Khurram Raza 2023F-BSE-006

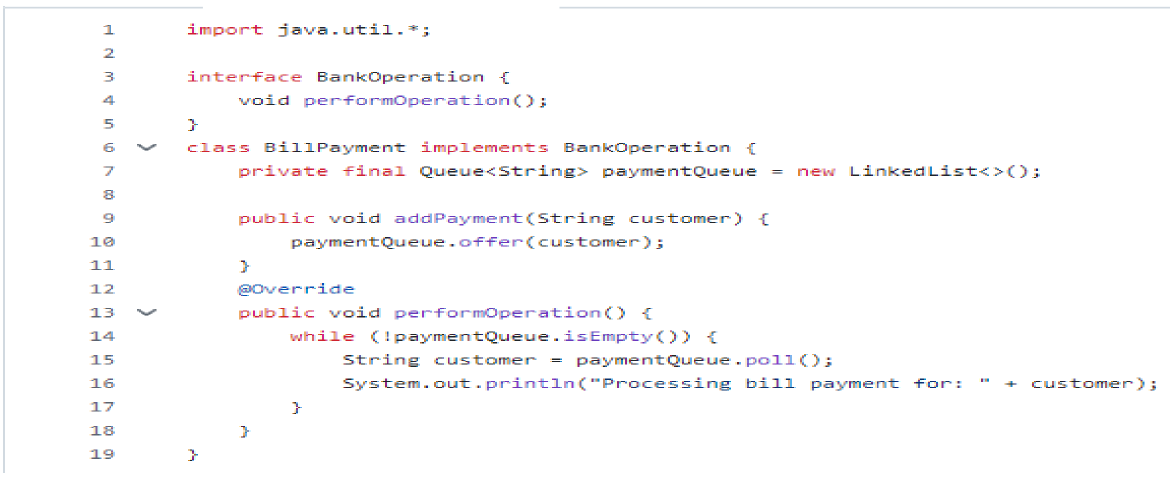
LAB # 13
Abstract Data Types

OBJECTIVE:

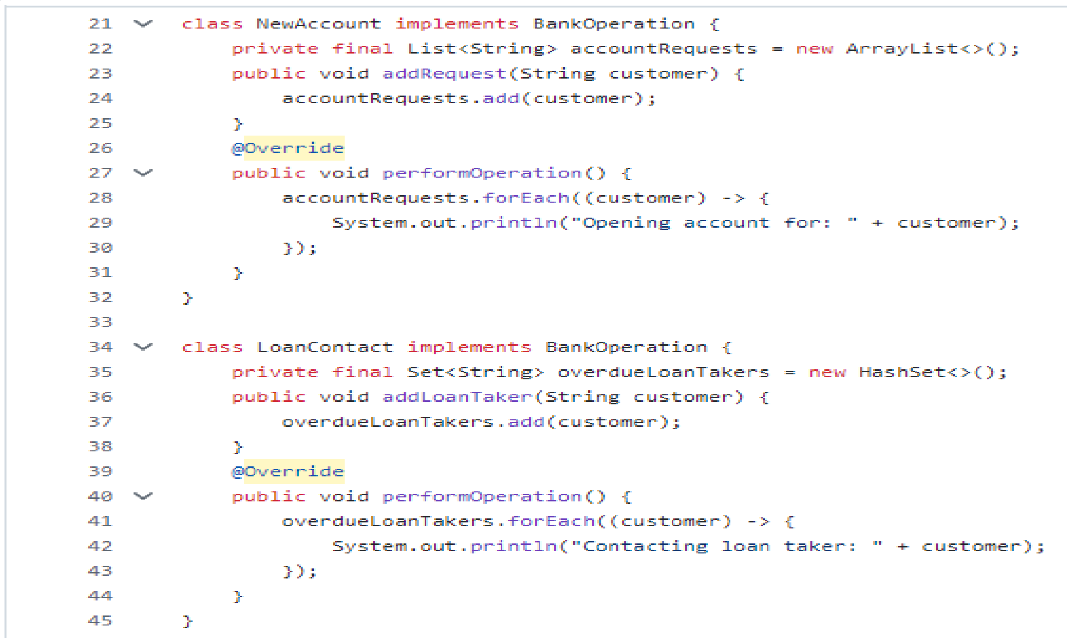
Understanding and implementing abstract data types

LAB TASK:

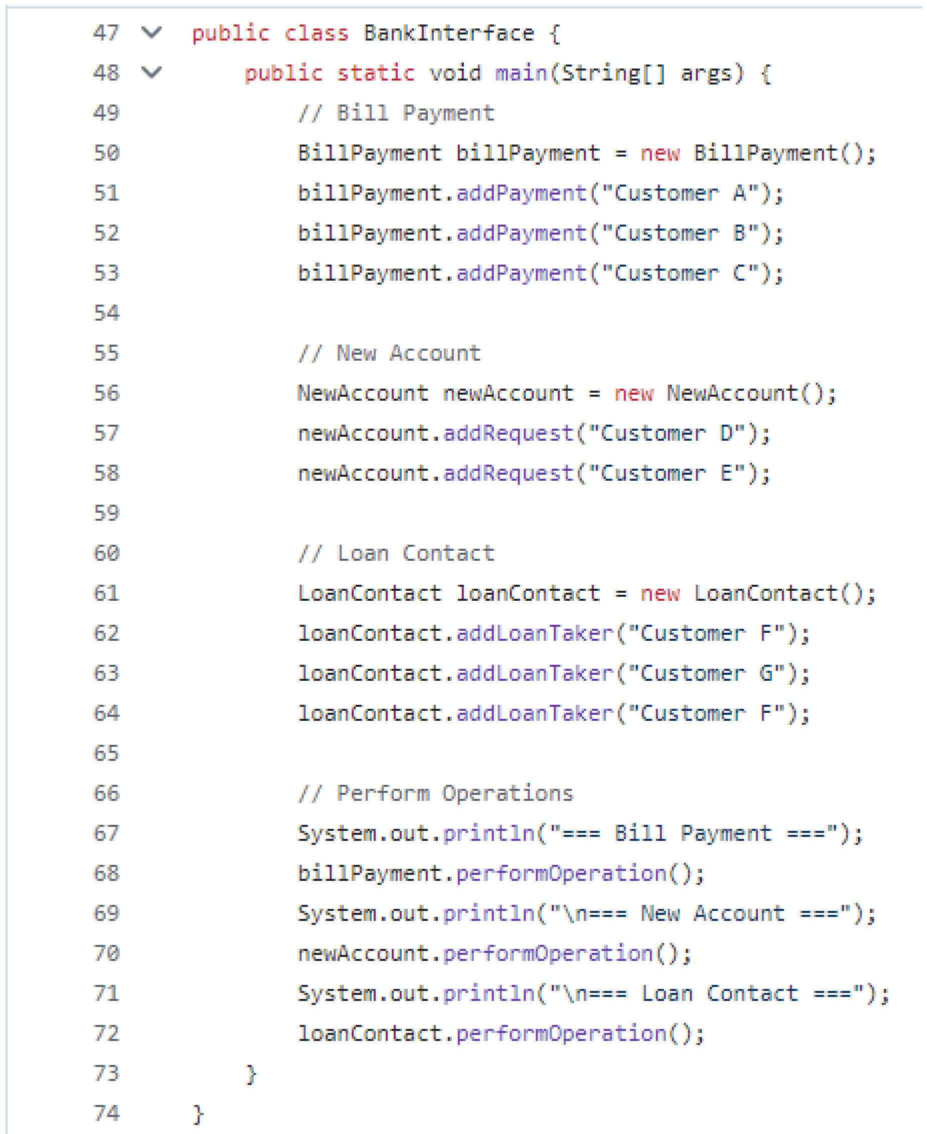
Considering a bank interface, it will have different concrete classes for receiving bill payments, opening of new accounts and contacting those loan takers whose limits of extension have been Expired. Now, develop this scenario on the described ADT and decide which ADT should be used for each concrete class, also implement the whole scenario in multiple java classes.



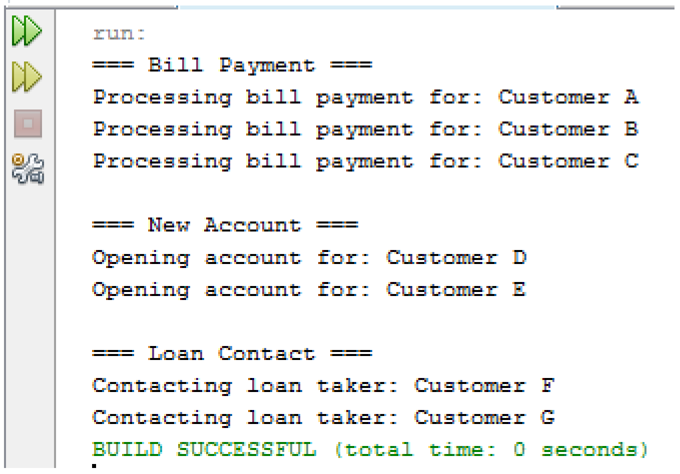
```java
import java.util.*;

interface BankOperation {
    void performOperation();
}
class BillPayment implements BankOperation {
    private final Queue<String> paymentQueue = new LinkedList<>();

    public void addPayment(String customer) {
        paymentQueue.offer(customer);
    }
    @Override
    public void performOperation() {
        while (!paymentQueue.isEmpty()) {
            String customer = paymentQueue.poll();
            System.out.println("Processing bill payment for: " + customer);
        }
    }
}
```



```java
class NewAccount implements BankOperation {
    private final List<String> accountRequests = new ArrayList<>();
    public void addRequest(String customer) {
        accountRequests.add(customer);
    }
    @Override
    public void performOperation() {
        accountRequests.forEach((customer) -> {
            System.out.println("Opening account for: " + customer);
        });
    }
}

class LoanContact implements BankOperation {
    private final Set<String> overdueLoanTakers = new HashSet<>();
    public void addLoanTaker(String customer) {
        overdueLoanTakers.add(customer);
    }
    @Override
    public void performOperation() {
        overdueLoanTakers.forEach((customer) -> {
            System.out.println("Contacting loan taker: " + customer);
        });
    }
}
```



```java
public class BankInterface {
    public static void main(String[] args) {
        // Bill Payment
        BillPayment billPayment = new BillPayment();
        billPayment.addPayment("Customer A");
        billPayment.addPayment("Customer B");
        billPayment.addPayment("Customer C");

        // New Account
        NewAccount newAccount = new NewAccount();
        newAccount.addRequest("Customer D");
        newAccount.addRequest("Customer E");

        // Loan Contact
        LoanContact loanContact = new LoanContact();
        loanContact.addLoanTaker("Customer F");
        loanContact.addLoanTaker("Customer G");
        loanContact.addLoanTaker("Customer F");

        // Perform Operations
        System.out.println("=== Bill Payment ===");
        billPayment.performOperation();
        System.out.println("\n=== New Account ===");
        newAccount.performOperation();
        System.out.println("\n=== Loan Contact ===");
        loanContact.performOperation();
    }
}
```



```
run:
=== Bill Payment ===
Processing bill payment for: Customer A
Processing bill payment for: Customer B
Processing bill payment for: Customer C

=== New Account ===
Opening account for: Customer D
Opening account for: Customer E

=== Loan Contact ===
Contacting loan taker: Customer F
Contacting loan taker: Customer G
BUILD SUCCESSFUL (total time: 0 seconds)
```