**【设计目的】**

1、教学目的

本课程设计是学生学习完《Python程序设计》课程后，进行的一次全面的综合训练，通过课程设计，更好地掌握使用Python语言进行程序设计的方法，加深对Python语言特点和使用Python语言进行程序设计开发过程的理解，加强动手能力。其主要目的是：

（1）进一步培养学生Python程序设计的思想，加深对高级语言基本语言要素和控制结构的理解；

（2）针对Python语言中的重点和难点内容进行训练，独立完成有一定工作量的程序设计任务，同时强调好的程序设计风格。

（3）掌握Python语言的编程技巧和上机调试程序的方法。

（4）掌握程序设计的常用算法。

2、教学要求

（1）要求从所给题目中任选若干个，每个学生必须独立完成课程设计，不能互相抄袭。

（2）设计完成后，对所完成的工作进行答辩。

（3）要求写出一份详细的课程设计报告。

（4）程序设计题目，须提交相应的程序，并需提供加注释的源程序，能正常运行。

**【需求分析】**

1、问题

根据需求，该系统所应包含的信息有以下一些:

后台用户管理员的基本信息：

账号、密码

各路公交车的基本信息：

每路公交车对应的线路信息

2、系统

采用Python语言实现，开发环境为PythonCharm，可以运行在操作系统windows98及以上皆可。

3、运行要求

界面友好，操作方便，容错性强。

**【总体设计】**

1、系统流程设计

根据系统要求，设计如下的系统流程



2、系统模块设计

根据系统的功能设计，如下模块：

主界面模块：完成系统菜单的显示。

功能选择模块：根据用户的选择，完成其相应的功能。

线路查询模块：根据用户提供的线路，查询该线路所经过的站点。

站点查询的模块：根据用户提供的站点，查询经过该站点的所有线路。

后台管理员的模块：后台管理员可以进行一些操作，例如：删除所有线路、添加线路等

后台管理员的账号系统管理的模块：注册后台管理员的账号和密码。

换乘信息的模块：根据用户提供的起点和终点，帮助用户找到最便捷的路线。

**【详细设计】**

1、数据结构设计

根据系统要求，系统中需要保存的数据有后台管理员的信息，其结构如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据项名称 | 数据项系统表示 | 数据类型 | 数据长度 | 备注 |
| 账号 | pser | 整型 |  |  |
| 密码 | password | 字符串 |  |  |
| 线路 | Line | 字符串 |  |  |
| 站点 | Site | 字符串 |  |  |

使用Python语言创建的结构体如下：

def new\_(): #创建用户账号和密码的函数

file\_path = 'D:\公交查询系统\info.pkl'

a = input('请设置你的账号：')

b = input('请设置入你的密码：')

w = input('请再次输入你的的密码：')

d = {

'user': a,

'password': b

}

if b == w:

if os.path.exists(file\_path) and os.path.getsize(file\_path):

o = open('info.pkl', 'rb')

m = pickle.load(o)

with open('info.pkl', 'ab') as fp:

pickle.dump(d, fp)

print('您已经注册成功！！')

else:

with open('info.pkl', 'wb') as fp:

pickle.dump(d, fp)

print('您已经注册成功！！')

else:

print('对不起，你两次输入的密码不一致，请您重新输入！！')

def add\_line(): # 添加线路和站点的函数

file\_path = 'D:\公交查询系统\line\_info.pkl'

a = input('请输入添加的线路：')

b = input('请输入经过的站点：')

d = {

a: {b}

}

if os.path.exists(file\_path) and os.path.getsize(file\_path):

o = open('line\_info.pkl', 'rb')

m = pickle.load(o)

with open('line\_info.pkl', 'ab') as f:

pickle.dump(d, f)

print('您已经添加线路成功！！')

else:

with open('line\_info.pkl', 'wb') as fp:

pickle.dump(d, fp)

print('您已经添加线路成功了！！')

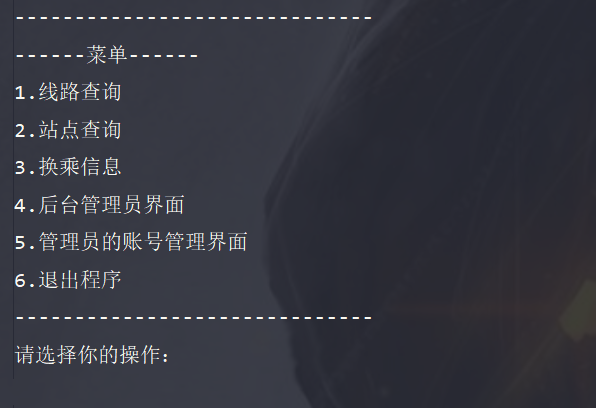
2、界面设计

主界面：

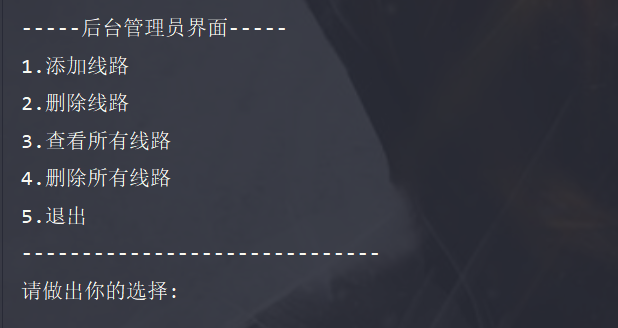
根据系统要求，设计一个主界面，显示系统的菜单，为了体现友好性，将在每个菜单前显示该菜单项对应的数字

用户输入菜单项前面的数字，主菜单消失，进入该模块的相应界面，进行相应的操作，完成后，回到主界面。

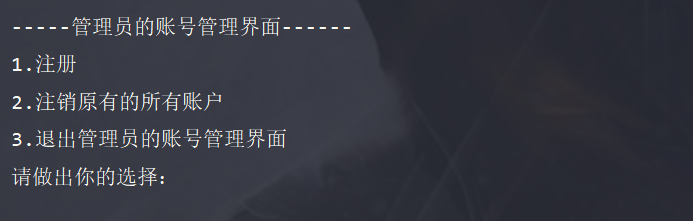
主界面：



后台管理员界面：



后台管理员的账号管理界面：



3、模块实现

主界面实现：调用print()函数，在屏幕上打印需要显示的字符信息。

功能选择模块实现：

接受从键盘输入的菜单选择，判断并调用相应的功能函数，完成其对应的功能。实现功能选择。

对应的函数：

1. 线路查询模块实现

模块说明：根据用户输入的线路，读取文件，从文件中查询出线路的所有站点。

对应程序：

def line\_check(): # 判断是否存在线路的函数

with open('line\_info.pkl', 'rb') as fp:

num = input('请输入你要查询的线路：\n')

while True:

try:

data = pickle.load(fp)

if num in data:

print('{}号线的路线经过的站点为{}'.format(num, data[num]))

return True

except:

break

def line\_find(): # 线路查询的函数

if line\_check():

pass

else:

print('您要查找的线路不存在....')

B.站点查询：

与线路查询类似

对应函数：

def site\_check(): # 判断站点是否存在的函数

with open('line\_info.pkl', 'rb') as fp:

num = input('请输入你要查询的站点名：\n')

while True:

try:

data = pickle.load(fp)

# 遍历字典

for key, value in data.items():

ls = str(value)

if num in ls:

print('经过{}站点的线路为{}号线'.format(num, key))

return True

except:

break

def site\_find(): # 站点查询的函数

if site\_check():

pass

else:

print('您要查询的站点不存在.....')

C. 后台管理员模块实现：

模块说明：用户需要登录用户名和密码才可以进入后台管理员的界面 ，进入后台管理员的界面后会有许多新的功能。

对应程序：

def login(): # 用户登录的函数

with open('info.pkl', 'rb') as fp:

q = input('请输入你的的账号：')

w = input('请输入你的的密码：')

while True:

try:

data = pickle.load(fp)

if q == data['user'] and w == data['password']:

print('您已经登录成功！！！')

return True

except:

break

def manager(): # 后台管理员界面的函数

if login():

while True:

print('-' \* 30)

print('-----后台管理员界面-----')

print('1.添加线路')

print('2.删除线路')

print('3.查看所有线路')

print('4.删除所有线路')

print('5.退出')

print('-' \* 30)

num = input('请做出你的选择:\n')

if num == '1':

add\_line()

if num == '5':

print('-' \* 25)

print('------菜单------')

print('1.线路查询')

print('2.站点查询')

print('3.换乘信息')

print('4.后台管理员界面')

print('5.管理员的账号管理界面')

print('6.退出程序')

print('-' \* 25)

break

if num == '4':

clear\_line\_all()

if num == '3':

show\_line\_all()

if num == '2':

clear\_line()

else:

print('请重新做出你的选择...')

else:

print('对不起，你登录失败，不可以让您进入后台管理员界面')

D.后台管理员的账号管理模块实现：

模块说明：用户可以在后台管理员的账号管理界面来注册或注销账号和密码。

对应函数：

def manager\_account(): # 管理员账号管理界面的函数

def new\_():

file\_path = 'D:\公交查询系统\info.pkl'

a = input('请设置你的账号：')

b = input('请设置入你的密码：')

w = input('请再次输入你的的密码：')

d = {

'user': a,

'password': b

}

if b == w:

if os.path.exists(file\_path) and os.path.getsize(file\_path):

o = open('info.pkl', 'rb')

m = pickle.load(o)

with open('info.pkl', 'ab') as fp:

pickle.dump(d, fp)

print('您已经注册成功！！')

else:

with open('info.pkl', 'wb') as fp:

pickle.dump(d, fp)

print('您已经注册成功！！')

else:

print('对不起，你两次输入的密码不一致，请您重新输入！！')

def xiaohui():

h = open('info.pkl', 'w+') # 清空文件里的内容。。。

h.truncate()

print('你已经销毁成功。。。')

while True:

print('-----管理员的账号管理界面------\n'

'1.注册\n'

'2.注销原有的所有账户\n'

'3.退出管理员的账号管理界面')

choice = input('请做出你的选择：\n')

if choice == '1':

new\_()

if choice == '2':

xiaohui()

elif choice == '3':

print('-' \* 35)

print('------菜单------')

print('1.线路查询')

print('2.站点查询')

print('3.换乘信息')

print('4.后台管理员界面')

print('5.管理员的账号管理界面')

print('6.退出程序')

print('-' \* 35)

break

else:

print('请重新做出你的选择')

E．换乘信息模块

模块说明：用户通过输入起点和终点，帮助用户找到从起点到终点的换乘路线。

对应函数：

def change\_line(): # 换乘信息的函数  
 begin = input('请输入你的起点站点：\n')  
 end = input('请输入你的终点站点：\n')  
 if direct\_line(begin, end):  
 pass  
 else:  
 ls = []  
 ls\_ = []  
 tag = True  
 data = check\_site(begin, end)  
 # print(type(data[0]),type(data[1]))  
 print(f'经过{begin}站点的线路为：{data[0]}，经过{end}站点的线路为{data[1]}') # 判断经过起始点和终点的对应线路  
 for i in range(len(list(data)[0])):  
 ls.append(show\_site(data[0][i]))  
 print(f"{line\_check(str(data[0][i]))}")  
 for j in range(len(list(data)[1])):  
 ls\_.append(show\_site(data[1][j]))  
 print(f"{line\_check(str(data[1][j]))}")  
 for i in ls: # i为集合的形式  
 for a in list(i):  
 for b in str(a).split(' '): # 切割字符串  
 if b in str(ls\_): # 判断是否存在相同的站点  
 tag = False  
 print(f"您可以通过从{data[0]}号线的{b}站转到{data[1]}号线来到达目的地！！")  
 if tag == True:  
 print('对不起，无法通过换乘来到达目的地')  
  
  
def direct\_line(begin, end): # 判断线路直达的函数  
 with open('line\_info.pkl', 'rb') as fp:  
 while True:  
 try:  
 data = pickle.load(fp)  
 # 遍历字典  
 for key, value in data.items():  
 ls = str(value)  
 if begin in ls and end in ls:  
 print(f'您可以通过{key}号线路直接到达')  
 return True  
 except:  
 break

**【课设总结】**

本学期实训课与Python语言课相结合，以上机实验操作为主，加深了我对Python语言的理解和运用能力，帮助我解决了许多Python语言学习时遇到的问题。通过在学习时遇到的问题查找资料使我接触和了解了更多课本以外的知识。特别是最后的大实训,让我认识到Python语言的重要性，同时还得出一个结论：知识必须通过应用才能实现其价值！有些东西以为学会了，但真正到用的时候才发现是两回事，所以我认为只有到真正会用的时候才是真的学会了。

**【课设感想】**

在设计的过程中遇到了许多问题，出现了许多bug，通过查找资料和询问老师最终都完美的解决了，这次课设很有意义！！！！

不足之处是程序可能存在漏洞，界面设计还不够十分完美！

Python程序源代码：

import os

import pickle

def main(): # 主函数

while True:

print('-' \* 30)

print('------菜单------')

print('1.线路查询')

print('2.站点查询')

print('3.换乘信息')

print('4.后台管理员界面')

print('5.管理员的账号管理界面')

print('6.退出程序')

print('-' \* 30)

num = int(input('请选择你的操作：\n'))

if num == 1:

line\_find()

continue

if num == 2:

site\_find()

continue

if num == 3:

change\_line()

continue

if num == 5:

manager\_account()

continue

if num == 6:

break

if num == 4:

manager()

continue

else:

print('请重新做出你的选择！！')

def manager\_account(): # 管理员账号管理界面的函数

def new\_():

file\_path = 'D:\公交查询系统\info.pkl'

a = input('请设置你的账号：')

b = input('请设置入你的密码：')

w = input('请再次输入你的的密码：')

d = {

'user': a,

'password': b

}

if b == w:

if os.path.exists(file\_path) and os.path.getsize(file\_path):

o = open('info.pkl', 'rb')

m = pickle.load(o)

with open('info.pkl', 'ab') as fp:

pickle.dump(d, fp)

print('您已经注册成功！！')

else:

with open('info.pkl', 'wb') as fp:

pickle.dump(d, fp)

print('您已经注册成功！！')

else:

print('对不起，你两次输入的密码不一致，请您重新输入！！')

def xiaohui():

h = open('info.pkl', 'w+') # 清空文件里的内容。。。

h.truncate()

print('你已经销毁成功。。。')

while True:

print('-----管理员的账号管理界面------\n'

'1.注册\n'

'2.注销原有的所有账户\n'

'3.退出管理员的账号管理界面')

choice = input('请做出你的选择：\n')

if choice == '1':

new\_()

if choice == '2':

xiaohui()

elif choice == '3':

print('-' \* 35)

print('------菜单------')

print('1.线路查询')

print('2.站点查询')

print('3.换乘信息')

print('4.后台管理员界面')

print('5.管理员的账号管理界面')

print('6.退出程序')

print('-' \* 35)

break

else:

print('请重新做出你的选择')

def login(): # 用户登录的函数

with open('info.pkl', 'rb') as fp:

q = input('请输入你的的账号：')

w = input('请输入你的的密码：')

while True:

try:

data = pickle.load(fp)

if q == data['user'] and w == data['password']:

print('您已经登录成功！！！')

return True

except:

break

def manager(): # 后台管理员界面的函数

if login():

while True:

print('-' \* 30)

print('-----后台管理员界面-----')

print('1.添加线路')

print('2.删除线路')

print('3.修改线路')

print('4.查看所有线路')

print('5.删除所有线路')

print('6.退出')

print('-' \* 30)

num = input('请做出你的选择:\n')

if num == '3':

revise\_line()

continue

if num == '1':

add\_line()

continue

if num == '6':

print('-' \* 25)

print('------菜单------')

print('1.线路查询')

print('2.站点查询')

print('3.换乘信息')

print('4.后台管理员界面')

print('5.管理员的账号管理界面')

print('6.退出程序')

print('-' \* 25)

break

if num == '5':

clear\_line\_all()

continue

if num == '4':

show\_line\_all()

continue

if num == '2':

clear\_line()

continue

else:

print('请重新做出你的选择...')

else:

print('对不起，你登录失败，不可以让您进入后台管理员界面')

def revise\_line(): # 定义修改线路的函数

num = input('请输入您要修改的线路：')

if check\_add\_line(num):

print('您要修改的线路已经存在了。。。。')

show\_line\_all()

tag = input('请确认是否继续修改线路：(1为是，0为否)\n')

if tag == '1':

line = input('请再次确认你要修改的线路：\n')

site = input('请输入你修改后的站点：\n')

with open('line\_info.pkl', 'rb') as fp:

ls = []

while True:

try:

data = pickle.load(fp)

# 遍历字典

for key, value in data.items():

if line in key:

ls.append({str(line): str(site)})

else:

ls.append(data)

except:

break

save\_data(ls) # 存储修改后的数据

print("您已经修改成功了！！！！")

else:

print('您要修改的线路不存在，需要创建一个新的线路。。。')

add\_line()

def line\_check(num): # 判断是否存在线路的函数

with open('line\_info.pkl', 'rb') as fp:

while True:

try:

data = pickle.load(fp)

if num in data:

print(f'{num}号线的路线经过的站点为{data[num]}')

return True

except:

break

def line\_find(): # 线路查询的函数

num = input('请输入你要查询的线路：\n')

if line\_check(num):

pass

else:

print('您要查找的线路不存在....')

def site\_check(): # 判断站点是否存在的函数

with open('line\_info.pkl', 'rb') as fp:

num = input('请输入你要查询的站点名：\n')

while True:

try:

data = pickle.load(fp)

# 遍历字典

for key, value in data.items():

ls = str(value)

if num in ls:

print(f'经过{num}站点的线路为{key}号线')

return True

except:

break

def site\_find(): # 站点查询的函数

if site\_check():

pass

else:

print('您要查询的站点不存在.....')

def check\_add\_line(num): # 检查添加线路是否存在的函数

with open('line\_info.pkl', 'rb') as fp:

while True:

try:

data = pickle.load(fp)

# 遍历字典

for key, value in data.items():

if num in key:

return True

except:

break

def add\_line(): # 添加线路的函数

file\_path = 'D:\公交查询系统\line\_info.pkl'

a = input('请输入添加的线路：')

b = input('请输入经过的站点：')

d = {

a: {b}

}

if os.path.exists(file\_path) and os.path.getsize(file\_path):

if check\_add\_line(a):

print('您要添加的线路已经存在了......')

else:

with open('line\_info.pkl', 'ab') as f:

pickle.dump(d, f)

print('您已经添加线路成功！！')

else:

with open('line\_info.pkl', 'wb') as fp:

pickle.dump(d, fp)

print('您已经添加线路成功了！！')

def clear\_line\_all(): # 清楚所有线路的函数

h = open('line\_info.pkl', 'w+') # 清空文件里的内容。。。

h.truncate()

print('你已经销毁成功。。。')

def show\_line\_all(): # 展示所有线路的函数

with open('line\_info.pkl', 'rb') as fp:

while True:

try:

data = pickle.load(fp)

# 遍历字典

for key, value in data.items():

print(f'{key}线路经过的站点是{value}')

except:

break

def clear\_line(): # 删除线路的函数

with open('line\_info.pkl', 'rb') as fp:

num = input('请输入你要删除的线路：\n')

ls = []

while True:

try:

data = pickle.load(fp)

# 遍历字典

for key, value in data.items():

if num in key:

pass

else:

ls.append(data)

except:

break

save\_data(ls)

def line\_check(num): # 判断是否存在线路的函数

with open('line\_info.pkl', 'rb') as fp:

while True:

try:

data = pickle.load(fp)

if num in data:

print(f'{num}号线的路线经过的站点为{data[num]}')

return True

except:

break

def show\_site(num): # 展示特定线路中的站点的函数

with open('line\_info.pkl', 'rb') as fp:

while True:

try:

data = pickle.load(fp)

if num in data:

return data[num]

except:

break

def change\_line(): # 换乘信息的函数

begin = input('请输入你的起点站点：\n')

end = input('请输入你的终点站点：\n')

if direct\_line(begin, end):

pass

else:

ls = []

ls\_ = []

tag = True

data = check\_site(begin, end)

# print(type(data[0]),type(data[1]))

print(f'经过{begin}站点的线路为：{data[0]}，经过{end}站点的线路为{data[1]}') # 判断经过起始点和终点的对应线路

for i in range(len(list(data)[0])):

ls.append(show\_site(data[0][i]))

print(f"{line\_check(str(data[0][i]))}")

for j in range(len(list(data)[1])):

ls\_.append(show\_site(data[1][j]))

print(f"{line\_check(str(data[1][j]))}")

for i in ls: # i为集合的形式

for a in list(i):

for b in str(a).split(' '): # 切割字符串

if b in str(ls\_): # 判断是否存在相同的站点

tag = False

print(f"您可以通过从{data[0]}号线的{b}站转到{data[1]}号线来到达目的地！！")

if tag == True:

print('对不起，无法通过换乘来到达目的地')

def direct\_line(begin, end): # 判断线路直达的函数

with open('line\_info.pkl', 'rb') as fp:

while True:

try:

data = pickle.load(fp)

# 遍历字典

for key, value in data.items():

ls = str(value)

if begin in ls and end in ls:

print(f'您可以通过{key}号线路直接到达')

return True

except:

break

def save\_data(ls): # 存储数据的函数

with open('line\_info.pkl', 'wb') as fp:

for data in ls:

pickle.dump(data, fp)

def check\_site(begin, end): # 判断站点所在的线路的函数。

with open('line\_info.pkl', 'rb') as fp:

ls = []

ls\_ = []

while True:

try:

data = pickle.load(fp)

# 遍历字典

for key, value in data.items():

ls1 = str(value)

if begin in ls1:

ls.append(key)

if end in ls1:

ls\_.append(key)

except:

break

return ls, ls\_

# def help\_change\_line\_info(i, j):

# with open('line\_info.pkl', 'rb') as fp:

# ls = []

# ls\_ = []

# while True:

# try:

# data = pickle.load(fp)

# # 遍历字典

# for key, value in data.items():

# ls1 = str(key)

# if i in ls1:

# ls.append(value)

# if j in ls1:

# ls\_.append(value)

# except:

# break

# return ls, ls\_

if \_\_name\_\_ == 'main':

main()

main()