

# 综合项目实战

## 1. 奖客富翁系统

### (1) 任务描述

某商场要求开发一套奖客富翁系统，要求客户注册成为商场会员，登录之后，就可以参加抽奖活动了。

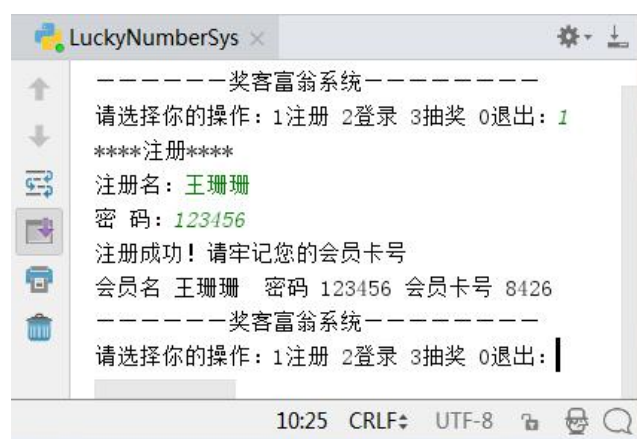
你主要负责本系统中的会员注册、登录、抽奖功能的实现。

### (2) 功能要求

具体功能要求如下：

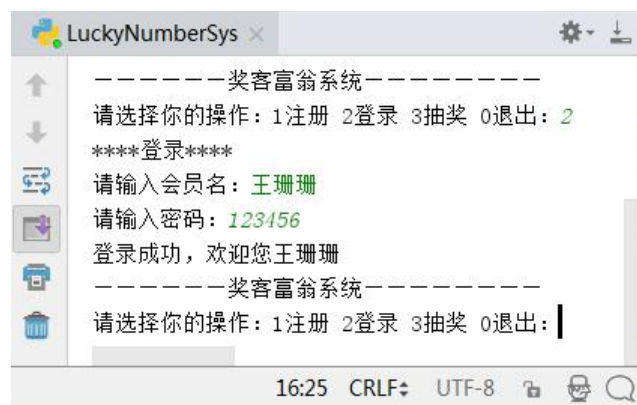
#### 1.注册

用户选择“注册”菜单，进入注册界面，输入会员名和密码后，系统提示注册成功，并给出会员卡号。



#### 2.登录

用户选择“登录”菜单，进入登录界面，输入会员名和密码，验证会员名和密码，给出登录成功与否的信息提示。

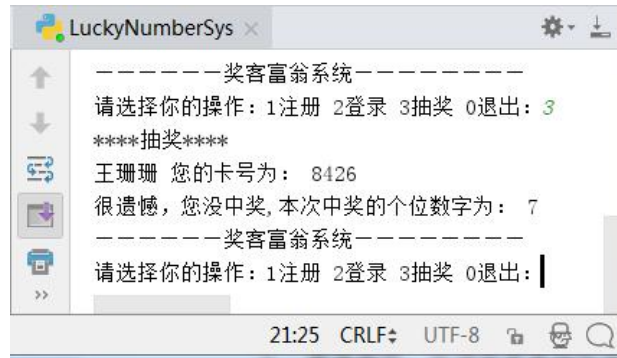


### 3.抽奖

随机生成一个数字（0~9），作为中奖号码。

用户选择“抽奖”菜单，判断用户是否是登录状态：

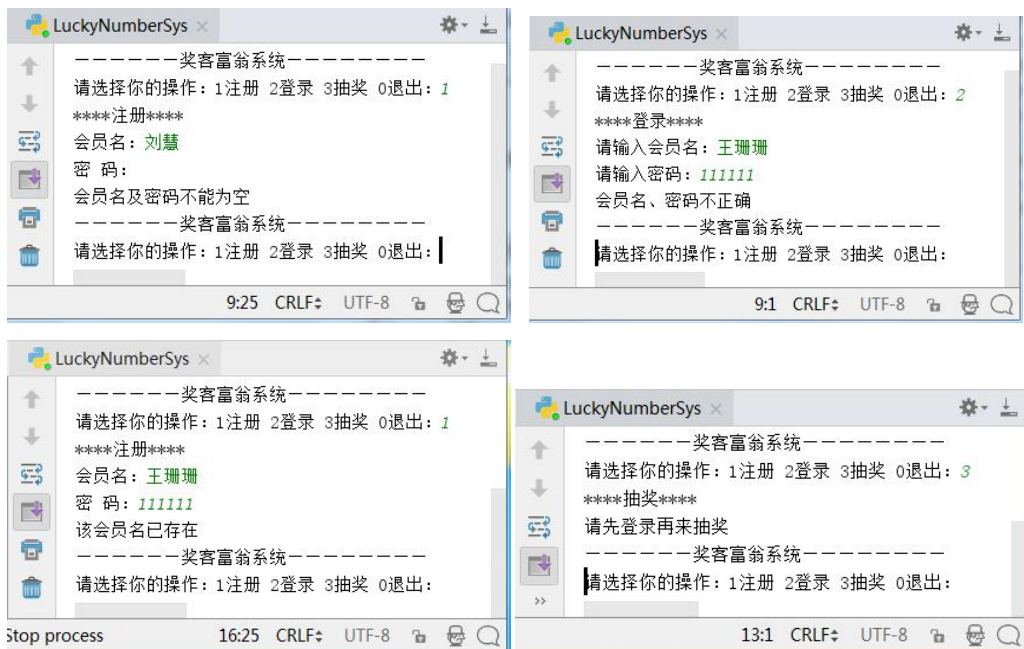
- 未登录：提示用户先登录再抽奖
- 登录状态：比对中奖号码与会员卡号个位数字是否相等，给出信息提示



### 4.纠错处理

要有各种纠错处理，界面友好，人机互动效果良好，信息提示明确。

- 会员名、密码不能为空
- 会员名唯一，不可重复
- 用户登录后才可以抽奖



### (3) 覆盖技能点

- 理解程序的逻辑业务
- 会使用全局变量和局部变量

- 会使用方法
- 会使用顺序、选择、循环、跳转语句控制程序业务

#### (4) 关键点分析

##### 1. 设计功能模块结构

根据功能需求，将各个功能模块化，使用函数实现，结构清晰，业务逻辑准确。下图所示的功能结构：



##### 2. 使用全局变量存储会员信息

根据项目的需求，程序运行期间，需要保存所有注册的会员信息，并且在各个功能模块之间要能够共享这些信息，所以，使用全局变量最合适不过。

全局变量主要存储会员信息：会员名、密码、会员卡号

##### 3. 使用列表存储会员信息

可以使用列表存储会员的信息，但是要考虑纠错问题，因为列表的一个特性是允许元素重复，而系统中要求会员名唯一。

#### (5) 实现用例

##### 用例 1：数据初始化

定义全局变量，用来存储会员信息：会员名、会员密码、会员卡号。

```
#定义全局变量，存储会员信息
name = []#会员姓名
password = []#会员密码
card = []#会员卡号
```

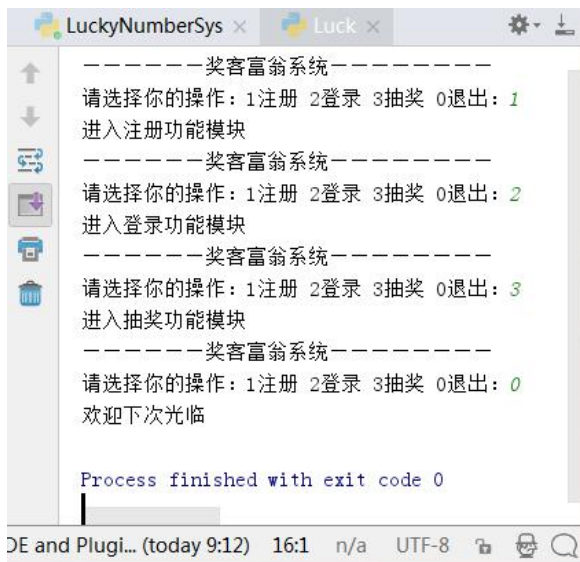
在其他地方使用全局变量，使用 `global` 关键字引用。

```
global name
global password
global card
```

## 用例 2：构建菜单项

运行程序，生成系统菜单。用户根据主菜单，输入功能编号实现菜单和菜单切换。具体要求如下：

- 输入“1 注册 2 登录 3 抽奖 0 退出”，进入不同功能模块。



- 要有纠错处理。



## 技术关键点：

使用 while 循环实现菜单的循环输出，直到用户输入 0 退出，使用 if 判断用户的输入菜单项，实现业务逻辑。

## 用例 3：实现注册功能

用户选择“1 注册”，进入注册功能，具体要求：

- 用户输入会员名、密码
- 会员名唯一，检测是否存在会员名
- 生成会员卡号，检测是否存在该卡号
- 必要的纠错处理
  - 用户输入不能为空
  - 会员是否存在

## ■ 会员卡号是否存在

### 技术关键点:

- 利用列表的 `append()` 方法，实施会员信息的增加。

```
#写入会员信息列表
name.append(name_re)
password.append(password_re)
card.append(card_re)
```

- 验证会员是否存在

```
if name_re in name:
    #提示信息
```

### 用例 4：实现登录功能

用户选择“2 登录”，进入登录功能，具体要求：

- 输入会员名、密码
- 验证会员名、密码，给出登录成功与否的信息

### 技术关键点:

判断会员名和密码是否正确，需要多个条件同时满足：

```
if (name_log in name) and (password_log in password) and
(name.index(name_log)==password.index(password_log)):
    print('登录成功，欢迎您%s'%name_log)
else:
    print('会员名、密码不正确')
```

### 用例 5：实现抽奖功能

用户选择“3 抽奖”，进入登录功能，具体要求：

- 验证用户是否登录状态
- 生成随机数（0~9），与会员卡个位数比对，提示中奖与否

### 技术关键点:

验证用户是否登录，需要设置一个全局变量，即登录标识，用来存储用户登录与否：

```
#定义全局变量，登录标识
name_log = ''
```

修改登录页面，添加登录成功，该标识赋值登录的会员名。

## 用例 6：功能测试

各种条件的组合，进行测试，完善功能。

### 2. 小课汽车租赁系统

#### (1) 任务描述

小课汽车租赁公司出租多种轿车和客车，出租费用以日为单位计算。出租车型及信息如表所示。

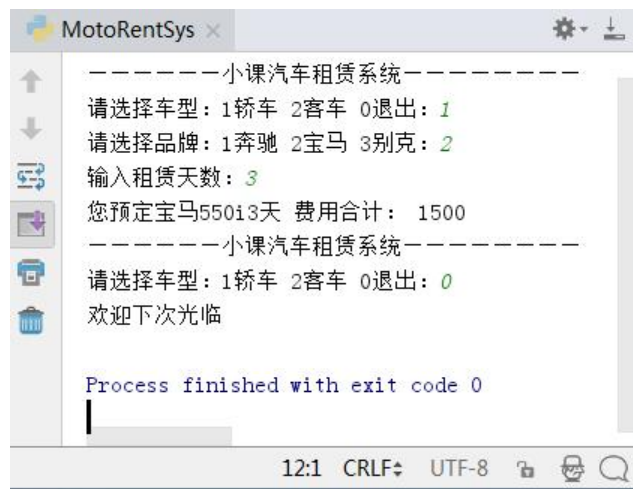
车 型	轿 车			客 车	
	奔驰 GLC	宝马 550i	别克林荫大道	金杯	金龙
日租费（元）	600	500	300	800	1500

要求采用面向对象思想进行设计，你负责完成计算汽车租赁价格合计的功能。

#### (2) 功能要求

具体功能要求如下：

- 用户选择车型和品牌，根据表格中的价格给出费用合计：



- 纠错处理

要有各种纠错处理，界面友好，人机互动效果良好，信息提示明确。

- 判断用户的菜单选择是否有误
- 租赁天数的判断

#### (3) 覆盖技能点

- 理解程序的逻辑业务
- 会使用面向对象程序设计思路解决问题
- 定义类和方法

- 继承关系处理
- 多态中的方法重写

#### (4) 关键点分析

##### 1. 由现实进行抽象

面向对象设计的过程就是抽象的过程。先通过在需求中找出名词的方式来确定类和属性，通过找出动词的方式来确定方法；然后对找到的词语进行筛选，剔除无关、不重要的词语，还要对词语之间的关系进行梳理，从而确定类、属性、属性值和方法。

面向对象程序设计分以下五步完成：

第一步：发现类

第二步：发现类的属性

第三步：发现类的方法

第四步：优化设计

第五步：梳理运行过程

##### 2. 关系分析

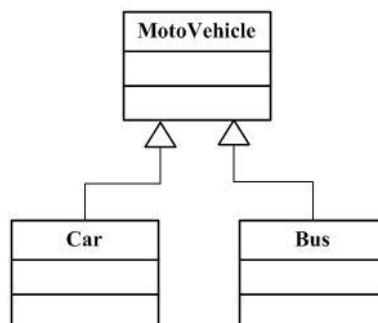
需求中和业务相关的名词主要有汽车租赁公司、汽车、轿车、客车、别克、宝马、金杯、金龙、商务舱 GL8、550i、林荫大道、座位数、日租金、租赁价等。动词主要是计算租赁价。

第一步：发现类

因为只有一家汽车租赁公司，所以在计算租赁价时不需要用该属性来标记某汽车，剔除该名词。

别克、宝马、金杯、金龙是汽车的品牌，没有必要设计为汽车的子类，作为汽车的一个属性品牌（brand）的值存在更简单、合理。

基于分析，从需求中抽象出如下类：汽车、轿车和客车。把汽车设计为父类，轿车和客车作为汽车的子类存在，结果下图所示。



第二步：发现类的属性

基于分析，汽车的属性有品牌（**brand**）等属性，品牌的属性值可以是奔驰、宝马、别克、金杯和金龙。

#### 第三步：发现类的方法

在本需求中，类的方法只有一个，就是计算租金，取名为 **calRent(days)**，设计为父类方法，子类重写。

#### 第四步：优化设计

把汽车设计为抽象类，把轿车和客车设计为子类。

#### 第五步：梳理运行过程

先编写汽车、轿车和客车的类代码，然后根据用户输入数据创建对象并调用 **calRent(days)** 方法计算租金。

### (5) 实现用例

#### 用例 1：创建父类

```
#汽车父类
class MotoVehicle():
    #汽车品牌
    #计算租金方法 calRent()
```

#### 用例 2：创建子类，继承父类

```
#子类：轿车
class Car(MotoVehicle):
    #重写父类 calRent() 方法
#子类：客车
class Bus(MotoVehicle):
    #重写父类 calRent() 方法
```

#### 用例 3：构建菜单项

运行程序，生成系统菜单。用户根据菜单，输入功能编号实现菜单和菜单切换。

#### 用例 4：创建对象调用方法，实现功能

根据菜单的选择，创建实例对象，调用实现方法，实现功能。

#### 用例 5：纠错完善功能

理顺业务思路，进行组合测试，增加必要的用户交互信息，完善功能。