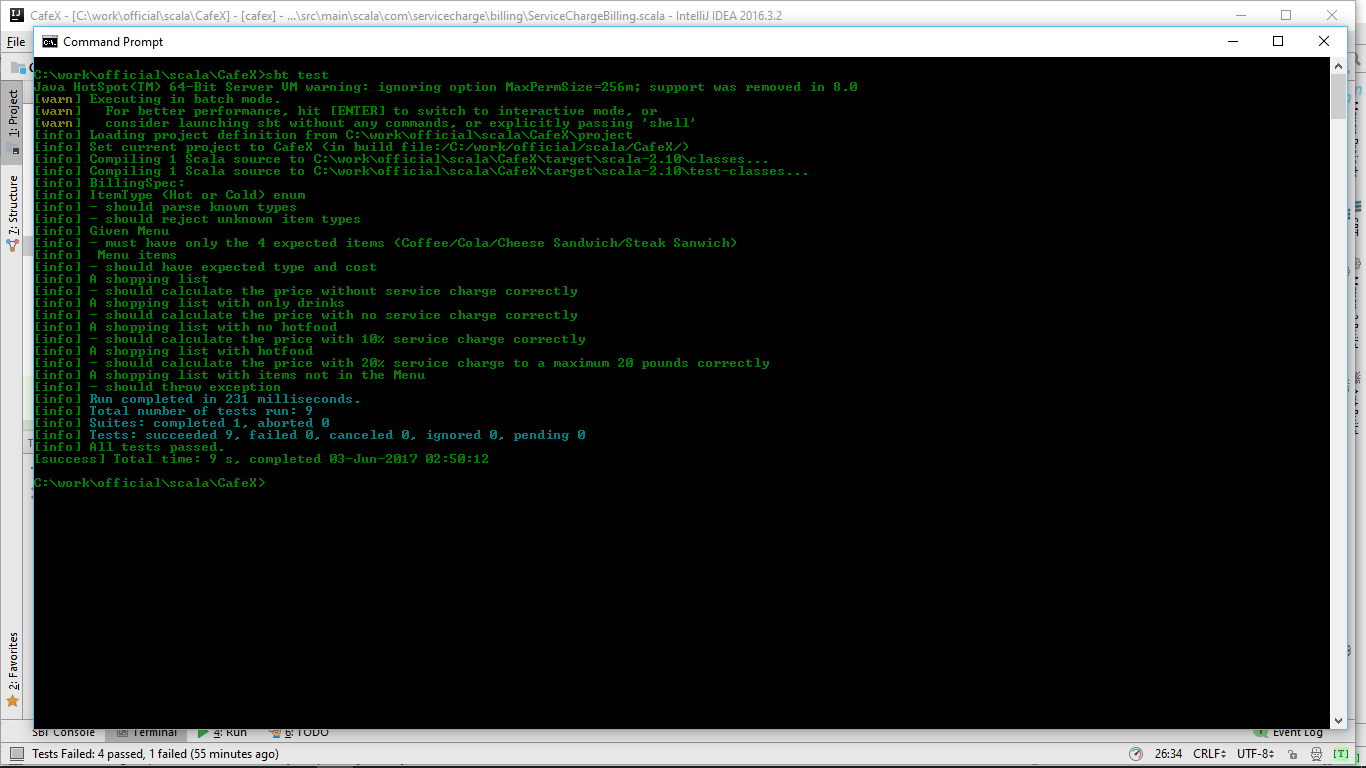
# Checkin 5 (Step 2):

The test case for just liquids, just cold food, with hot food too and shopping list with item not defined in menu are added and its failures are recorded as below.



# Checkin 6 (Step 2):

Having implemented the billing with service charges, all tests that failed in checkin 5 is successful as below.



***Thanks for the opportunity. Looking forward to hear back from you …***