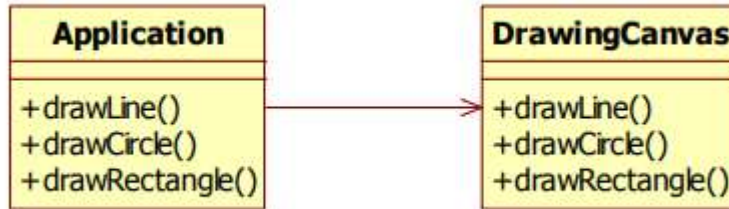


Lab 2

a.

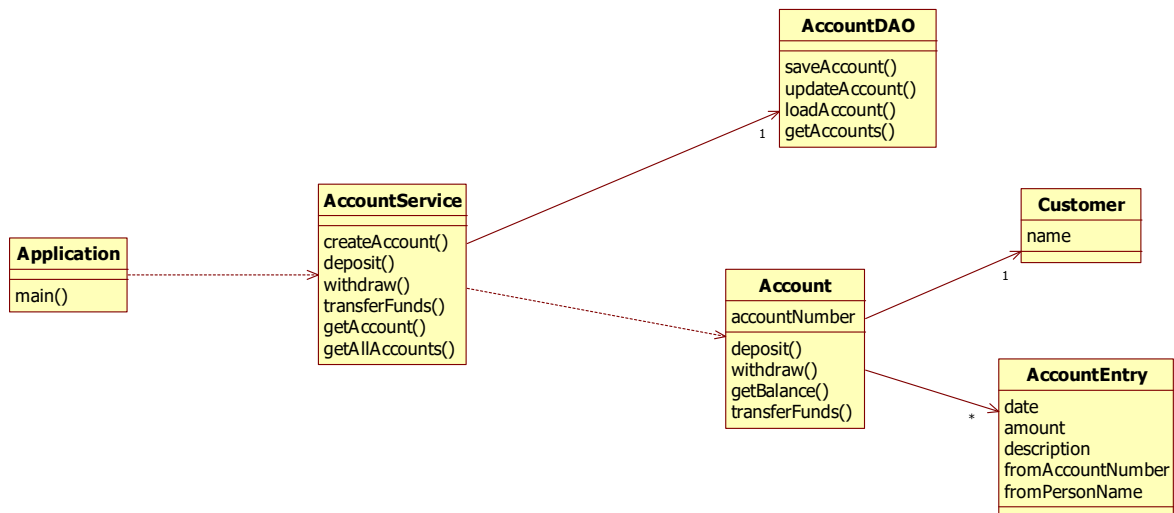
Suppose we have the following simple drawing program:

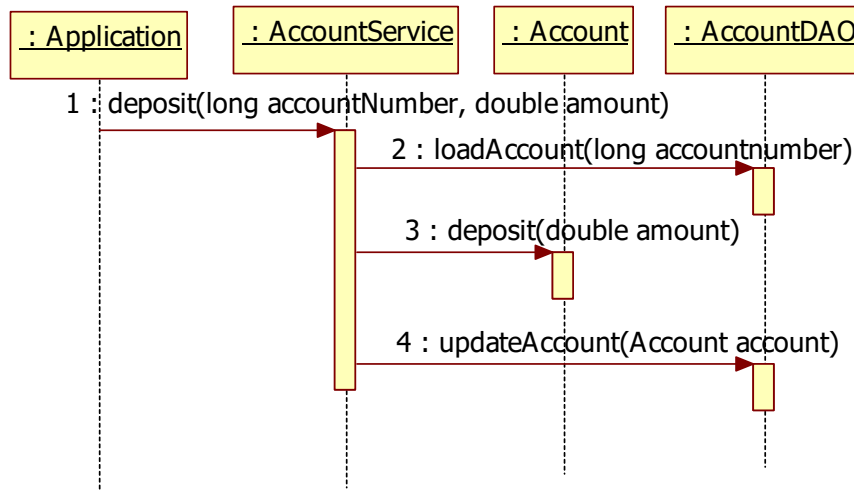


If you call `drawLine()` from the **Application** class, then the `drawLine()` of the **DrawingCanvas** gets called, and a line is drawn on the canvas. A disadvantage of the given drawing application is that it only draws lines, circles and rectangles, and if we also want to draw triangles, we have to change the code of the **DrawingCanvas** class. Redraw the class diagram such that it will be easier to add new shapes to the drawing program. Also draw a sequence diagram that shows how we can draw for example a line and 2 circles.

b.

Given is the following bank application:





Add functionality such that this bank supports Savings and Checkings accounts. For all accounts we want to add interest, and the interest is calculated with the algorithms given below. Use the strategy pattern.

Interest for a savings account:

- If balance < 1000 then you get 1% interest
- If balance > 1000 and balance < 5000 then you get 2% interest
- If balance > 5000 then you get 4% interest

Interest for a checkings account:

- If balance < 1000 then you get 1,5% interest
- If balance > 1000 then you get 2,5% interest

Draw the modified class diagram with the strategy pattern applied.

c.

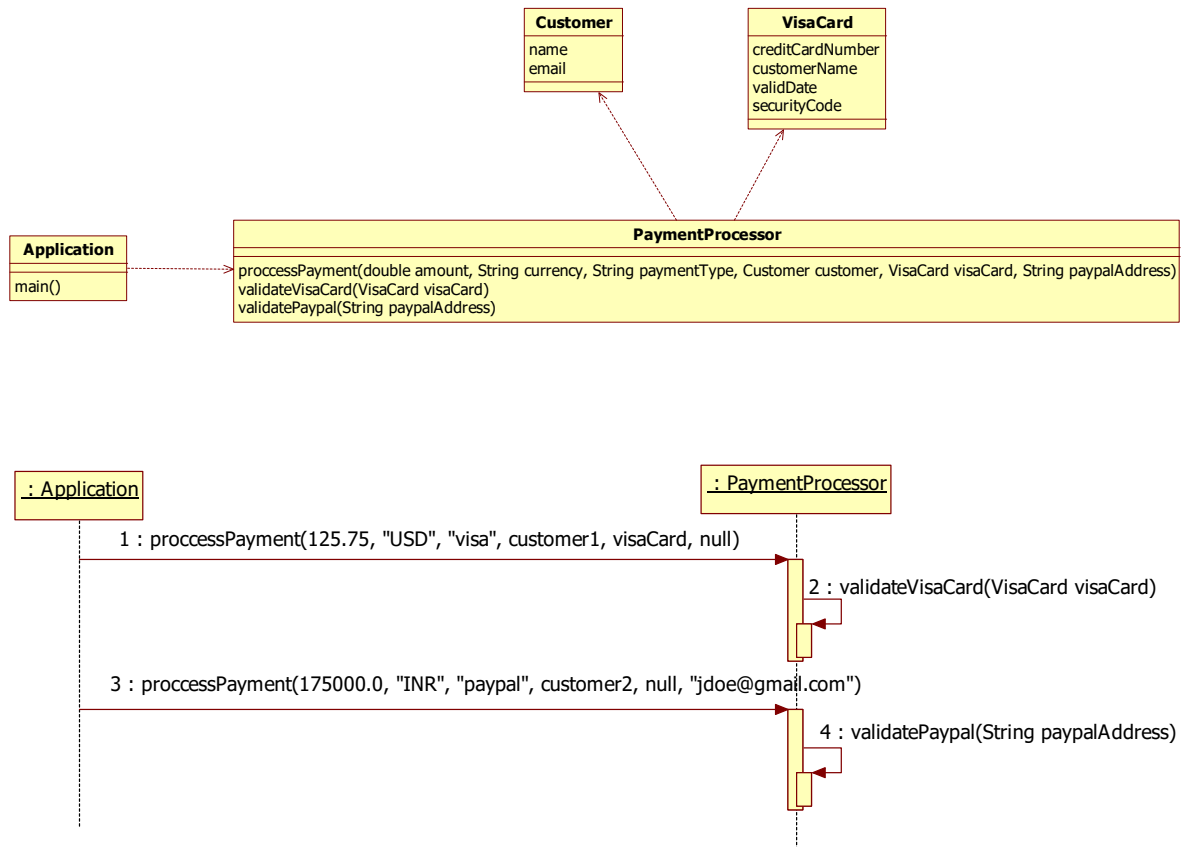
Draw a sequence diagram that shows how your new design works. On the sequence diagram you should show how the strategy pattern works. The AccountService should have a method addInterest(), and then the interest should be calculated and added to all accounts.

d.

Implement the addInterest() functionality using the strategy pattern in the given code.

e.

Given is the following application:



With the `PaymentProcessor` you can process payments. The given implementation supports both Visa Card payment and PayPal payments. The problem with the given implementation is the `processPayment` method is large and complex which makes it hard to add support for other payment methods like MasterCard payments.

Draw the class diagram of a better design using the template method pattern.

Draw the corresponding sequence diagram

f.

Implement the template method in the given Payment processing application so that it is easy to add support for new payment types. The output of your application should be similar as the output of the given application.