实 验 报 告

**实验课程名称　　 Python程序设计**

**专业班级　 数据科学与大数据技术2班**

**学　　号　　　 22022402430**

**学生姓名　　　 　　樊宗豪**

**指导教师　　　　　 张辉辉**

**2023至 2024学年第 二 学期**

**潍坊学院计算机工程学院**

实 验 报 告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目**  **名　　称** | **Python文件读写与模块操作** | | **实验**  **类型** | **演示□　验证□**  **综合□　设计√** | |
| **实验室名称** | | **7325** | **实验日期** | | **2023.05.21** |

**一、实验目的**

1．会处理事件；

2. 利用pyinstaller工具打包python程序；

3．会做模块整合。

**二、实验仪器设备**

一台配置好Python环境的PC机

PyCharm

**三、实验内容（步骤）**

[实验题目]

采用读写文件的方式模拟一个用户管理系统，实现用户注册、用户登录、用户注销等功能。具体要求如下：

1.用户信息存储在文件中。

2.注册时判断用户是否已存在。

3.登录时比较用户的登录信息与文件中存储的信息是否一致。

（不使用正则，用文件读写函数和字符串操作相关知识实现）

4.用户注销

[代码实现]

import os

def welcome():

   # 用户注册 用户登录 退出

   op = [0, register, login, exi]

   while True:

       print("欢迎使用账户管理程序",

             "1.用户注册",

             "2.用户登录",

             "3.退出", sep='\n')

       try:

           option = int(input("请选择功能:"))

           assert 0 < option <= 3

           op[option]()

       except AssertionError: pass

       except ValueError: pass

# 将文件中的数据转换为字典

def convert\_data():

   info\_li = []

   with open('./info.txt', mode='r+', encoding='utf8') as f:

       info\_data = f.readlines()

       for i in info\_data:

           info\_dict = dict()

           # 替换{ 和 } 并去掉空格

           step\_one = i.replace('{', '').replace('}', '')

           # 以冒号进行分隔

           step\_two = step\_one.split(':')

           # 拼接字典

           info\_dict["姓名"] = step\_two[1].split(',')[0].replace("'", '').strip()

           info\_dict["密码"] = step\_two[2].replace("'", '').strip()

           # 保存到列表中

           info\_li.append(info\_dict)

   return info\_li

# 注册

def register():

   if not os.path.exists('./info.txt'):

       with open('./info.txt', mode='w', encoding='utf8') as f:

           f.write('')

   # 用户名列表

   name\_li = []

   info\_li = convert\_data()

   # 接收注册信息

   person\_info = {}

   name = input("请输入注册用户名：\n")

   # 获取用户列名列表

   for i in info\_li:

       name\_li.append(i['姓名'])

   # 判断用户是否存在

   if name in name\_li:

       print('用户已注册')

   else:

       password = input("请输入注册密码：\n")

       person\_info['姓名'] = name

       person\_info['密码'] = password

       # 写入注册信息

       with open('./info.txt', mode='a+', encoding='utf8') as info\_data:

           info\_data.write(str(person\_info) + '\n')

# 登录

def login():

   if os.path.exists('./info.txt') is not True:

       print('当前无数据，请先注册')

   else:

       # 用户名列表

       name\_li = []

       info\_li = convert\_data()

       name = input("请输入登录用户名：\n")

       password = input("请输入登录密码：\n")

       # 获取用户列名列表

       for i in info\_li:

           name\_li.append(i['姓名'])

       # 判断用户是否存在

       if name in name\_li:

           # 获取修改用户的索引

           modify\_index = name\_li.index(name)

           # 判断密码是否正确

           if password == info\_li[modify\_index]['密码']:

               print('登录成功')

           else:

               print('用户名或密码不正确')

       else:

           print('用户名或密码不正确')

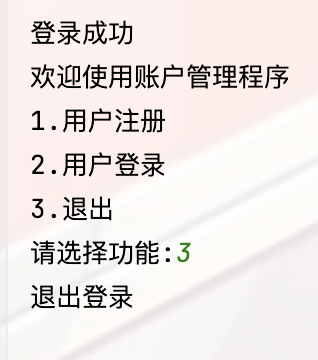
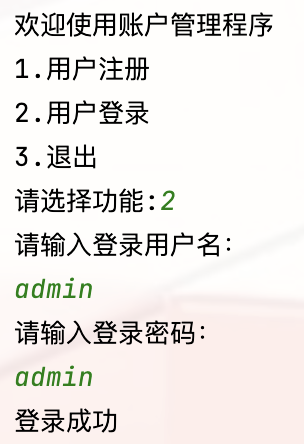
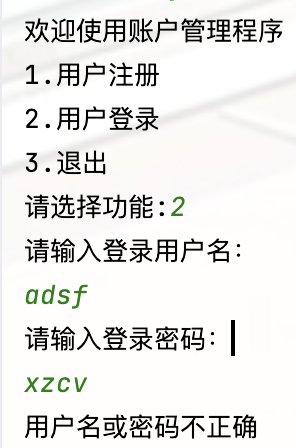
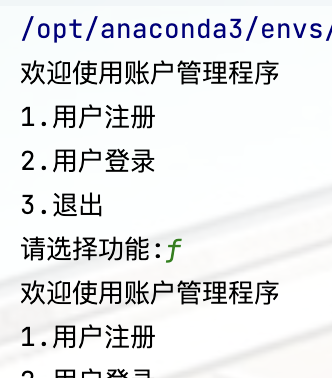
# 退出

def exi(): print('退出登录'); exit(0)

if \_\_name\_\_ == '\_\_main\_\_':

   welcome()

**四、实验数据记录**



**五、实验体会、收获及及建议**

实验体会：在这次设计类实验中，我有机会深入学习Python文件读写操作，以及如何使用pyinstaller工具打包Python程序。通过模拟用户管理系统，我不仅掌握了如何存储和管理用户信息，还学会了如何处理用户注册、登录和注销的事件。这个实验让我对模块整合有了更加深刻的理解，并且提高了我的编程能力。

收获：我的主要收获是对文件操作和模块化编程的实际应用。在实现用户管理系统的过程中，我学会了如何创建和维护一个文件数据库，以及如何通过文件读写和字符串操作来验证用户信息。这个过程不仅锻炼了我的编程技能，也加深了我对事件处理和模块化设计的理解。

建议：对于这类设计类实验，我建议同学们在编程时多关注代码的结构和可读性。例如，我们可以尝试将代码分割成不同的模块和函数，这样不仅有助于代码的维护，也使得功能的扩展更加容易。此外，我认为我们应该多利用Python的内置库和工具，如pyinstaller，这样可以提高我们的开发效率。最后，我觉得我们可以在课余时间进行小型项目，比如开发一个简单的应用或游戏，这将有助于我们更好地理解和应用所学的知识。

**六、指导教师评分**

**成绩： 　　　　签名（电子）：　　 日期：**