МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федерально автономное бюджетное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационных систем

Куркчи Ариф Эрнестович

Институт информационных технологий и управления в технических системах

курс 3 группа ИС/б-31-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Тестирование программного обеспечения»

на тему «Исследование способов интеграционного тестирования программного обеспечения»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

ст. преподаватель   В. А. Строганов

(должность) (подпись) (инициалы, фамилия)

Севастополь 2016

1. ЦЕЛЬ РАБОТЫ

Исследовать основные принципы интеграционного тестирования программного обеспечения. Приобрести практические навыки организации интеграционных тестов для объектно-ориентированных программ.

1. ПОСТАНОВКА ЗАДАЧИ

Тестируемые классы Stock, Invoice, Bill, CashBox и Consumer должны проходить полный цикл предполагаемой покупки товаров от составления накладной и преобразования её в счёт, оплаты пользователем-«заглушкой» через кассу и получения товара в его карман до возврата товара в магазин по счёту.

1. ХОД РАБОТЫ

Исходный код тестового класса ConsumerTest

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  | | --- | --- | | 1 | package ru.justnero.study.sevsu.pptc; | | 2 |  | | 3 | import org.junit.After; | | 4 | import org.junit.Before; | | 5 | import org.junit.Test; | | 6 |  | | 7 | import ru.justnero.study.sevsu.pptc.util.FakeConsumer; | | 8 |  | | 9 | import java.math.BigInteger; | | 10 | import java.util.Random; | | 11 |  | | 12 | import static *org.hamcrest.CoreMatchers*.\*; | | 13 | import static *org.junit.Assert*.\*; | | 14 |  | | 15 | public class ConsumerTest { | | 16 |  | | 17 | private *FakeConsumer*    consumer; | | 18 | private *Invoice*         invoice; | | 19 | private *String*          name; | | 20 |  | | 21 | *@Before* | | 22 | public *void* setUp() throws *Exception* { | | 23 | name        = new *BigInteger*(130, new *Random*()).toString(32); | | 24 | consumer    = new *FakeConsumer*(name, 1000.0D); | | 25 | *Stock* stock = *Stock*.getInstance(name); | | 26 | stock   .addItem(new *Item*(1,      "Монитор", 149.99D, 20)) | | 27 | .addItem(new *Item*(2,   "Клавиатура",  49.99D, 55)) | | 28 | .addItem(new *Item*(3,         "Мышь",  29.99D, 50)) | | 29 | .addItem(new *Item*(4,       "Корпус",  99.99D, 10)) | | 30 | .addItem(new *Item*(5,    "Процессор", 199.99D,  5)) | | 31 | .addItem(new *Item*(6, "Жёсткий диск", 129.99D, 15)); | | 32 | } | | 33 |  | | 34 | *@Test* | | 35 | public *void* fullCycle() throws *Exception* { | | 36 | invoice = new *Invoice*(); | | 37 | invoice .addItem(new *Item*(1,    "Монитор", 149.99D, 1)) | | 38 | .addItem(new *Item*(2, "Клавиатура",  49.99D, 1)) | | 39 | .addItem(new *Item*(2, "Клавиатура",  49.99D, 1)); | | 40 |  | | 41 | assertThat(invoice.getSum(), is(149.99D + 2.0D \* (49.99D))); | | 42 | assertThat(invoice.items.size(), is(2)); | | 43 |  | | 44 | *CashBox* cashBox = *CashBox*.getInstance(name); | | 45 | *Bill* bill = cashBox.createBill(invoice); | | 46 | cashBox.pay(bill, consumer); | | 47 |  | | 48 | assertThat(consumer.items().size(), is(2)); | | 49 | assertThat(consumer.money(), is(750.03D)); | | 50 |  | | 51 | cashBox.refund(bill, consumer); | | 52 |  | | 53 | assertThat(consumer.money(), is(1000.0D)); | | 54 |  | | 55 | *Report* report = *CashBox*.getInstance(name).createReport(); | | 56 |  | | 57 | report.print(); | | 58 | assertThat(report.getSize(), is(3)); | | 59 | } | | 60 |  | | 61 | *@After* | | 62 | public *void* tearDown() throws *Exception* { | | 63 | *System*.out.println(); | | 64 | } | | 65 |  | | 66 | } | |  | |  |

В тестовом случае составляется накладная на 3 товара, два из которых представляют собой одно и тоже наименование, происходит проверка общей суммы по счёту и группировки однотипных товаров. После этого с помочью кассового аппарата создаётся чек, который оплачивает пользователь-«заглушка». На данном этапе происходит проверка получения им нужного количества товаров и убыли его баланса на сумму покупки. Последний этап теста полного цикла – это возврат товаров по чеку, после чего проверяется, вернулись ли деньги в полном объёме на баланс пользователя. Дополнительно проверяется наполнение данных отчёта по кассовому аппарату, которые и выводятся на экран.

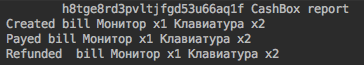


Рисунок 1 – Результат выполнения теста

ВЫВОДЫ

В ходе данной лабораторной работы были изучены подходы к интеграционному тестированию ПО. Был написан интеграционный тест по заданному тестовому случаю к большинству классов проекта по дисциплине «ТСПП». В результате выполнения теста была подтверждена правильность взаимодействия тестируемых классов, а также их способность совершать полный цикл выбора-покупки-возврата.