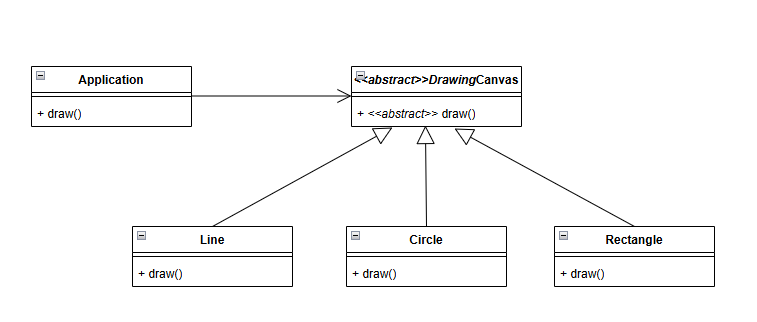
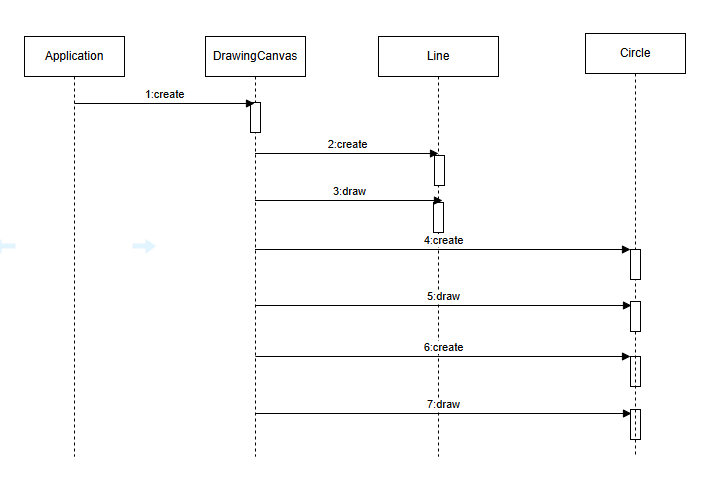
|  |  |
| --- | --- |
| Course | Advanced Software Design – CS525 |
| Assignment | Lab 2 |
| Week | 02 |
| Due | 7/06/2025 |
| Student name | Toe Toe Aung |
| Student ID | 618090 |

1. I redraw the class diagram such that it will be easier to add new shapes to the drawing program. If we also want to draw triangles, we don't have to change the DrawingCanvas class. We just add one more class for triangle.





b)

A diagram of a computer

AI-generated content may be incorrect.

c)

A diagram of a diagram

AI-generated content may be incorrect.

d)

package bank.service;

import java.util.ArrayList;

import java.util.Collection;

import java.util.List;

import bank.dao.AccountDAO;

import bank.dao.IAccountDAO;

import bank.domain.Account;

import bank.domain.Customer;

import bank.domain.AccountTypes;

public class AccountService implements IAccountService {

private IAccountDAO accountDAO;

private InterestCalculator interestCalculator = new InterestCalculator();

public AccountService(){

accountDAO=new AccountDAO();

}

public Account createAccount(long accountNumber, String customerName) {

Account account = new Account(accountNumber);

Customer customer = new Customer(customerName);

account.setCustomer(customer);

accountDAO.saveAccount(account);

return account;

}

public void deposit(long accountNumber, double amount) {

Account account = accountDAO.loadAccount(accountNumber);

account.deposit(amount);

accountDAO.updateAccount(account);

}

public Account getAccount(long accountNumber) {

Account account = accountDAO.loadAccount(accountNumber);

return account;

}

public Collection<Account> getAllAccounts() {

return accountDAO.getAccounts();

}

public void withdraw(long accountNumber, double amount) {

Account account = accountDAO.loadAccount(accountNumber);

account.withdraw(amount);

accountDAO.updateAccount(account);

}

public void transferFunds(long fromAccountNumber, long toAccountNumber, double amount, String description) {

Account fromAccount = accountDAO.loadAccount(fromAccountNumber);

Account toAccount = accountDAO.loadAccount(toAccountNumber);

fromAccount.transferFunds(toAccount, amount, description);

accountDAO.updateAccount(fromAccount);

accountDAO.updateAccount(toAccount);

}

public void addAccountType(Account account, AccountTypes type) {

account.setType(type);

}

public void addInterest(Account account) {

double balanceWithInterest = interestCalculator.calculateInterest(account.getType(), account.getBalance());

System.out.println(

"Account: " + account.getAccountnumber() +

"\nAccount Holder Name: " + account.getCustomer().getName() +

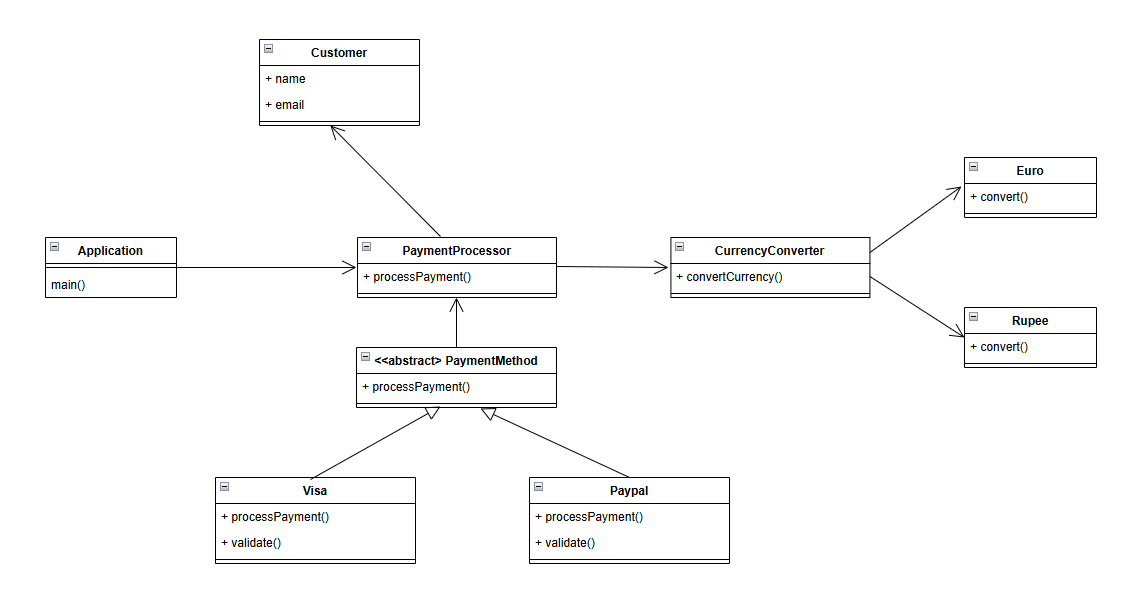
"\nType: " + account.getType() +

"\nBalance with interest: " + balanceWithInterest);

}

}

e) The class diagram of Payment Processor



The sequence diagram of Payment Processor

A diagram of a diagram

AI-generated content may be incorrect.

f) The solution is implemented in the code located in the TemplateMethod folder named with.