

Java 作业一：移动业务大厅

一、移动业务大厅的框架

1.1 项目框架

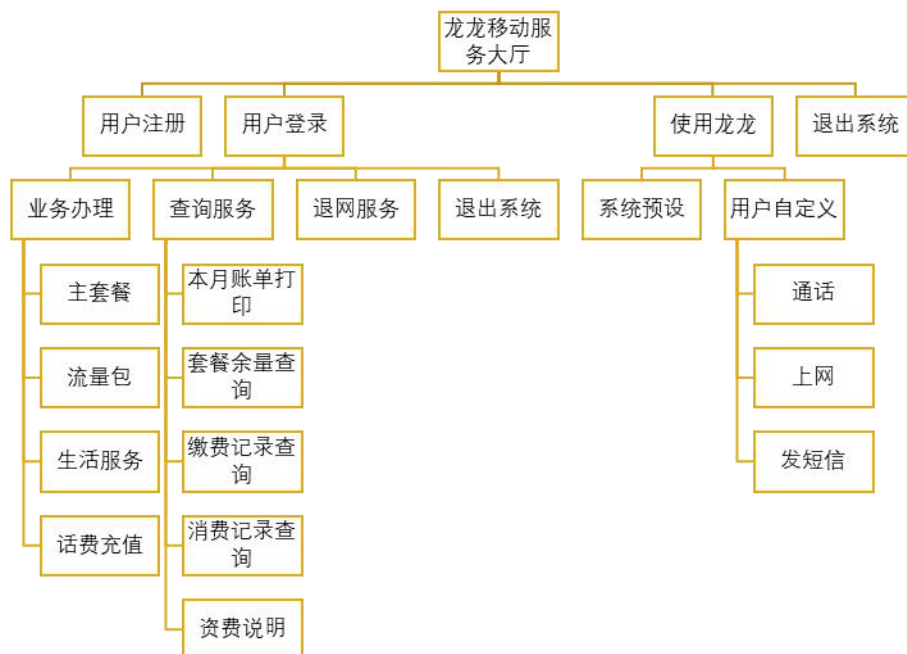


图 1-1

如图所示，我的移动服务大厅分为三级菜单。进入服务大厅后将打印一级菜单，有四种模式供用户选择，分别是用户注册、用户登录、使用龙龙、退出系统。用户只需要输入对应内容的编号，即可进入相应子菜单中。选择一级菜单服务内容并进入服务大厅的实现封装于 ServiceHall 类中的 RootMenu 函数中。主函数只需要调用 RootMenu 函数即可完成服务大厅需要完成的所有工作。

进入用户注册子菜单后将提供九个随机生成手机号供用户选择，并要求顾客输入姓名、密码等信息，并要求顾客再次确认密码无误后，为顾客开通套餐并按要求充值。完成上述内容后将回到根目录，让顾客选择登录办理其他业务、立即使用或退出系统。

进入用户登录子菜单后将向用户展示二级菜单，分别有业务办理、查询服务、退网服务和退出系统四个子菜单，其中业务办理子菜单下又有主套餐、流量包、生活服务、话费充值四个子菜单，查询服务下有帐单打印、套餐余量查询、缴费记录查询、消费记录查询、资费说明五个子菜单。该菜单的设置是结合了作业要求和移动、电信、联通三大通信运营商的网站架构进行设计的，并增添了一些其他功能，丰富用户的体验度。

进入使用龙龙菜单后将先进入系统预设示范样例，从事先内置的六个场景中随机抽取一个进行使用，向用户展示如何使用龙龙移动，然后会进入自定义场景，根据用户的套餐提供可使用的通话、上网、发短信的场景，并可以由用户自定义选择通话的对象、通话时长、发短信的对象、发短信条数、上网耗费的流量等，增加灵活度。

进入退出系统的子菜单后，程序会打印“谢谢，欢迎下次光临”，并结束整个系统。

```

1  import java.io.BufferedWriter;
2  import java.io.FileWriter;
3  import java.text.DecimalFormat;
4  import java.util.*;
5  |
6  public class ServiceHall implements InputAssure{
7      static private Customer customers = new Customer();
8      static private Util utility = new Util();
9      static private int getInput(Scanner scanner, int input, int up, int down) throws Exception{...}
31  static private void RootMenu() throws Exception {...}
81
82  static private int FirstMenu_1() throws Exception {...}
168  static private int FirstMenu_2() throws Exception{...}
286  static private int FirstMenu_3(){...}
883
884  static private int SecondMenu_1(SimCard curr_customer){...}
942  static private int SecondMenu_2(SimCard curr_customer) throws Exception{...}
999
1000  static private int ThirdMenu_1(SimCard curr_customer){...}
1176  static private int ThirdMenu_2(SimCard curr_customer){...}
1205  static private int ThirdMenu_3(SimCard curr_customer){...}
1221  static private int ThirdMenu_4(SimCard curr_customer){...}
1246
1247  static public void main(String[] args) throws Exception {
1248      RootMenu();
1249  }

```

图 1-2

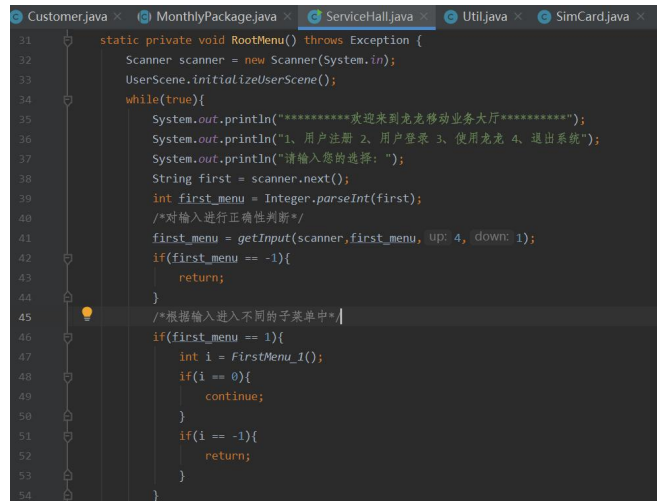
1.2 实现框架

项目构建了 ServiceHall 类、SimCard 类、Customer 类、Util 类、Userscene 类、MyException 类、MonthlyPackage 父类、TalkingTooMuch 子类、NetWorm 子类、Superman 子类、NormalPackage 子类，以及 Call 接口、Surf 接口、Text 接口、DataPack 接口、InputAssure 接口。其中 ServiceHall 类进行服务大厅框架的搭建；SimCard 类即用户类，记录用户的电话号码、用户名、密码、选择的套餐、余额等信息；Customer 类相当于数据库，记录每一个用户信息以及每一个用户的消费和充值信息；Util 类封装了一些常用的工具函数，没有数据成员；Userscene 类是为了初始化使用龙龙移动场景而建的；MyException 类是继承于 Exception 类的一个自定义报错类，为了服务于特定的捕获异常应用场景；MonthlyPackage 是套餐的父类，抽象了通话时间、流量、短信条数、套餐标号、套餐名、套餐花费、以及打印信息的函数；TalkingTooMuch 类、NetWorm 类、Superman 类、NormalPackage 类是继承于 MonthlyPackage 类的子类，实现了打印以及消费的函数重写，并根据不同套餐的特点实现 Call、Surf、Text 接口；NormalPackage 类的创建是为了增加用户的灵活度，不限制套餐用量，即普通计费；DataPack 接口是用来实现流量包的增添而设计的；InputAssure 接口是实现对用户选择子菜单或电话号码时输入的正确性的判断。

二、移动业务大厅的功能

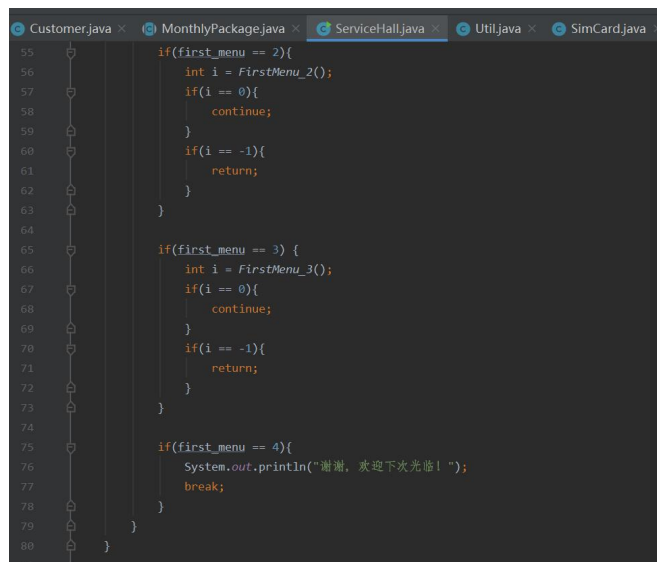
2.1 一级菜单的实现

采用 while 循环中嵌套 switch-case 结构,通过不同子菜单函数返回值的不同选择继续停留在一级菜单，让用户继续操作还是直接退出系统。



```
31 static private void RootMenu() throws Exception {
32     Scanner scanner = new Scanner(System.in);
33     UserScene.initializeUserScene();
34     while(true){
35         System.out.println("*****欢迎来到龙龙移动业务大厅*****");
36         System.out.println("1、用户注册 2、用户登录 3、使用龙龙 4、退出系统");
37         System.out.println("请输入您的选择: ");
38         String first = scanner.next();
39         int first_menu = Integer.parseInt(first);
40         /*对输入进行正确性判断*/
41         first_menu = getInput(scanner,first_menu, up: 4, down: 1);
42         if(first_menu == -1){
43             return;
44         }
45         /*根据输入进入不同的子菜单中*/
46         if(first_menu == 1){
47             int i = FirstMenu_1();
48             if(i == 0){
49                 continue;
50             }
51             if(i == -1){
52                 return;
53             }
54         }
```

图 2-1



```
55 if(first_menu == 2){
56     int i = FirstMenu_2();
57     if(i == 0){
58         continue;
59     }
60     if(i == -1){
61         return;
62     }
63 }
64
65 if(first_menu == 3) {
66     int i = FirstMenu_3();
67     if(i == 0){
68         continue;
69     }
70     if(i == -1){
71         return;
72     }
73 }
74
75 if(first_menu == 4){
76     System.out.println("谢谢, 欢迎下次光临!");
77     break;
78 }
79 }
80 }
```

图 2-2

2.1.1 用户注册

用户注册功能实现了随机生成九个电话号码给用户选择、模拟一般注册流程要求用户输入两次密码并判断前两次密码是否一致、要求用户选择套餐并充值、判断用户输入的金额是否能支付套餐等功能。

```

83 Scanner scanner = new Scanner(System.in);
84 System.out.println("*****龙龙为您挑了9个电话号码, 请您选择*****");
85 /*随机生成9个电话号码并打印*/
86 HashMap<Integer, String> randomTele = new HashMap<>();
87 String tele;
88 Random r = new Random();
89 for(int i = 0; i < 9; i++){
90     tele = "189";
91     for(int j = 0; j < 8; j++){
92         int bit = r.nextInt(10);
93         tele = tele + bit;
94     }
95     if(randomTele.containsKey(tele)){
96         i--;
97     }
98     else{
99         randomTele.put(i, tele);
100     }
101 }
102 for(int i = 0; i < 9; i++){
103     String random_telephone = randomTele.get(i);
104     if((i + 1) % 3 == 0){
105         System.out.print((i+1)+". "+random_telephone+"\n");
106     }
107     else{
108         System.out.print((i+1)+". "+random_telephone+" ");
109     }

```

图 2-3

随机生成电话号码采用的是逐位随机的方法, 并判断生成的数必须在 0-9 的范围内, 电话号码的前三位都是“189”, 完成一个电话号码的生成后存入 hashmap 类型的 randomTele 中, 以方便接下来的打印电话号码的功能实现。

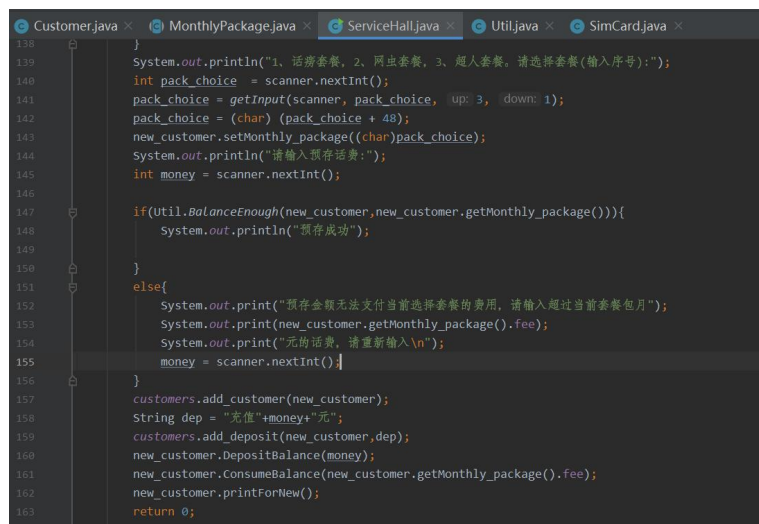
```

110 }
111 int telephone = Integer.parseInt(scanner.next());
112 telephone = getInput(scanner, telephone, up: 9, down: 1);
113 SimCard new_customer = new SimCard();
114 new_customer.setTelephone(randomTele.get(telephone));
115 System.out.println("请输入您的用户名:");
116 String name = scanner.next();
117 new_customer.setUsername(name);
118 System.out.println("请设置您的手机账户密码:");
119 String new_password = scanner.next();
120 System.out.println("请再输入一遍:");
121 String new_password_a = scanner.next();
122 int count = 0; //当输入密码前后多次不匹配则返回上一级
123 while(!new_password.equals(new_password_a)){
124     System.out.println("两次输入的密码不一致, 请重新输入");
125     new_password_a = scanner.next();
126     count++;
127     if(count > 3){
128         break;
129     }
130 }
131 if(new_password.equals(new_password_a)){
132     new_customer.setPassword(new_password);
133     System.out.println("密码设置成功");
134 }
135 else if(count > 3){
136     System.out.println("密码多次输入不匹配, 请重新注册");
137     return 0;
138 }

```

图 2-4

用户名和密码的设置实现如图, 此处实现时添加了对两次密码输入的一致性判断, 若多次输入不匹配, 将会退回到一级菜单的界面, 要求用户重新注册。



```

138 }
139
140 System.out.println("1. 100M套餐, 2. 200M套餐, 3. 300M套餐, 请选择套餐(输入序号):");
141 int pack_choice = scanner.nextInt();
142 pack_choice = getInput(scanner, pack_choice, up: 3, down: 1);
143 pack_choice = (char) (pack_choice + 48);
144 new_customer.setMonthly_package((char)pack_choice);
145 System.out.println("请输入预存话费:");
146 int money = scanner.nextInt();
147
148 if(Util.BalanceEnough(new_customer,new_customer.getMonthly_package())){
149     System.out.println("预存成功");
150 }
151 else{
152     System.out.print("预存金额无法支付当前选择套餐的费用, 请输入超过当前套餐包月");
153     System.out.print(new_customer.getMonthly_package().fee);
154     System.out.print("元的话费, 请重新输入\n");
155     money = scanner.nextInt();
156 }
157 customers.add_customer(new_customer);
158 String dep = "充值"+money+"元";
159 customers.add_deposit(new_customer, dep);
160 new_customer.DepositBalance(moeny);
161 new_customer.ConsumeBalance(new_customer.getMonthly_package().fee);
162 new_customer.printForNew();
163 return 0;
164

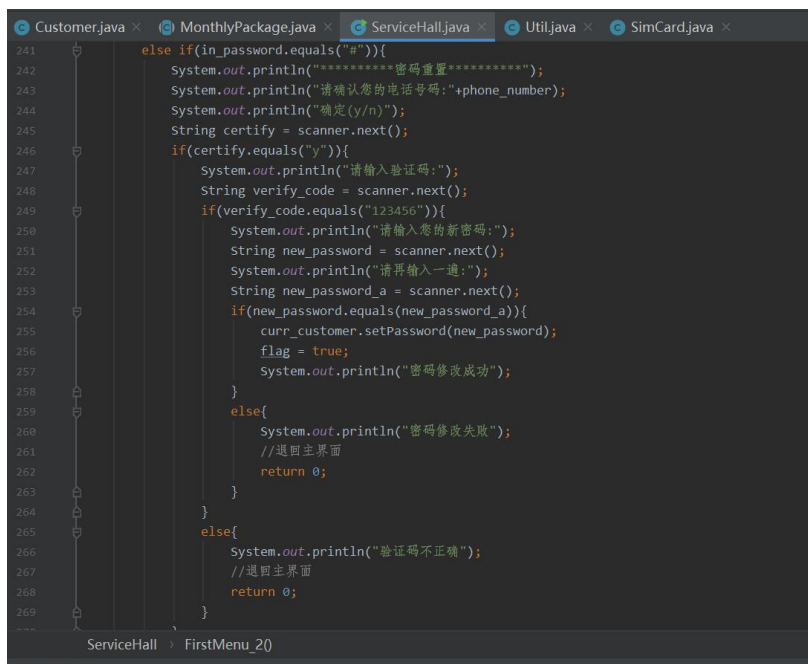
```

图 2-5

用户注册成功之后需要选择套餐并预存话费，若预存话费不足以支付当前套餐的费用，则需重新预存。预存成功后，`customers.add_customer(new_customer)`，`customers.add_deposit(new_customer, dep)`这两个语句就完成了将新注册的用户信息存入数据库 Customer 类的 `customers` 对象中，并完成 `new_customer.DepositBalance(moeny)`，`new_customer.ConsumeBalance(new_customer.getMonthly_package().fee)` 的充值和消费操作，更新 `new_customer` 中的余额，并打印信息，告诉用户他已经完成的所有操作内容。

2.1.2 用户登录

进入用户登录子菜单后，需要用户输入手机号和密码，如果用户的手机号在系统中不存在，将会及时的报错并退回到一级菜单，要求用户注册或重新输入手机号码。如果用户密码输入不正确，同样会退回一级菜单。当用户忘记密码时，可以输入#重置密码。由于暂时无法实现动态验证码的发送，所以验证码统一设为“123456”，但尽可能模拟真实的找回密码的场景。如果密码正确并登陆成功，则用户可以选择进入二级菜单。



```

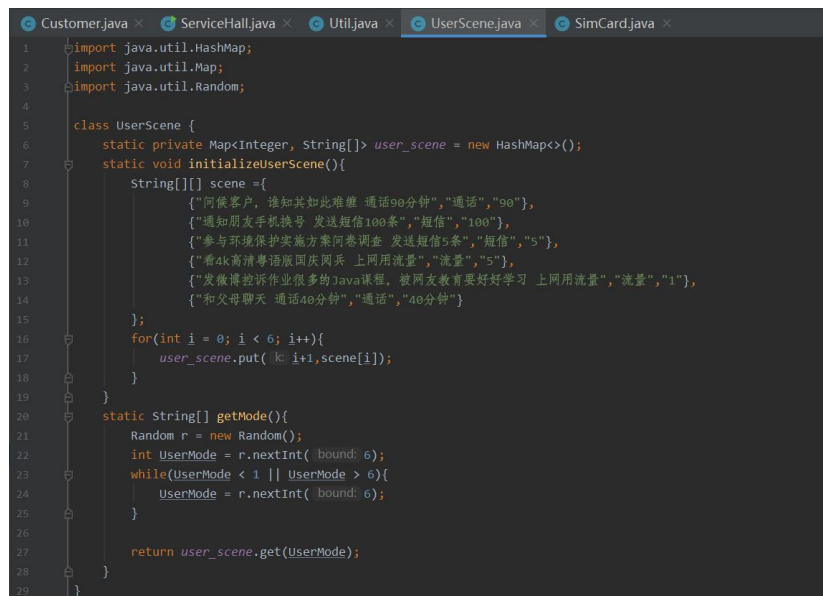
241 else if(in_password.equals("#")){
242     System.out.println("*****密码重置*****");
243     System.out.println("请确认您的电话号码:"+phone_number);
244     System.out.println("确定(y/n)");
245     String certify = scanner.next();
246     if(certify.equals("y")){
247         System.out.println("请输入验证码:");
248         String verify_code = scanner.next();
249         if(verify_code.equals("123456")){
250             System.out.println("请输入您的新密码:");
251             String new_password = scanner.next();
252             System.out.println("请再输入一遍:");
253             String new_password_a = scanner.next();
254             if(new_password.equals(new_password_a)){
255                 curr_customer.setPassword(new_password);
256                 flag = true;
257                 System.out.println("密码修改成功");
258             }
259             else{
260                 System.out.println("密码修改失败");
261                 //退回主界面
262                 return 0;
263             }
264         }
265     }
266     else{
267         System.out.println("验证码不正确");
268         //退回主界面
269         return 0;
270     }
271 }
272

```

图 2-6

2.1.3 使用龙龙

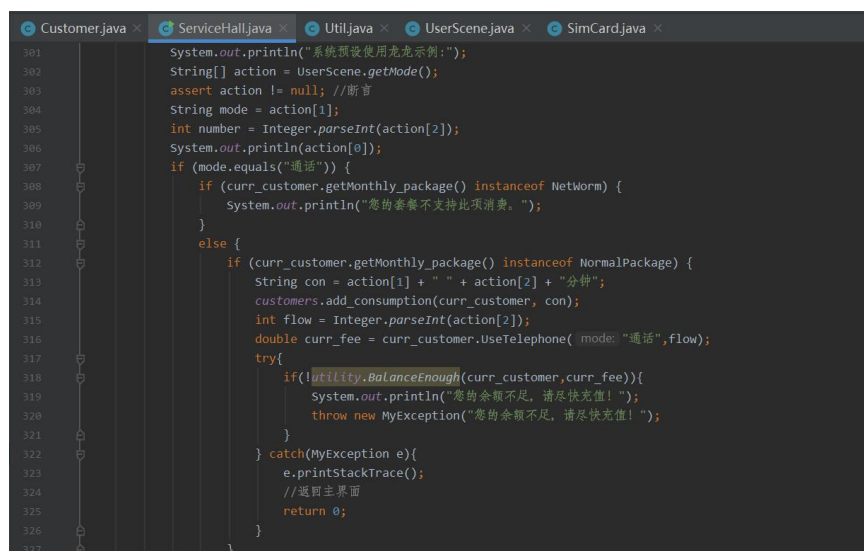
使用龙龙分为两部分，一部分是系统随机选择一种使用场景，根据该用户的套餐选择、卡内现存的通话时长、短信条数、流量剩余量、余额的情况进行判断，及时给出提醒并记录信息。应用场景的 UserScene 类实现如图 2-7。



```
1 import java.util.HashMap;
2 import java.util.Map;
3 import java.util.Random;
4
5 class UserScene {
6     static private Map<Integer, String[]> user_scene = new HashMap<>();
7     static void initializeUserScene(){
8         String[][] scene = {
9             {"问候客户，谁知其如此难缠 通话90分钟", "通话", "90"},
10            {"通知朋友手机换号 发送短信100条", "短信", "100"},
11            {"参与环境保护实施方案问卷调查 发送短信5条", "短信", "5"},
12            {"看4k高清粤语版国庆阅兵 上网用流量", "流量", "5"},
13            {"发微博控诉作业很多的Java课程，被网友教育要好好学习 上网用流量", "流量", "1"},
14            {"和父母聊天 通话40分钟", "通话", "40分钟"}
15        };
16        for(int i = 0; i < 6; i++){
17            user_scene.put(i, scene[i]);
18        }
19    }
20    static String[] getMode(){
21        Random r = new Random();
22        int UserMode = r.nextInt( bound: 6);
23        while(UserMode < 1 || UserMode > 6){
24            UserMode = r.nextInt( bound: 6);
25        }
26
27        return user_scene.get(UserMode);
28    }
29 }
```

图 2-7

另一部分是由用户自定义使用龙龙移动的具体场景，并对卡内现存的通话时长、短信条数、流量剩余量、余额的情况进行判断，看是否可以完成该场景，及时给出提醒并记录信息。首先要求用户输入手机号，并核实手机号是否存在于系统数据库 customers 对象中。然后开始使用。



```
301 System.out.println("系统预设使用龙龙示例:");
302 String[] action = UserScene.getMode();
303 assert action != null; //断言
304 String mode = action[1];
305 int number = Integer.parseInt(action[2]);
306 System.out.println(action[0]);
307 if (mode.equals("通话")) {
308     if (curr_customer.getMonthly_package() instanceof NetWorm) {
309         System.out.println("您的套餐不支持此项消费。");
310     }
311     else {
312         if (curr_customer.getMonthly_package() instanceof NormalPackage) {
313             String con = action[1] + " " + action[2] + "分钟";
314             customers.add_consumption(curr_customer, con);
315             int flow = Integer.parseInt(action[2]);
316             double curr_fee = curr_customer.UseTelephone( mode: "通话", flow);
317             try{
318                 if(!Utility.BalanceEnough(curr_customer, curr_fee)){
319                     System.out.println("您的余额不足，请尽快充值!");
320                     throw new MyException("您的余额不足，请尽快充值!");
321                 }
322             } catch(MyException e){
323                 e.printStackTrace();
324                 //返回主界面
325                 return 0;
326             }
327         }
328     }
329 }
```

图 2-8

图 2-9

以通话模式为例（图 2-8 图 2-9），程序会根据随机的得到的应用场景与用户当前的套餐进行匹配，若目前应用场景是“问候客户，谁知其如此难缠 通话 90 分钟”，则要完成使用操作的是通话 90 分钟。若当前用户的套餐是网虫套餐，则不支持此操作，通过打印信息提醒用户，若当前用户的套餐是话痨套餐，则程序会对该用户的套餐通话时长剩余量进行判断，如果未超出套餐包含的时间，则正常打印使用操作说明；如果超出套餐，则会及时提醒用户，并告知计费价格将按照超出套餐的价格进行计算。除此之外，程序还会对用户的余额进行判断，如果此次使用增加的额外费用导致余额不足，将会通过打印信息进行提醒并直接返回一级菜单，要求用户及时充值。

若是用户自定义模式下（图 2-10），以话痨套餐用户为例，他只可以进行打电话或发短信，则在这种情况下界面只会显示这两个功能，用户任选其一并按要求输入相应内容即可以完成使用龙龙移动的具体操作。程序除了实现这些操作以外，还会将用户每一次的操作和消费情况以字符串形式记录下来，存储到以电话号码为主键的链表中，方便打印详细消费信息（图 2-11 图 2-12）。

图 2-10

```
Customer.java x ServiceHall.java x Util.java x UserScene.java x SimCard.java x
40 public void add_deposit(SimCard cus, String de){
41     if(all_customers.containsValue(cus)){
42         LinkedList<String> temp;
43         temp = deposit.get(cus.getTelephone());
44         if(temp.getFirst().equals("您还没有充值")){
45             temp.remove(0);
46             temp.add(de);
47             deposit.put(cus.getTelephone(),temp);
48         }
49         else{
50             temp.add(de);
51             deposit.put(cus.getTelephone(),temp);
52         }
53     }
54     else{
55         System.out.println("您好像没有在龙龙移动注册噢，龙龙找不到您的信息");
56     }
57 }
```

图 2-11

```
Customer.java x ServiceHall.java x Util.java x UserScene.java x SimCard.java x
52 }
53 }
54 else{
55     System.out.println("您好像没有在龙龙移动注册噢，龙龙找不到您的信息");
56 }
57 }
58 public void add_consumption(SimCard s, String con){
59     if(all_customers.containsValue(s)){
60         LinkedList<String> temp;
61         temp = consumption.get(s.getTelephone());
62         if(temp.getFirst().equals("您还没有消费")){
63             temp.remove(0);
64             temp.add(con);
65             consumption.put(s.getTelephone(),temp);
66         }
67         else{
68             temp.add(con);
69             consumption.put(s.getTelephone(),temp);
70         }
71     }
72     else{
73         System.out.println("您好像没有在龙龙移动注册噢，龙龙找不到您的信息");
74     }
75 }
```

图 2-12

2.1.4 退出系统

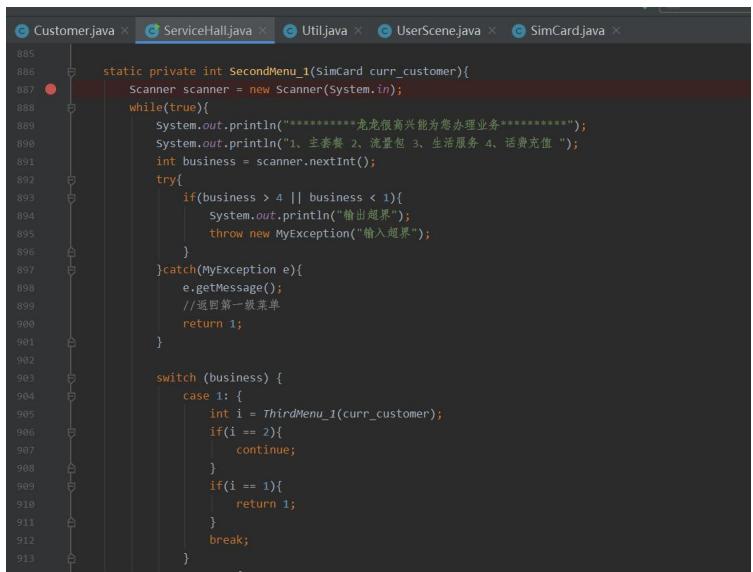
用户选择该子菜单的时候，程序结束。

2.2 二级菜单

二级菜单是进入一级菜单的用户登录子菜单并成功登录后所展示的子菜单，分别有业务办理、查询服务、办理退网和退出系统。同样采取 while 循环里嵌套 switch-case 结构进入二级菜单。根据各子菜单函数的返回值的不同，程序会执行返回上一级菜单或直接退出系统的不同操作。

2.2.1 业务办理

业务办理子菜单下有主套餐、流量包、生活服务、话费充值四个三级菜单，采取 while 循环里嵌套 switch-case 结构进入三级菜单。根据各子菜单函数的需要，程序执行返回二级、一级或退出系统的不同操作，同时在选择三级菜单时，如用户输入超界，则返回上一级（二级）菜单。



```
885
886 static private int SecondMenu_1(SimCard curr_customer){
887     Scanner scanner = new Scanner(System.in);
888     while(true){
889         System.out.println("*****龙龙很高兴能为您服务*****");
890         System.out.println("1. 主套餐 2. 流量包 3. 生活服务 4. 话费充值 ");
891         int business = scanner.nextInt();
892         try{
893             if(business > 4 || business < 1){
894                 System.out.println("超出范围");
895                 throw new MyException("输入超限");
896             }
897         }catch(MyException e){
898             e.getMessage();
899             //返回第一级菜单
900             return 1;
901         }
902
903         switch (business) {
904             case 1: {
905                 int i = ThirdMenu_1(curr_customer);
906                 if(i == 2){
907                     continue;
908                 }
909                 if(i == 1){
910                     return 1;
911                 }
912                 break;
913             }
```

图 2-13



```
914             case 2: {
915                 int i = ThirdMenu_2(curr_customer);
916                 if(i == 2){
917                     continue;
918                 }
919                 if(i == 1){
920                     return 1;
921                 }
922                 break;
923             }
924             case 3: {
925                 int i = ThirdMenu_3(curr_customer);
926                 if(i == 1){
927                     return 1;
928                 }
929                 break;
930             }
931             case 4: {
932                 int i = ThirdMenu_4(curr_customer);
933                 if(i==2){
934                     continue;
935                 }
936                 if(i == -1){
937                     return -1;
938                 }
939                 break;
940             }
941         }
942     }
```

图 2-14

2.2.2 查询服务

查询服务菜单采用 switch-case 结构,通过用户输入查询的内容的编号调用工具类 Util 和 Customer 类中对应的函数,向用户展示他想了解的信息。其中, printBill 函数是打印本月账单,内容包括使用的通话时长、流量使用量、短信发送条数,除套餐外额外消费钱数,账户上的余额等; printLeft 函数是打印套餐余量的,内容包括套餐中剩余的通话时长、流量、短信发送条数等; Customer 类 customers 对象中存储了所有用户的缴费记录和消费记录,因此调用 customers.print_deposit 和 customers.print_consumption 两个函数可以查询缴费记录和消费记录; printPackageInfo 函数是资费说明,将输出到文件中,详细展示每个套餐的内容和收费以及超出套餐后的计价。图 2-15 为二级菜单的实现,图 2-16 为打印账单函数实现,图 2-17 为套餐余量查询的实现,图 2-18 为缴费记录查询,图 2-19、图 2-20 为消费记录查询,根据作业要求输出到文件中,图 2-21、图 2-22 为打印资费说明,按要求输出到文件中。

```
Customer.java x ServiceHall.java x Util.java x UserScene.java x SimCard.java x
946 System.out.println("*****青龙很高兴能帮您查询*****");
947 System.out.println("1. 本月账单打印 2. 套餐余量查询 3. 缴费记录查询 4. 消费记录查询 5. 资费说明");
948 int information = scanner.nextInt();
949 try{
950     if(information < 1 || information > 5){
951         System.out.println("输出越界");
952         throw new MyException("输出越界, 返回上一级");
953     }
954 }catch (MyException e){
955     e.getMessage();
956     return 1;//返回一级菜单
957 }
958 switch(information){
959     case 1:{
960         Util.printBill(curr_customer);
961         break;
962     }
963     case 2:{
964         Util.printLeft(curr_customer);
965         break;
966     }
967     case 3:{
968         customers.print_deposit(curr_customer);
969         break;
970     }
971     case 4:{
972         customers.print_consumption(curr_customer);
973         break;
974     }
975     case 5:{
976         Util.PrintPackageInfo();
```

图 2-15

```
Customer.java x 消费记录.txt x ServiceHall.java x Util.java x TalkingTooMuch.java x 资费说明.txt
6 //打印本月账单
7 static void printBill(SimCard s){
8     DecimalFormat formatDouble = new DecimalFormat( pattern: "0.00");
9     MonthlyPackage p = s.getMonthly_package();
10     if(p instanceof TalkingTooMuch){
11         TalkingTooMuch talkingTooMuch = (TalkingTooMuch) p;
12         talkingTooMuch.printPackageUse(s);
13     }
14     if(p instanceof NetWorm){
15         NetWorm netWorm = (NetWorm) p;
16         netWorm.printPackageUse(s);
17     }
18     if(p instanceof Superman){
19         Superman superman = (Superman) p;
20         superman.printPackageUse(s);
21     }
22     if(p instanceof NormalPackage){
23         NormalPackage normalPackage = (NormalPackage) p;
24         normalPackage.printPackageUse(s);
25     }
26     if(s.getBalance() >= 0){
27         String format_left = formatDouble.format(s.getBalance());
28         System.out.println("本月余额:"+format_left+"元");
29     }
30     else{
31         System.out.println("本月余额:"+0+"元");
32         String format_left = formatDouble.format(s.getBalance());
33         System.out.println("本月欠费:"+format_left+"元");
34     }
}
```

图 2-16

```

4
5 class Util {
6     //打印本月账单
7     @ static void printBill(SimCard s){...}
8     //打印套餐余额
9     @ static void printLeft(SimCard s){
10         MonthlyPackage p = s.getMonthly_package();
11         if(p instanceof TalkingTooMuch){
12             TalkingTooMuch talkingTooMuch = (TalkingTooMuch) p;
13             talkingTooMuch.printPackageLeft(s);
14         }
15         if(p instanceof NetWorm){
16             NetWorm netWorm = (NetWorm) p;
17             netWorm.printPackageLeft(s);
18         }
19         if(p instanceof Superman){
20             Superman superman = (Superman) p;
21             superman.printPackageLeft(s);
22         }
23         if(p instanceof NormalPackage){
24             NormalPackage normalPackage = (NormalPackage) p;
25             normalPackage.printPackageLeft();
26         }
27     }
28     //判断用户的余额是否充足
29     @ static boolean BalanceEnough(SimCard s, MonthlyPackage p){...}
30     @ static boolean BalanceEnough(SimCard s, double num) { return s.getBalance() >= num; }
31     //用户每月套餐的月租消费

```

图 2-17

```

1 import java.io.BufferedWriter;
2 import java.io.FileWriter;
3 import java.io.IOException;
4 import java.util.*;
5
6 public class Customer {
7     Customer() { initializeCustomers(); }
8     private Map<String, SimCard> all_customers = new HashMap<>();
9     private Map<String, LinkedList<String>> consumption = new HashMap<>();
10    private Map<String, LinkedList<String>> deposit = new HashMap<>();
11    private void initializeCustomers(){...}
12    public void add_customer(SimCard cus){...}
13    public void add_deposit(SimCard cus, String de){...}
14    public void add_consumption(SimCard s, String con){...}
15    public void print_consumption(SimCard s) throws Exception{...}
16    public void print_deposit(SimCard s) throws Exception{
17        if(all_customers.containsKey(s) && deposit.containsKey(s.getTelephone())){
18            LinkedList curr = deposit.get(s.getTelephone());
19            System.out.println(s.getTelephone()+"的充值记录:");
20            System.out.println(curr);
21        }
22        else{
23            System.out.println("您好像没有在龙龙移动注册噢, 龙龙找不到您的信息");
24        }
25    }
26    public void remove(String phone){...}
27    public SimCard getCustomer(String phone) { return all_customers.getOrDefault(phone, null); }
28    public void print_all_customers(){...}
29 }

```

图 2-18

```

58 public void add_deposit(SimCard cus, String de){...}
59 public void add_consumption(SimCard s, String con){...}
60 public void print_consumption(SimCard s) throws Exception{
61     if(all_customers.containsKey(s) && consumption.containsKey(s.getTelephone())){
62         LinkedList curr = consumption.get(s.getTelephone());
63         System.out.println(s.getTelephone()+"的消费记录:");
64         System.out.println(curr);
65         FileWriter fw = new FileWriter("消费记录.txt");
66         try{
67             BufferedWriter bw = new BufferedWriter(fw);
68             String str = s.getTelephone() + " " + curr.getFirst()+"\n";
69             bw.append(str);
70             fw.flush();
71             for(int i = 1; i < curr.size(); i++){
72                 str = s.getTelephone() + " " + curr.get(i)+"\n";
73                 bw.append(str);
74                 bw.flush();
75             }
76         }catch (Exception e){
77             e.printStackTrace();
78         }
79         fw.close();
80     }
81     else{
82         System.out.println("您好像没有在龙龙移动注册噢, 龙龙找不到您的信息");
83     }
84 }

```

图 2-19

```
1 18902810777 短信 12条
2 18902810777 通话 90分钟
3 18902810777 通话 80分钟
4
```

图 2-20

```
93 static boolean balanceEnough(SimCard s, double num) { return s.getBalance() >= num; }
94 //用户每月套餐的月租消费
95 static void ConsumeMonthlyPackage(SimCard s){...}
117 //资费说明
118 static void PrintPackageInfo()throws Exception{
119     FileWriter fw = new FileWriter( fileName: "资费说明.txt");
120     try{
121         BufferedWriter bw = new BufferedWriter(fw);
122         bw.write( str: "-----资费说明-----\n");
123         TalkingTooMuch t = new TalkingTooMuch();
124         NetWorm n = new NetWorm();
125         Superman s = new Superman();
126         NormalPackage np = new NormalPackage();
127         t.printToText(bw);
128         n.printToText(bw);
129         s.printToText(bw);
130         t.printOutToText(bw);
131         np.printToText(bw);
132     }catch(Exception e){
133         System.out.println("文件不存在");
134         e.printStackTrace();
135     }
136     fw.close();
137 }
138 }
139 }
140 }
```

图 2-21

```
1 -----资费说明-----
2 套餐类型: 话费套餐
3 通话时长: 1500分钟
4 上网流量: 0GB
5 短信条数: 100条
6 月资费: 150元
7
8 套餐类型: 网话套餐
9 通话时长: 0分钟
10 上网流量: 3GB
11 短信条数: 0条
12 月资费: 160元
13
14 套餐类型: 超人套餐
15 通话时长: 2000分钟
16 上网流量: 1GB
17 短信条数: 150条
18 月资费: 170元
19
20 超出套餐的计费:
21 通话: 0.2元/分钟
22 上网流量: 0.3元/MB
23 短信: 0.1元/条
24
25 普通计费
26 通话: 0.3元/分钟
27 上网流量: 0.3元/MB
28 短信: 0.2元/条
29
```

图 2-22

2.2.3 办理退网和退出系统

选择办理退网，数据库会将用户的信息移除。选择退出系统将使程序结束。

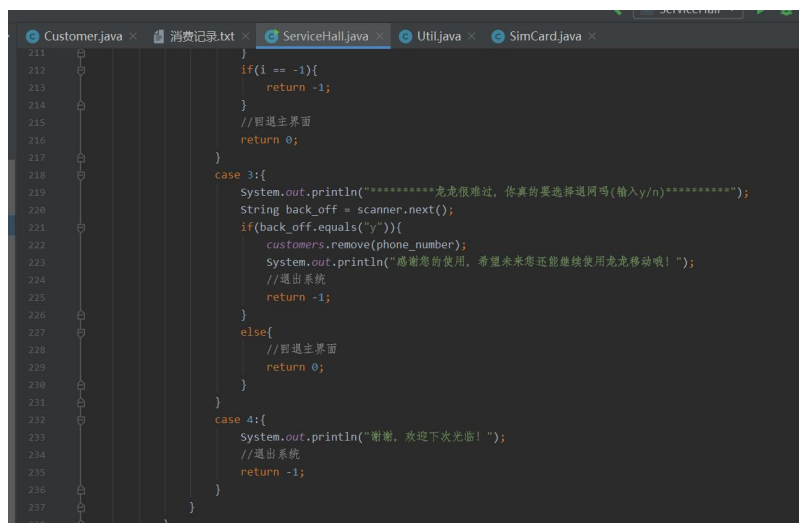


图 2-23

2.3 三级菜单

三级菜单是进入二级菜单办理业务之后对细分的主套餐、流量包、生活服务和话费充值进行更具体的实现。

2.3.1 主套餐

主套餐有四种选择，分别为话痨套餐、网虫套餐、超人套餐和普通计费，用户进入后选择任意一种套餐，输入相应编号后会打印该套餐的计费说明，并根据用户在用的套餐选择判断是否可以更换成当前查询的套餐。如果用户在用的套餐与查询套餐一致，则会有信息提示，无法进行套餐更换。如果可以，用户输入 y，如果不可以则系统自动返回到上一级菜单（二级菜单）。用户选择 y 后，程序还会判断用户当前余额是否足够支付新的套餐，如果余额不足，会提醒用户先充值再更换套餐。具体实现如图 2-24、图 2-25，以话痨套餐（TalkingTooMuch）为例。

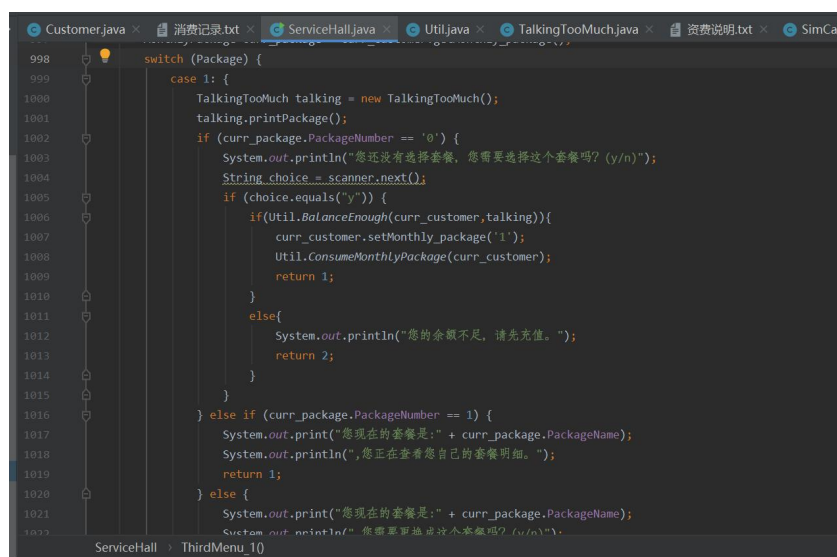


图 2-24

```

1013         return 2;
1014     }
1015 }
1016 } else if (curr_package.PackageNumber == 1) {
1017     System.out.print("您现在的套餐是:" + curr_package.PackageName);
1018     System.out.println("您正在查看您自己的套餐明细。");
1019     return 1;
1020 } else {
1021     System.out.print("您现在的套餐是:" + curr_package.PackageName);
1022     System.out.println("您需要更换成这个套餐吗? (y/n)");
1023     String choice = scanner.next();
1024     if (choice.equals("y")) {
1025         if (Util.BalanceEnough(curr_customer, talking)) {
1026             curr_customer.setMonthly_package("1");
1027             Util.ConsumeMonthlyPackage(curr_customer);
1028             return 1;
1029         } else {
1030             System.out.println("您的余额不足, 请先充值。");
1031             return 2;
1032         }
1033     }
1034 }
1035 }
1036 break;
1037 }

```

图 2-25

2.3.2 流量包

流量包一开始的设计是为了方便所有用户单独增加流量而设计的。但考虑到作业要求中话唠套餐用户没有上网使用流量的权限, 因此该流量包只对使用网虫套餐、超人套餐和普通计费的用户有效, 购买流量包将可以增加他们的套餐内的流量。网虫套餐、超人套餐、普通计费的三个类将会重写 DataPack 接口。如图 2-26、图 2-27、图 2-28。

```

1180     System.out.println("*****龙龙超大流量包等你来拿*****");
1181     System.out.println("1. 1GB 2. 2GB 3. 3GB ");
1182     int datapack = scanner.nextInt();
1183     try {
1184         if (datapack < 1 || datapack > 3) {
1185             throw new MyException("输出越界, 返回上一级");
1186         }
1187     } catch (MyException e) {
1188         e.getMessage();
1189         return 2;
1190     }
1191     if (curr_customer.getMonthly_package() instanceof NetWorm) {
1192         ((NetWorm) curr_customer.getMonthly_package()).setDataPack(datapack);
1193         if (Utility.BalanceEnough(curr_customer, num: datapack*10)) {
1194             curr_customer.ConsumeBalance( money: datapack*10);
1195         } else {
1196             System.out.println("您的余额不足, 请先充值。");
1197             return 2;
1198         }
1199     }
1200     else if (curr_customer.getMonthly_package() instanceof Superman) {
1201         ((Superman) curr_customer.getMonthly_package()).setDataPack(datapack);
1202         if (Utility.BalanceEnough(curr_customer, num: datapack*10)) {
1203             curr_customer.ConsumeBalance( money: datapack*10);
1204         }
1205     }

```

图 2-26

```

1 public interface DataPack {
2     void setDataPack(int i);
3 }
4

```

图 2-27

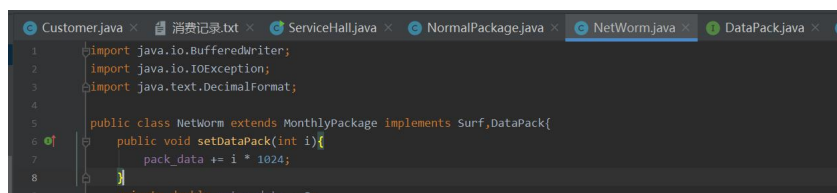


图 2-28

2.3.3 生活服务

根据三大运营商的网站设计，我在这里增加了一个生活服务，用户可以在移动服务大厅购买到视频 VIP 和音乐 VIP，设计的目的是为了练习链表 (LinkedList) 的使用。如图 2-29、图 2-30。

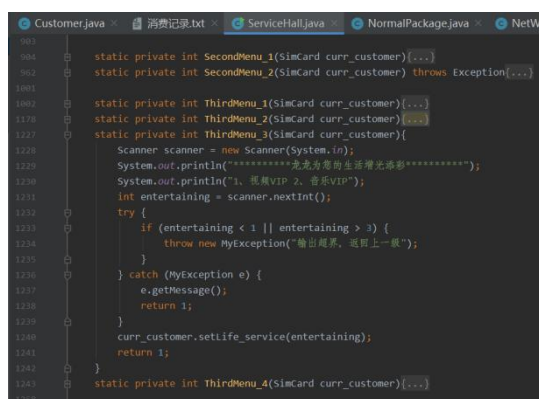


图 2-29

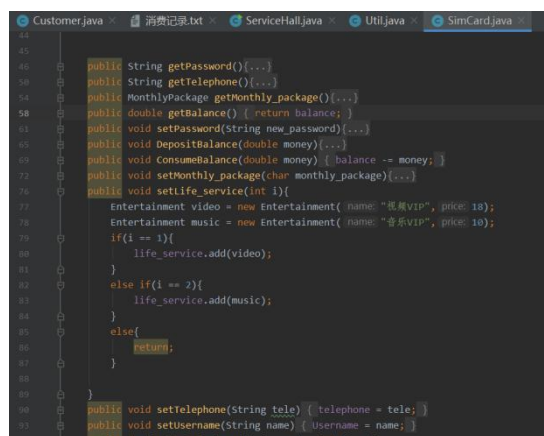


图 2-30

2.3.4 话费充值

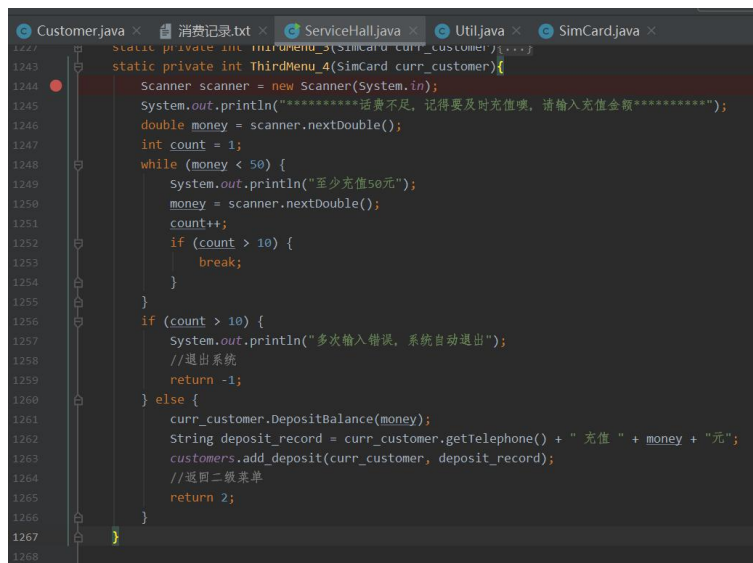


图 2-31

如图 2-31,话费充值子菜单进入后会请用户输入充值金额,每次充值至少为 50 元,若用户没有按要求充值,则需要重新输入,若用户恶意输入(输入不符合要求超过十次),程序将会自动退出。充值成功的话将会将此次充值记录存储到数据库中。

三、用户的使用

以下图片将展示实现的部分功能和界面。

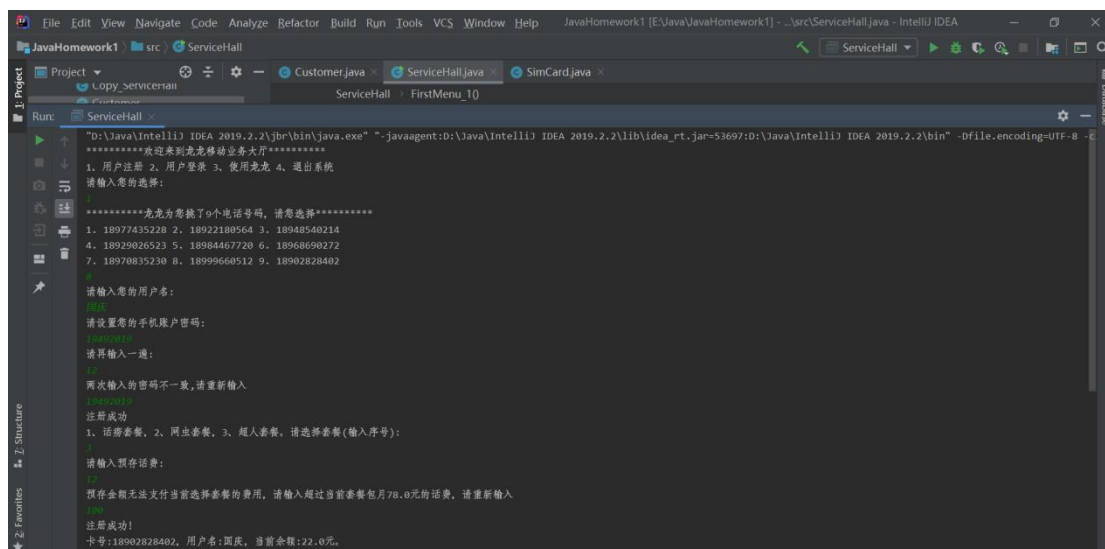


图 3-1 用户注册

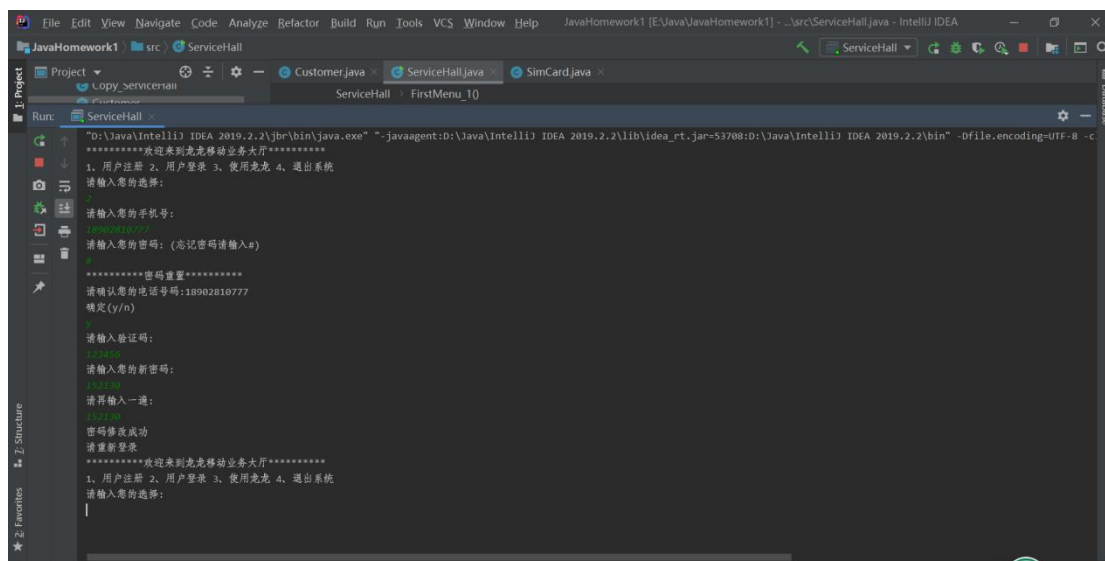


图 3-2 用户登录忘记密码

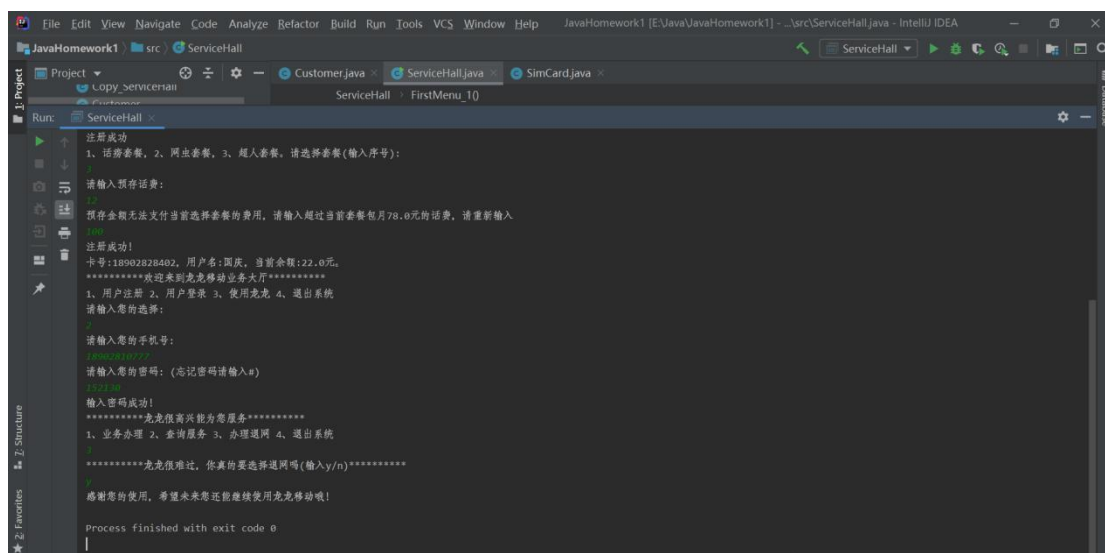


图 3-3 办理退网

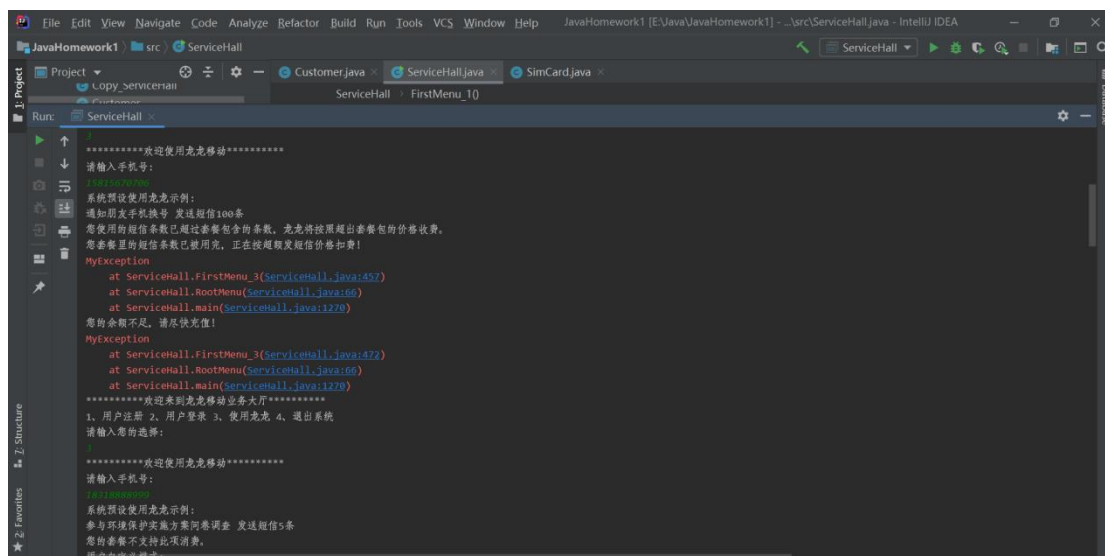


图 3-4 使用龙龙移动

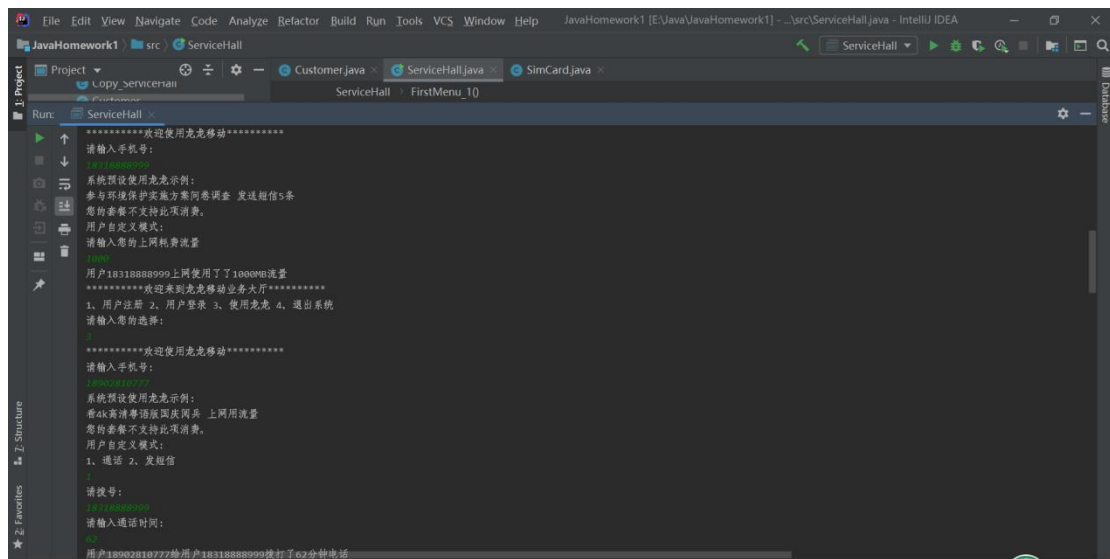


图 3-5 使用龙龙移动

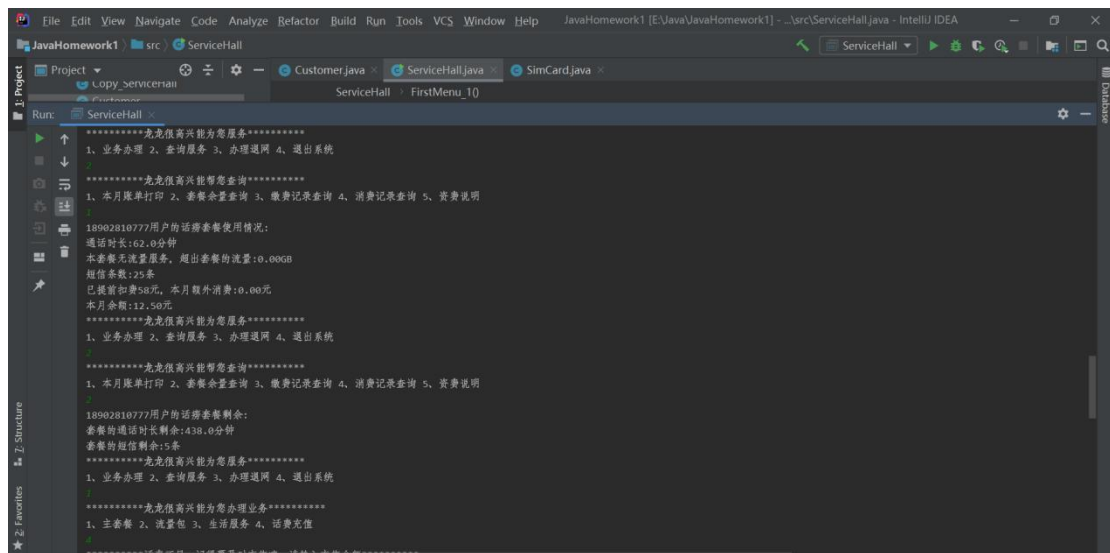


图 3-6 打印当月账单、打印套餐余量

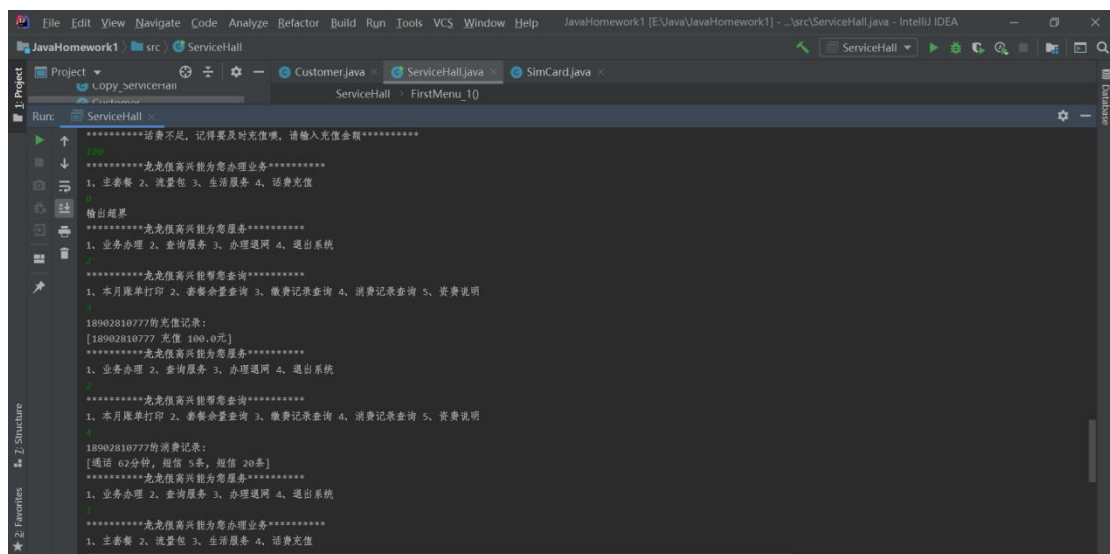


图 3-7 打印消费记录、打印充值记录

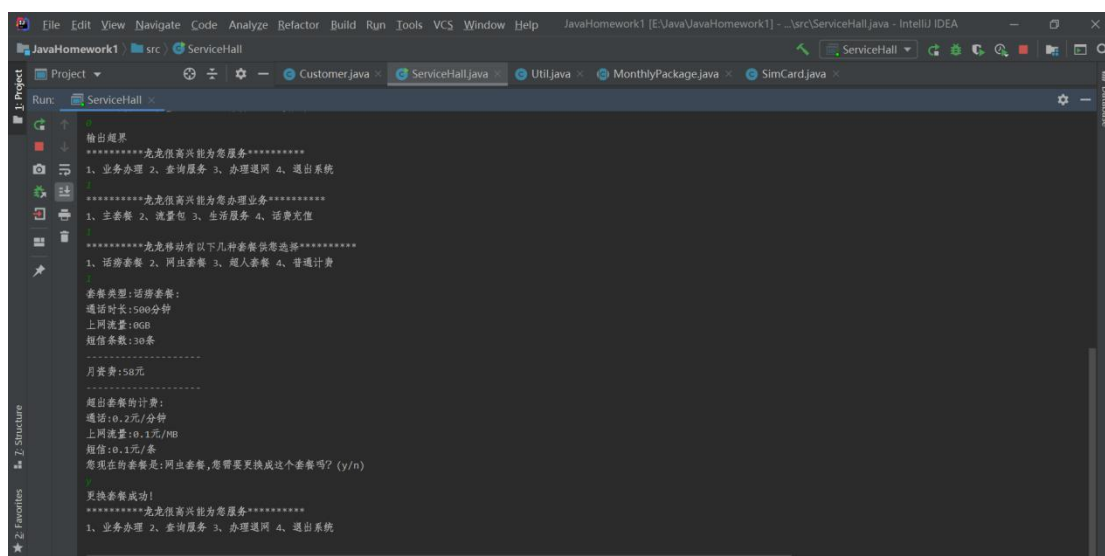


图 3-8 更换套餐

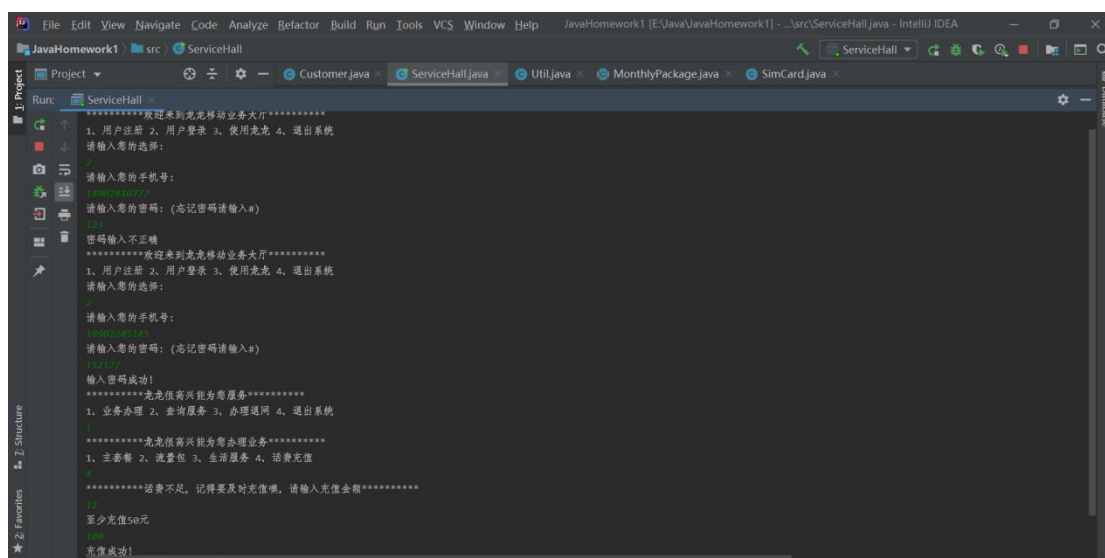


图 3-9 充值

四、问题与改进

4.1 代码设计

本次移动服务大厅项目的开发,设计时存在比较多相似度高的代码。因此在完成作业时,一开始选择复制粘贴的方法,但这给后期调试和修改带来了非常大的不便。因此在调试过程中进行了进一步的封装。由于某些代码的封装效果并不好,因此还是在完成调试后还是对相似度高的代码逐一进行特例化实现(尤其在FirstMenu_3函数中)。这是后期学习更深入的Java知识后希望可以进一步去完善的板块。

另外,在进行代码设计的时候,Idea会对代码的优化给予建议,如对于类的static函数,建议直接以类名进行调用;有些类内函数适合设置为package-private,有些类内函数适合设置为private等。这些优化有一部分我还无法理解它的优越性,所以没有根据建议修改。但随着学习的深入,这些优化建议应该是可以根据自己的需求更好地采纳。

4.2 功能实现

本次移动服务大厅的设计，原则上按照作业要求进行开发，但过程中发现了一些不合理的地方，比如说套餐里没有流量包，则用户无法上网。因此我在设计所有的套餐子类中，对上网、通话和发短信的功能都进行了实现，但基于作业要求，某些函数没有进行具体的调用。

4.3 异常

本次移动服务大厅项目的调试阶段，计算欠费的时候由于扣费的执行顺序不正确而导致重复多扣了很多钱。这是由于没有很深入的理解 try-catch-throw 的异常机制而造成代码执行顺序设计不合理。因此在接下来的学习中，设计自定义异常、使用系统预设的异常等板块还需要多加练习。

但在实现对用户输入的内容正确性判断的功能中，我进行了优化，从只允许客户输入数字才能正常运行，到无论用户恶意输入什么内容，程序都能正确识别，正确判断，并保证程序正常运行。这个功能封装在 InputAssure 接口的 getInput 的函数实例化中，并有利于不断对该函数完善提升。

4.4 调试

Java 的调试和 C++ 的调试不一样，由于过多的封装，导致在调试的时候代码会经常进入系统底层的代码块，影响了调试的效率。这需要在接下来的学习中进一步摸索更好的调试策略。

4.5 代码的注释

由于我还没有养成非常好的写注释的习惯，而是通过命名方式来帮助自己理清思路，清楚地知道每一个函数和每一个步骤需要完成的工作。另外在功能实现部分我也已经比较详尽的写出整个开发的层次结构的实现，因此如果这仍然给老师和助教带来阅读代码上的困难，我先表示非常的抱歉。养成良好的写简明易懂过的注释的习惯也是接下来学习中需要重点关注的。