SE2832 Lab 4: The Traveling Salesman and his pay

Due: April 9, 2014

1 Introduction

Engineering survives on the ability of people in the sales department to sell products. Without this, a company would go out of business. Many salesman are paid on the basis of commission. In this lab, you will help to verify a system which calculates the commission that is to be paid on a monthly basis to a salesman.

2 Lab Objectives

- Construct a series of testNG tests using a combination of equivalence classes and boundary value analysis.
- Execute testNG tests to uncover simple programming mistakes.
- Correct programming mistakes from within a source code module.
- Regression test software to ensure that the fixing of defects has not broken other portions of the software.

3 Prelab

Prior to lab, you should install the testing tool in your environment. This can be done through the Eclipse "Install New Software" option on the help menu. You will want to use http://beust.com/eclipse as the site.

4 Lab Overview

Every company is different in calculating commissions. The rate may vary by the type of item as well as the seller. The rate may also change with the amount that has been sold for the month.

In this lab, you are tasked with verifying the operation of a program which calculates the commission a traveling salesman will receive over the course of a month according to the details given in the tables below. Some items offer high profit margins, and thus, the salesman receives a higher commission. Other items have lower margins.

Table 1 Commission rates by item type.

Item Type	Employee Type	Commission rate
Basic Sales Item	Probationary	2%
	Experienced	4%
Maintenance Contract	Probationary	3%
	Experienced	6%
Replacement Parts	Probationary	1%
	Experienced	1.5%
Consulting Services	Probationary	3%
	Experienced	8%

Table 2 Minimum sales required before commissions occur.

Employee Type	Required sales before any commission is paid
Probationary	2000.00
Experienced	5000.00

Table 3 Minimum sales for a bonus commission and the bonus commission rate.

Employee Type	Minimum sales	Bonus commission rate
Probationary	50000.00	.5%
Experienced	100000.00	1.5%

Commission is calculated in the following manner. First, commission is based upon the type of employee, either Probationary or Experienced. Before a commission can be earned, a certain minimum sales amount must be achieved. After this, commission will be paid on the amount exceeding this basic amount for subsequent sales. If the employee exceeds a second threshold, then they will receive a bonus commission on sales above the threshold in addition to their regular commissions.

5 Examples

Jim is a, experienced traveling salesman. In the month of August, he had the following sales in order.

Order	Item Type	Amount	Commission
1	Replacement Parts	$1500.00	$0
2	Replacement Parts	$2500.00	$0
3	Replacement Parts	$2000.00	$15
4	Consulting services	$10000.00	$800
5	Maintenance Contract	$25000.00	$1500
6	Basic Sales Item	$9000.00	$360
7	Maintenance Contract	$40000.00	$2400
8	Replacement Parts	$10000.00	$150
9	Basic Sales Item	$1000.00	$55
10	Consulting Services	$5000.00	$475
Total			$5755

The commission was calculated based on the rates shown for an experienced employee.

Steve is a probationary employee who isn't very good at sales. His overall sales are as follows:

Order	Item Type	Amount	Commission
1	Replacement Parts	$100.00	$0
2	Replacement Parts	$250.00	$0
3	Replacement Parts	$200.00	$0
4	Consulting services	$1000.00	$0
5	Maintenance Contract	$250.00	$0
6	Basic Sales Item	$200.00	$0
7	Maintenance Contract	$400.00	$12
Total			$12.00

6 Lab Assignment

Working with a partner, and given this information, you are to devise a complete set of testNG tests using Equivalence testing and / or boundary value analysis to exercise the implementation of the commission. Source code for the class is provided on the course website and JavaDoc is also available.

In your testing, you will uncover a few bugs which will result in test failures. Your job is log the presence of the bug (what failed and how), as well as fix the bug. (Most will be one line changes.)
To do this, comment out the line that was fixed with your initials and add a line detailing the problem as well as the correction. This is shown in Figure 1 and 2.

```
public int sum(int x, int y)
{
        return x-y;
}
```

Figure 1: Example Initial code with bug.

```
public int sum(int x, int y)
{
        return x+y;
        // WWS return x-y;
        // WWS Corrected incorrect return value.
        // WWS Return was calculating the difference of x and y (x-y).
        // WWS Return value should be the sum of x and y (x+y).

}
```

Figure 2: Example Initial code with fixed bug.

When all is done, you should have a completely functioning commission calculator and a complete set of testNG test cases to use. Your testing test cases should be located in a parallel development tree parallel to the source directory.

7 Process Hints

Since you are allowed to work with a partner, it is advised that you divide and conquer the work, focusing first on the functions that are most important to the operation of this calculator.

Since many of the calculations use doubles, it is safe to assume that the answer will only be accurate to the nearest cent. Thus, your testNG test cases should take this into account by providing an appropriate definition for the calculation accuracy. (For other doubles, you may also assume that they are accurate to four decimal places.)

It may also be advisable to use an external tool, such as Excel or Matlab, to aid in test case generation.

If you work in parallel fixing bugs, you may want to download a difference program, such as Araxis Merge or Beyond Compare, which will allow you to see the differences between your code baselines. (Both of these tools have free evaluation versions available.)

8 Bug Reports

As with last lab, you will be working with the Mantis bug tracking system, available at http://emerald.msoe.edu/mantis/. You should file the bug report in the appropriate project SE2832SchillingSalesman2014.

9 Deliverables

1. Source code with bugs corrected.
    a. When correcting bug, provide a comment with your initials indicating where the fixes have been made.
2. JUnit test cases.
    a. Upload your JUnit test cases. Test cases should have a JavaDoc header indicating the author and purpose for testing, and should run with the JUnit environment.
3. Lab report (submitted in pdf format) with the following
    a. Introduction
        i. What are you trying to accomplish with this lab? This section shall be written IN YOUR OWN WORDS. DO NOT copy directly from the assignment.
    b. Strategy
        i. How did you go about determining your test cases?
        ii. How did you organize your test cases?
        iii. Why did you choose the test cases and how did you approach organizing them in a logical fashion.
    c. Bug Reports
        i. List, in tabular format, the bug reports submitted by number, high level description, and submitter.
    d. Things gone right / Things gone wrong
        i. This section shall discuss the things which went correctly with this experiment as well as the things which posed problems during this lab.
    e. Conclusions
        i. What have you learned with this experience?
        ii. How has this lab experience changed your attitude toward testing and when you should work on testing your projects?

If you have any questions, consult your instructor.