Lab 5

a. Given is the following bank application:

AccountDAO

saveAccount()
updateAccount()
loadAccount()
getAccounts()

AccountService

createAccount()
deposit()
withdraw()
transferFunds()
getAccount()
getAllAccounts()

Application

main()

Customer

name

Account

accountNumber

deposit()
withdraw()
getBalance()
transferFunds()

AccountEntry

date
amount
description
fromAccountNumber
fromPersonName

: Application   : AccountService   : Account   : AccountDAO

1 : deposit(long accountNumber, double amount)

2 : loadAccount(long accountnumber)

3 : deposit(double amount)

4 : updateAccount(Account account)
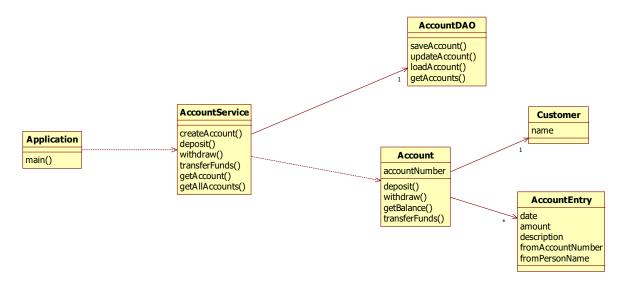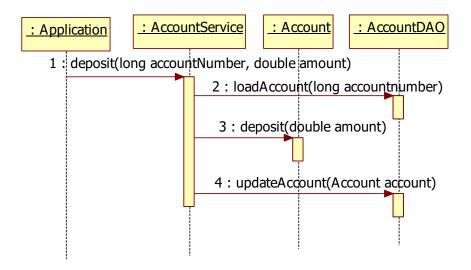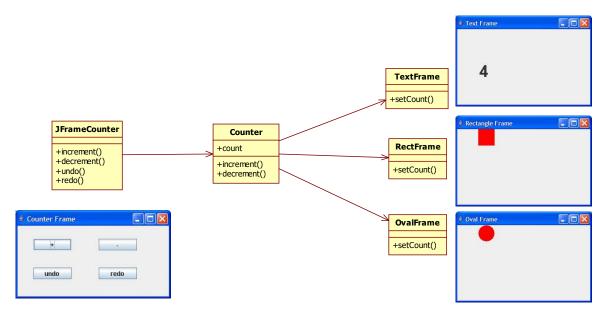
Now we want undo/redo functionality for the methods deposit(), withdraw() and transferFunds()
Draw the modified class diagram.

b. Draw a sequence diagram that shows how your new design works. On the sequence diagram show the following scenario:
   1. First deposit a certain amount
   2. Then withdraw a certain amount
   3. Then call undo

c. Implement the command pattern in the given code.

d. In lab 3 we applied the observer pattern to the following application:

Now we want the undo/redo buttons to work.

Draw the class diagram so that the undo/redo buttons work correctly. Your class diagram needs to show both the observer pattern of lab 3 and the command pattern in one diagram.

e. Draw the sequence diagram that shows the following scenario:
   a. The user clicks the increment button
   b. The user clicks the decrement button
   c. The user clicks undo

f. Implement your new design in Java. Your starting code should be the solution of lab 3. So the code should also implement the observer pattern.