Homework Assignment 2

Matthew O'Brien ID: 965010701 Email: matthewo@pdx.edu

Late Days: Using one for this assignment.

Submission Date : May 5th, 2015

Colloboration: Andrew McCann

Solutions Chapter 7.

Problem 17.

Convert the comments of figure 7.23 to Java or C++. Using Java.

```
import java.io.*
import java.util.*

class JobRecordClass{
public int jobNumber;
public JobRecordClass next;
public JobRecordClass previous;
}

class JobQueueClass{
private JobRecordClass front;
private JobRecordClass rear;

        public JobRecordClass getFront(){
                return front;
        }

        public JobRecordClass getRear(){
                return rear:
        }

        public void setFront(JobRecordClass newJob){
                front = newJob;
        }

        public void setRear(JobRecordClass newJob){
                rear = newJob;
        }

        public void initializeJobQueue(){
```

```
            setFront(null);
            setRear(null);
        }

        void addJobToQueue(int jobNumber){
            JobRecordClass newJob = new JobRecordClass();
            newJob.JobNumber = jobNumber;
            newJob.previous = rear;
            newJob.next = null;
            if(getFront() == null)
                setFront(newJob)
            else{
                getRear().next = newJob;
                setRear(newJob);
        }

        public void removeJobFromQueue(){
            setFront(getFront().next);
            if(getFront() == null){
                setRear(null);
            }
            else{
                getFront().previous = null;
            }
        }

        public void printJobQueue(){
            JobRecordClass temp;
            temp = getFront();
            System.out.println("Queue contains:");
            while(temp != null){
                System.out.println(temp.jobNumber);
                temp = temp.next;
        }
}
```

Solutions Chapter 12.

Problem 8:

Assume that only transactions allowed are checks and deposits.

Sort the checks in order by their check numbers. Check each number for a corresponding entry in the bank statement. If not, there is an error with the statement. Match each number to the corresponding statement, and check for equal amounts. If not, there is an error with the statement.

Sort the deposits by the date of deposit. Check the date of the deposit against the bank statements, if not there is an error with the statements. Check the matching dates with the matching statements to ensure equal amounts, if not there is an error with the statement.

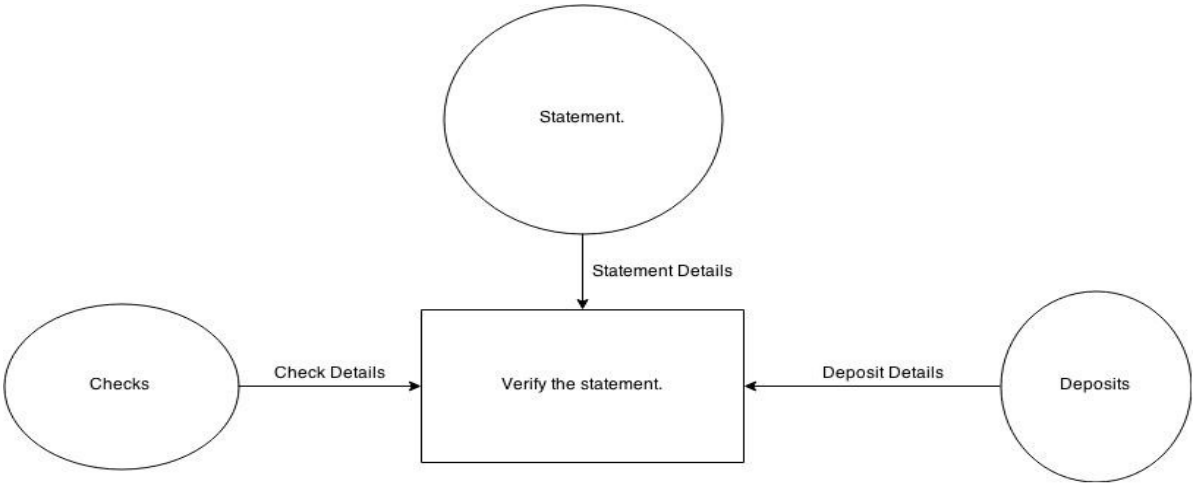
After all of the deposits and checks are processed, check to make sure all of the statements have

a corresponding deposit or check, if not there is an error.

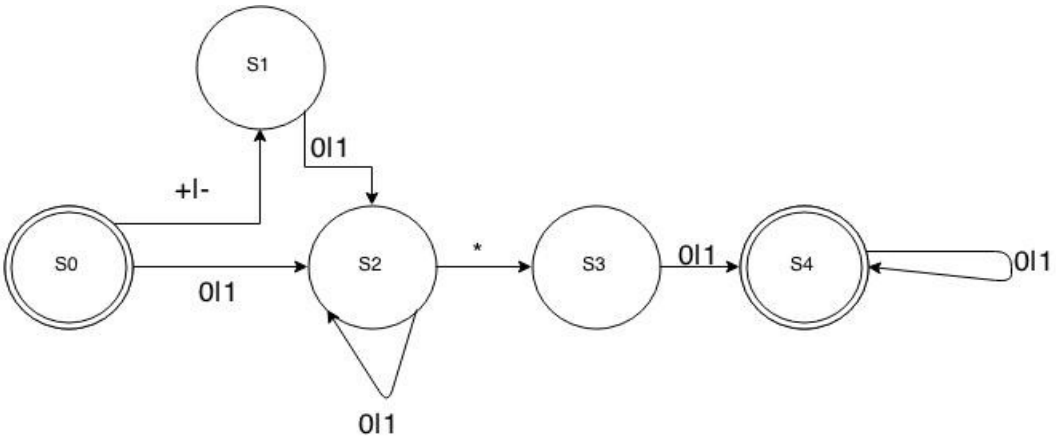
Final check add all of the deposits up and subtract the added total of checks. The remaining balance should be equal to the balance of the account. If not there is an error with the statements.

Problem 9:

Draw a data flow diagram for problem 8.



Problem 13:



Q = {S0, S1, S2, S3, S4}

$\sum$ ❑ = {+, -, *, 0, 1}

q_0 = S0

F = {S4}

Chapter 13

Problem 20

1. Read the bank statement.

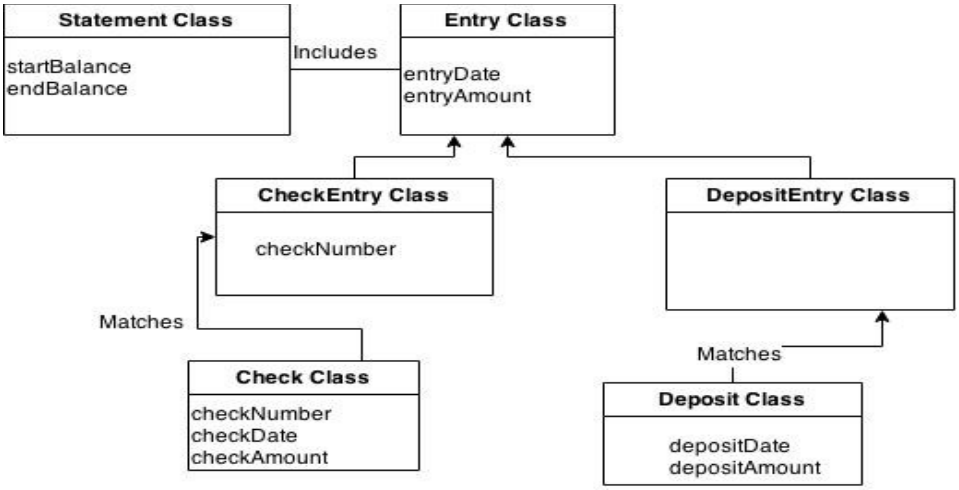
2. Read and sort the checks by number. Each check has a mark for if the bank statement matches the check number and the amount. If not, no mark.
3. Sort the checks by the date on the check. Mark the entry if there is matching date with the statements.
4. All have been marked
5. Add the amount deposit to balance minus the sum of the amount in checks. Resulting amount should match the available balance.

A. At least one check does not match, one bank statement does not match the check number.
B. At least one check does not match. One bank statement does not match the amount of the check.
C. At least one deposit slip. The entry does not match the date of the statement.
D. At least one deposit slip. The amount does not match the amount on the statement.
E. All processed. One or more of the statements remain unmarked.
F. The final total of the two sums subtracted does not match the balance of the account.

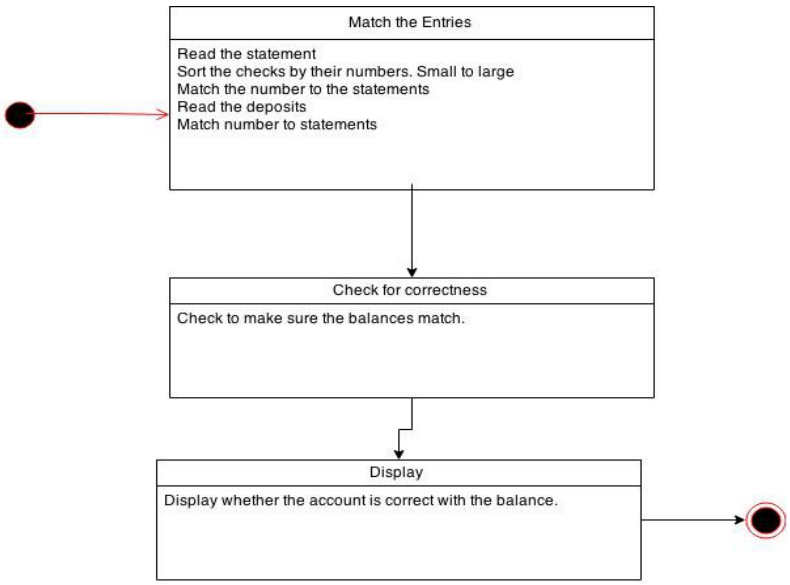
Description: The purpose of this is to determine if the information of an account is accurate. It uses the information of the bank statements in dates and amount. The information of the checks in number, the amount, and date. The information of the deposits in date and amount. All of this information is then used to check for a match with the balance of the account.

Nouns: Product, Statement, Date, entry, balance, number, deposit, check, and amount.

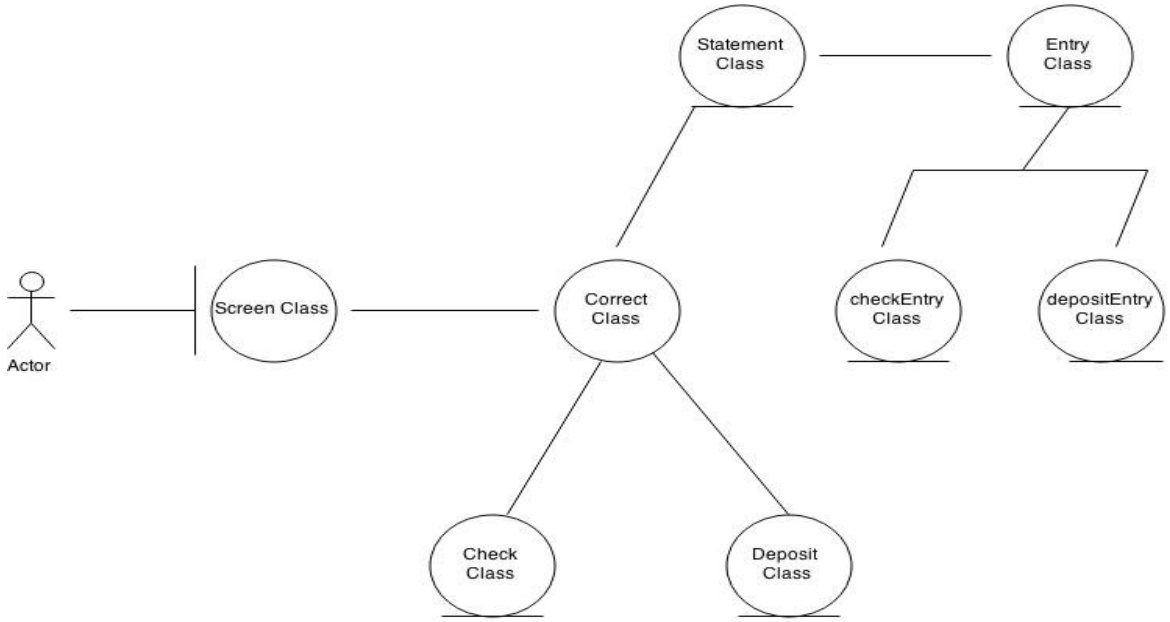
The controlling class for this project is the correctClass to make sure the account, checks, and deposits are accurate.



Checking to make sure the checks, deposits, and balance are the same. Class diagram.



Statechart for the project.

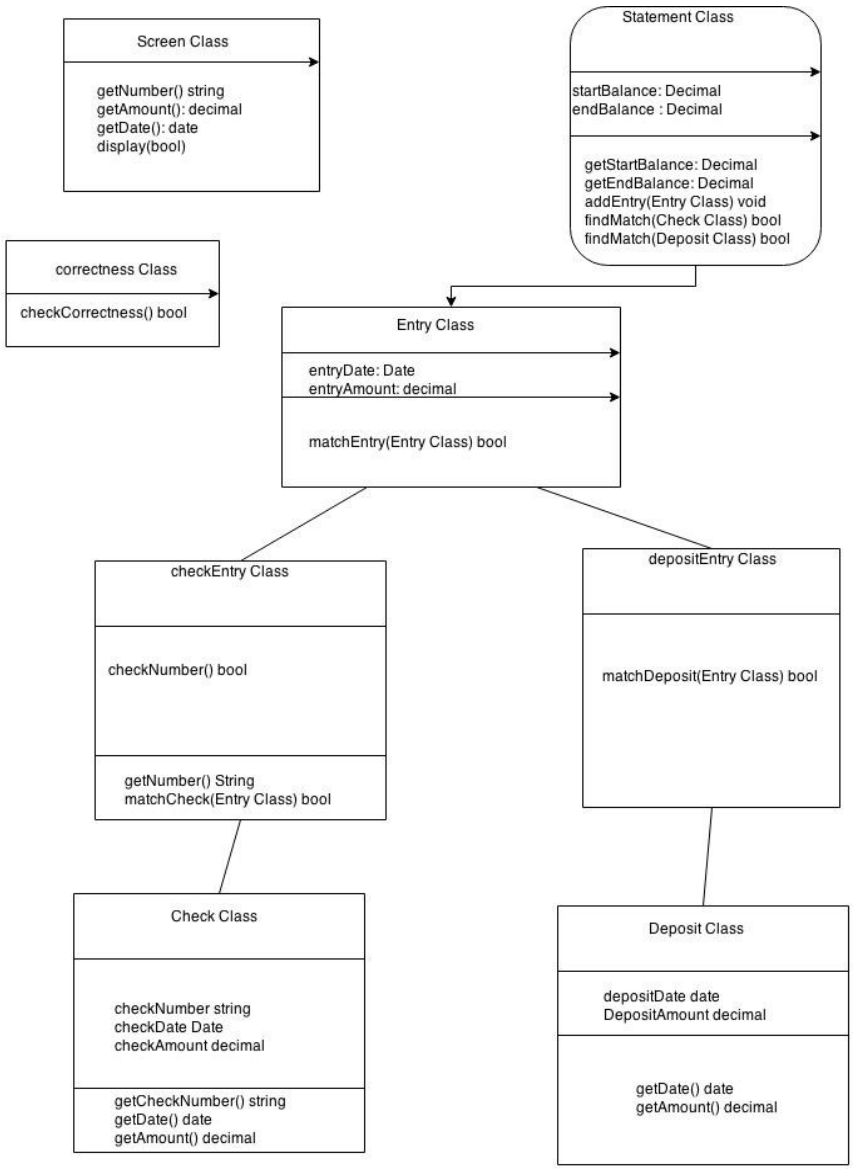


The “actor”, since I couldn't figure out how to change it's name, will have all their information of deposits, checks, and balances. Initially they enter the screen class as an interface with the user. The correct class will use the other classes in order to determine if the balance is correct.

Refer to next question for a more detailed class diagram.

Chapter 14

Problem 14




```
Boolean checkCorrectness(void){
        if(no match)
                return false;
        else{
                //mark the entry with match
                //check the amount
                if(checkAmount doesn't match)
                        return false;
        //get the deposit slips
        //sort the slips by date
        //if matching statement to slip with date and amount
        if(no match found)
```

```
                return false

        else{
                //mark entry
                //verify amount and date.
                if(deposit doesn't match)
                        return false;
        if(all statements aren't marked)
                return false;

        if(statement.startBalance + sum(depositAmount) – sum(checkAmount) ==
statement.endBalance)
                return true;
        else
                return false;
}
```