

```

package Debugger;

/**
 * Author: Yibo Guo (yguo25@ucsc.edu)
 * Date: Jun 5, 2018
 *
 * What I did: I did all following functions for testing purpose except
 * placeOrder() and cancelOrder().
 * Although Ho did giveRate() and acceptRate(), but I rewrite them due to
 * server performance improvement.
 */

import java.sql.SQLException;
import java.sql.Timestamp;
import Object.*;

public class Tester {

    static String sessionID = "";

    public static void main (String args[]) throws SQLException {
        register("yoona", "snsd.or.kr", "loveYOONA!");
        login("yoona", "snsd.or.kr", "loveYOONA!");
        placeOrder();
        addAddress();
        loadAddress(getSessionID());
        deleteAddress();
        loadPersonOrder(getSessionID());
        loadPersonSold(getSessionID());
        loadOrder();
        addCard();
        loadCard();
        deleteCard();
        giveRate();
        acceptRate();
        cancelOrder();
        searchOrderByName();
        searchOrderByID();
        changePC();
        checkExt();
        changePC();
        loadOrderDetail();
        loadPartialOrder();
        loadOffer("KR10000000");
    }

    private static void register(String user, String domain, String password)
        throws SQLException {
        LinkedList l = new LinkedList();
        l.insert("reg");
        User u = new User(user, domain, password);
    }

```

```

        l.insert(u);
        System.out.println(IntermediateAPI.API.getCommand(l));
    }

private static void login(String user, String domain, String password)
    throws SQLException {
    LinkedList l = new LinkedList();
    l.insert("lgi");
    User u = new User(user, domain, password);
    l.insert(u);
    String sid = (String)IntermediateAPI.API.getCommand(l);
    if (!(sid.length() == 6 && sid.charAt(0) == '0' && sid.charAt(1) ==
        'x')) setSessionID(sid);
    System.out.println(sid);
}

private static void placeOrder() throws SQLException {

    String res = "";
    LinkedList l = new LinkedList();
    l.insert("plo");
    l.insert(getSessionID());

    for (int i = 0; i < 1; i++) {
        String p, b, c, img;
        String l1, ci, s, z, cd;
        int q;
        Timestamp ts;

        p = "Yoona\\'s Choice";
        b = "Innisfree";
        q = 1;
        c = "KR";
        img = "http://imgstore04.cdn.sogou.com/app/a/
            100520024/877e990117d6a7ebc68f46c5e76fc47a";
        ts = new Timestamp(System.currentTimeMillis());
        Order u = new Order(p, b, q, c, img, ts);
        l1 = "Yeongdong-daero 513";
        ci = "Seoul";
        s = "KR";
        z = "01234";
        cd = "379820083712345";
        UserShippingInfo usi = new UserShippingInfo(l1, ci, s, z, cd);
        InitialOrder io = new InitialOrder(u, usi);
        l.insert(io);

        p = "Zero Balance Cleasing";
        b = "Banila Co.";
        q = 1;
        c = "KR";
    }
}

```

```

        img = "https://images-na.ssl-images-amazon.com/images/I/
        51bwEB1zwTL._SX522_.jpg";
        ts = new Timestamp(System.currentTimeMillis());
        u = new Order(p, b, q, c, img, ts);
        l1 = "Yeongdong-daero 513";
        ci = "Seoul";
        s = "KR";
        z = "01234";
        cd = "379820083712345";
        usi = new UserShippingInfo(l1, ci, s, z, cd);
        io = new InitialOrder(u, usi);
        l.insert(io);

    }

    System.out.println("Finish Insert.");
    res = (String)IntermediateAPI.API.getCommand(l);

    System.out.println(res);
}

private static void addAddress() throws SQLException {

    LinkedList l1 = new LinkedList();
    l1.insert("ada");
    l1.insert(getSessionID());
    String f, l, co, l1, l2, c, s, z;

    f = "Yoona";
    l = "Lim";
    co = "SM Ent\\'l";
    l1 = "Yeongdong-daero 513";
    l2 = "Gangnam-gu";
    c = "Seoul";
    s = "KR";
    z = "06164";
    Address a = new Address(f,l, co, l1, l2, c, s, z);
    l1.insert(a);

    f = "Taeyeon";
    l = "Kim";
    co = "SM Ent\\'l";
    l1 = "Yeongdong-daero 513";
    l2 = "Gangnam-gu";
    c = "Seoul";
    s = "KR";
    z = "06164";
    a = new Address(f,l, co, l1, l2, c, s, z);
    l1.insert(a);

    String res = (String)IntermediateAPI.API.getCommand(l1);

```

```

        System.out.println(res);
    }

    private static void loadAddress(String id) throws SQLException {
        LinkedList ll = new LinkedList();
        ll.insert("lda");
        ll.insert(getSessionID());
        ll = (LinkedList) IntermediateAPI.API.getCommand(ll);
        Node temp = ll.head;
        if (temp != null && temp.getObject().getClass().equals("").getClass()))
        {
            System.out.println(temp.getObject());
            return;
        }
        while (temp != null) {
            Address a = (Address)temp.getObject();
            System.out.println(a.getFN() + " " + a.getLN() + " " + a.getCom()
                + " " + a.getL1()
                + " " + a.getL2() + " " + a.getCity() + " " + a.getState() + " " +
                a.getZip());
            temp = temp.getNext();
        }
    }

    private static void loadPersonSold(String sessionID) throws SQLException {
        LinkedList l = new LinkedList();
        l.insert("lds");
        l.insert(sessionID);

        Object ob = IntermediateAPI.API.getCommand(l);
        if (!ob.getClass().equals((new LinkedList()).getClass())) {
            System.out.println(ob);
            return;
        }

        l = (LinkedList) ob;
        Node temp = l.head;
        while (temp != null) {
            UserOrderHis u = (UserOrderHis) temp.getObject();
            Order o = u.getOrder();
            System.out.println(o.getImage() + " " + o.getBrand() + " " +
                o.getProduct() +
                " " + o.getQuantity() + " " + o.getCountry() + " " +
                o.getTimestamp() + " " + u.getOrderStatus());
            temp = temp.getNext();
        }
    }

    private static void loadPersonOrder(String sessionID) throws SQLException {
        LinkedList l = new LinkedList();
        l.insert("ldl");
    }

```

```

l.insert(sessionID);

Object ob = IntermediateAPI.API.getCommand(l);
if (!ob.getClass().equals((new LinkedList()).getClass())) {
    System.out.println(ob);
    return;
}

l = (LinkedList) ob;
Node temp = l.head;
while (temp != null) {
    UserOrderHis u = (UserOrderHis) temp.getObject();
    Order o = u.getOrder();
    System.out.println(o.getImage() + " " + o.getBrand() + " " +
        o.getProduct() +
        " " + o.getQuantity() + " " + o.getCountry() + " " +
        o.getTimestamp() + " " + u.getOrderStatus());
    temp = temp.getNext();
}

}

private static void loadOrder() throws SQLException {
    LinkedList ll = new LinkedList();
    ll.insert("ldo");
    ll.insert(getSessionID());
    ll.insert("RKR");
    Object obj = IntermediateAPI.API.getCommand(ll);
    LinkedList res = (LinkedList) obj;
    Node temp = res.head;
    if (temp != null && temp.getObject().getClass().equals("").getClass()))
    {
        System.out.println(temp.getObject());
        return;
    }
    while (temp != null) {
        Order o = (Order) temp.getObject();
        System.out.println(o.getImage() + " " + o.getBrand() + " " +
            o.getProduct() +
            " " + o.getQuantity() + " " + o.getCountry() + " " +
            o.getTimestamp());
        temp = temp.getNext();
    }
}

private static void addCard() throws SQLException {
    LinkedList ll = new LinkedList();
    ll.insert("adc");
    ll.insert(getSessionID());

    String f, l, i, c, e, z;
    f = "Yoona";

```

```

l = "Lim";
i = "VISA";
c = "4678901234567893";
z = "00000";
e = "1025";
Card cd = new Card(f, l, i, c, e, z);
ll.insert(cd);

f = "Yoona";
l = "Lim";
i = "AMEX";
c = "379812345678903";
z = "00000";
e = "0123";
cd = new Card(f, l, i, c, e, z);
ll.insert(cd);

String res = (String)IntermediateAPI.API.getCommand(ll);
System.out.println(res);
}

private static void deleteAddress() throws SQLException {
    LinkedList ll = new LinkedList();
    ll.insert("dta");
    ll.insert(getSessionID());
    Address a = new Address("Yoona", "Lim", "SM Ent\\'l", "Yeongdong-daero
        514", "Gangnam-gu", "Seoul", "KR", "00001");
    ll.insert(a);
    String res = (String)IntermediateAPI.API.getCommand(ll);
    System.out.println(res);
}

private static void loadCard() throws SQLException {
    LinkedList ll = new LinkedList();
    ll.insert("ldc");
    ll.insert(getSessionID());

    ll = (LinkedList) IntermediateAPI.API.getCommand(ll);
    Node temp = ll.head;
    while (temp != null) {
        Card c = (Card) temp.getObject();
        System.out.println(c.getFN() + " " + c.getLN() + " " +
            c.getIssuser()
            + " " + c.getCardNum() + " " + c.getExp() + " " + c.getZip());
        temp = temp.getNext();
    }
}

private static void deleteCard() throws SQLException {
    LinkedList ll = new LinkedList();
    ll.insert("dte");

```

```

        ll.insert(getSessionID());
        ll.insert("379812345678901");
        ll.insert("4678901234567890");

        String res = (String)IntermediateAPI.API.getCommand(ll);
        System.out.println(res);
    }

    private static void giveRate() throws SQLException {
        LinkedList ll = new LinkedList();
        ll.insert("gvr");
        ll.insert(getSessionID());

        String OID, SID, RM;
        double RA, EC;
        int SM;
        boolean A;

        OID = "US10000001";
        SID = "snok10000";
        RA = 100.00;
        EC = 10.00;
        SM = 1;
        A = false;
        RM = "Rate Example";

        Offer offer = new Offer(OID, SID, RA, EC, SM, A, RM);
        ll.insert(offer);

        String res = (String)IntermediateAPI.API.getCommand(ll);
        System.out.println(res);
    }

    private static void acceptRate() throws SQLException {
        LinkedList ll = new LinkedList();
        ll.insert("art");
        ll.insert(getSessionID());

        String OID, SID;
        OID = "US10000206";
        SID = "snok10000";
        ll.insert(OID);
        ll.insert(SID);

        String res = (String)IntermediateAPI.API.getCommand(ll);
        System.out.println(res);
    }

    private static void cancelOrder() throws SQLException{
        LinkedList ll = new LinkedList();
        ll.insert("cco");
    }

```

```

        ll.insert(getSessionID());

        String orderID = "US10000004";
        ll.insert(orderID);

        String res = (String)IntermediateAPI.API.getCommand(ll);
        System.out.println(res);
    }

    private static void searchOrderByName() throws SQLException{
        LinkedList ll = new LinkedList();
        ll.insert("spn");
        ll.insert(getSessionID());
        ll.insert("US");
        ll.insert(null);
        ll.insert("%ona\\'s Ch%");
        Object obj = IntermediateAPI.API.getCommand(ll);
        if (obj.getClass().equals("").getClass())) {
            System.out.println((String)obj);
            return;
        }
        ll = (LinkedList) obj;

        Node temp = ll.head;
        while (temp != null) {
            Order o = (Order) temp.getObject();
            System.out.println(o.getImage() + " " + o.getBrand() + " " +
                o.getProduct() +
                " " + o.getQuantity() + " " + o.getCountry() + " " +
                o.getTimestamp());
            temp = temp.getNext();
        }
    }

    private static void searchOrderByID() throws SQLException{
        LinkedList ll = new LinkedList();
        ll.insert("spi");
        ll.insert(getSessionID());
        ll.insert("KR10000000");
        Object obj = IntermediateAPI.API.getCommand(ll);
        if (obj.getClass().equals("").getClass()))
            System.out.println(obj.toString());
        else {
            Order o = (Order) obj;
            System.out.println(o.getImage() + " " + o.getBrand() + " " +
                o.getProduct() +
                " " + o.getQuantity() + " " + o.getCountry() + " " +
                o.getTimestamp());
        }
    }
}

```



```

private static void changePC() throws SQLException {
    LinkedList ll = new LinkedList();
    ll.insert("cgp");
    ll.insert(getSessionID());
    ll.insert("loveYOONA!");
    ll.insert("test12");
    System.out.println( (String) IntermediateAPI.API.getCommand(ll) );
}

private static void checkExt() throws SQLException {
    LinkedList ll = new LinkedList();
    ll.insert("cke");
    ll.insert(getSessionID());
    ll.insert("CAB100000");
    System.out.println( (String) IntermediateAPI.API.getCommand(ll) );
}

private static void loadOrderDetail() throws SQLException {
    LinkedList ll = new LinkedList();
    ll.insert("lod");
    ll.insert(getSessionID());
    ll.insert("KR10000005");
    System.out.println(IntermediateAPI.API.getCommand(ll).getClass());
}

private static void loadPartialOrder() throws SQLException{
    functionTester.run(getSessionID());
}

private static void loadOffer(String str) throws SQLException{
    LinkedList ll = new LinkedList();
    ll.insert("lor");
    ll.insert(getSessionID());
    ll.insert(str);
    Object o = IntermediateAPI.API.getCommand(ll);
    if (o.getClass().equals("").getClass())) System.out.println((String) o);
    ll = (LinkedList) o;
    Node temp = ll.head;
    while (temp != null) {
        Offer of = (Offer) temp.getObject();
        System.out.println(str + " " + of.getSellerID() + " " +
            of.getShippingMethod()
                + " " + of.getRate() + " " + of.getExpressCost() + " "
                + of.getRemark());
        temp = temp.getNext();
    }
}

private static String getSessionID() {
    return sessionID;
}

```

```
private static void setSessionID(String str) {  
    sessionID = str;  
}  
  
}
```