

Code:

import java.util.\*;

interface BankInterface {

void process();

}

class BillPayment implements BankInterface {

private Queue<String> paymentQueue = new LinkedList<>();

public void addPayment(String payment) {

paymentQueue.add(payment);

}

@Override

public void process() {

while (!paymentQueue.isEmpty()) {

String payment = paymentQueue.poll();

System.out.println("Processing payment: " + payment);

}

}

}

class NewAccount implements BankInterface {

private Stack<String> accountStack = new Stack<>();

public void addAccount(String account) {

accountStack.push(account);

}

@Override

public void process() {

while (!accountStack.isEmpty()) {

String account = accountStack.pop();

System.out.println("Processing new account: " + account);

}

}

}

class LoanExtension implements BankInterface {

private Set<String> expiredLoans = new HashSet<>();

public void addExpiredLoan(String loan) {

expiredLoans.add(loan);

}

@Override

public void process() {

for (String loan : expiredLoans) {

System.out.println("Contacting loan taker for loan: " + loan);}}}

public class BankMain {

public static void main(String[] args) {

// Create instances of concrete classes

BillPayment billPayment = new BillPayment();

NewAccount newAccount = new NewAccount();

LoanExtension loanExtension = new LoanExtension();

// Add sample data

billPayment.addPayment("Bill 1");

billPayment.addPayment("Bill 2");

newAccount.addAccount("Account 1");

newAccount.addAccount("Account 2");

loanExtension.addExpiredLoan("Loan 1");

loanExtension.addExpiredLoan("Loan 2");

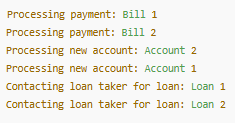
// Process each class

billPayment.process();

newAccount.process();

loanExtension.process();}}

Output:



Git hub: