Quibble, The Queen's Basic Event Ticket Service

Assignment 5: Back Office Unit Testing

Christopher Thomas, 10066835
Nicholas Smith, 10098522

Tuesday, November 24th, 2015

Overview: For our Back Office Unit testing, we have been asked to choose two <u>different</u> methods of testing for each transaction (Sell Transactions and Event Creation Transactions). We have chosen to use the systematic testing method Decision Coverage testing and Basic Block Coverage testing for each transaction respectively. Below we have divided each transaction into two separate parts, with our sell transaction detailing all of our testing methods for it, and then below all the testing methods for our event creation transaction.

<u>Sell Transactions</u>

i) Source Listing

```
1       while(!line.equals("00")) {

            //read in the transaction code
            transCode = Integer.parseInt(line.substring(0, 1));

            //read in the name of the event
            name = line.substring(3,23);
            name = name.trim();


            //read in the date of the event
            date = line.substring(24,30);

            //read in the number of the tickets in the transaction
            numTickets = Integer.parseInt(line.substring(31, 36));

            //sell tickets transaction
2           if (transCode == 1){
                index = names.indexOf(name);
3               if(index != -1){
4                   if(tickets.get(index) - numTickets >= 0)
                        tickets.set(index, tickets.get(index) - numTickets);
                    else
                        System.out.println("Error: event " + name + " cannot oversell its tickets");
                }
                else{
                    System.out.println("Error: event " + name + " does not exist.");
                }
            }
```

ii) Table of Cases to be tested based on Decision Coverage Testing

Decision	Input	Test	Action	Output
1: true	Line == Party	T1	Enters while loop	N/A
1:false	Line = 00	T2	N/A	N/A
2:true	transCode = 01		Enters if statement	
2:false	transCode = 02	T3	Goes to next if statement	N/A
3:true	Index = 0		Enters if statement	N/A
3:false	Index = -1	T4	Goes to else statement	Error: event + name + does not exist.
4:True	Tickets.get(index) - numTickets = 10		Reduces the number	N/A
4:false	Tickets.get(index) – numTickets = -1	T5	Goes to else statement	Error: Event + name + cannot oversell its tickets

iii) Sets of Tests (Transaction Inputs).
To keep all of the information in one PDF, I will be explaining the contents of the .txt files here.
T1 – True

T1currentEvents.txt – Party 00020
T1masterEvents.txt – 20151214 00020 Party
T1mergedDailyTransaction.txt – 01 Party 20151214 00005

T1 - False
T1currentEvents.txt – Party 00020
T1masterEvents.txt – 20151214 00020 Party
T1mergedDailyTransaction.txt – 00

T2 – True
T2currentEvents.txt – Party 00020
T2masterEvents.txt – 20151214 00020 Party
T2mergedDailyTransaction.txt – 01 Party 20151214 00005

T2 – False

T2currentEvents.txt – Party 00020
T2masterEvents.txt – 20151214 00020 Party
T2mergedDailyTransaction.txt – 02 Party 20151214 00005

T3 – True

T3currentEvents.txt – Party 00020
T3masterEvents.txt – 20151214 00020 Party
T3mergedDailyTransaction.txt – 01 Party 20151214 00005

T3 – False

T3currentEvents.txt –
T3masterEvents.txt –
T3mergedDailyTransaction.txt – 01 Party 20151214 00005

T4 – True

T1currentEvents.txt – Party 00020
T1masterEvents.txt – 20151214 00020 Party
T1mergedDailyTransaction.txt – 01 Party5 20151214 00005

This test is quite redundant, as these files would *never* be able to make it to the backend. In order for index = -1, the name would have to not be in the arrayList, and that would not make it past the front end. For the testing sake, we have left it in.

T5 – True

T1currentEvents.txt – Party 00020
T1masterEvents.txt – 20151214 00020 Party
T1mergedDailyTransaction.txt – 01 Party 20151214 00005

T5 – False

T1currentEvents.txt – Party 00020
T1masterEvents.txt – 20151214 00020 Party
T1mergedDailyTransaction.txt – 01 Party 20151214 00021

This is false because the user is trying to buy more tickets than the currentEvents.txt currently has left, making it an error.

iv) Ran these tests through the back end.

v) Table of Failure from the backend.

Test Case	Line of Failure	Reason of Failure	Fixture to Failure

I ran all of the test cases through the back end, and there were *0* failures that occurred.

Test Report for Sell Transaction block of Code

Test Case	Did it fail?	Was the output what you expected?	How did you fix the issue?
Test 1	No	Yes	N/A
Test 2	No	Yes	N/A
Test 3	No	Yes	N/A
Test 4	No	Yes	N/A

There were no failures from the code when tested. Each if-statement was accompanied with an appropriate else statement and each possibility was carefully considered and thought upon. Using XP programming, and having several years' experience with Java, there were no failures when testing the back end.

Event Creation Transactions

i) Source Listing

```
// create event
else if (transCode == 03){
    if(!names.contains(name)){
        names.add(name);
        dates.add(date);
        tickets.add(numTickets);
    }
    else{
        System.out.println("Error: event with the name " + name + " already exists.");
    }
}
```

1 2 3 4

BASIC BLOCK COVERAGE TESTING TABLE

Block	Input: masterEvents	Input: dailyTransactions	Output: currentEvents	Output: masterEvents	Output: Standard Error
1	151201_00155_Party_____	03_PartyTwo_____ _____151203_000 10 00	None	None	N/A
2	151201_00155_Party_____	03_PartyTwo_____ _____151203_000 10 00	None	None	N/A
3	151201_00155_Party_____	03_PartyTwo_____ _____151203_000 10 00	Party_____ _____00155 PartyTwo_____ _____00100 END 00000	151201_00155_Party_____ 151202_00100_PartyTwo_____	N/A
4	151201_00155_Party_____	03_Party_____ _____151203_000 10 00	Party_____ _____00155 END 00000	151201_00155_Party_____	Error: event with the name Party already exists.

When we ran the four test cases above to cover all basic blocks in the "create event" transaction, we did not discover any errors (no exceptions were thrown and all output was as expected). We utilized good programming practices and mutual code reviews to ensure our system worked as designed immediately.

Test Report for Event Creation block of Code

Test Case	Did it fail? Unexpected?	Was the output what you expected?	How did you fix the issue?
Test 1	**No**	**Yes**	**N/A**
Test 2	**No**	**Yes**	**N/A**
Test 3	**No**	**Yes**	**N/A**
Test 4	**No**	**Yes**	**N/A**