

信息工程学院 2023-2024 第一学期 集中实践报告

——《课程设计 I》

题目：图书管理系统开发文档

学生姓名：_____何炜_____

学生学号：_____2022010611_____

学生姓名：_____和广洋_____

学生学号：_____2022060407_____

学生姓名：_____杜志成_____

学生学号：_____2022010610_____

所在班级：_____计科 226_____

所学专业：_____计算机科学与技术_____

完成时间：_____2023-12_____

图书管理系统

摘 要

这项目实现了一个图书馆管理系统，分为用户和管理员两个登录模式。用户能够浏览图书信息、借阅和归还图书，而管理员具备对图书信息的增、删、改、查等管理权限。系统采用 CSV 文件存储用户和图书信息，通过 Python 编写核心逻辑，包括用户注册、登录，图书管理和相应交互功能。通过命令行界面，用户可以方便地执行各项操作，实现了一个基本的图书馆管理系统。

关键词： 图书馆管理系统、用户注册、用户登录、管理员权限、图书信息、CSV 文件、图书借阅、图书归还、命令行界面、Python 编写。

目录

摘 要	2
一 需求分析	3
1.1 系统目标	4
1.2 用户类型	4
1.3 功能需求	4
1.4 文件存储	4
二 系统功能结构	4
2.1 文件架构	5
2.2 模块说明	6
2.2.1 main 模块	6
三 系统实现	6
3.1 模块说明	7
3.1.1 book 模块	7
3.1.2 login 模块	8
四 系统测试	9
登录界面	10
普通用户界面	10
管理员界面	11
五 结论	11
附录:	13

一 需求分析

1.1 系统目标

该系统的主要目标是实现一个简单的图书馆管理系统，包括用户登录、图书管理（列出、增加、删除、更新、借阅和归还书籍）、管理员权限等功能。

1.2 用户类型

系统支持两种用户类型：普通用户（读者）和管理员。普通用户可以浏览图书、借阅和归还书籍，而管理员则具有对图书进行增加、删除、更新等管理权限。

1.3 功能需求

系统需要实现以下主要功能：

用户登录和注册

列出所有书籍信息

借阅和归还书籍

管理员权限下的图书管理（新增、删除、修改）

1.4 文件存储

在系统的实现中，我们选择使用 CSV 文件作为用户信息和图书信息的存储载体。具体而言，用户信息存储在名为 `users.csv` 的文件中，而图书信息则存储在名为 `books.csv` 的文件中。这种文件存储方式具有一定的优势和方便之处。

users.csv 文件：

文件用途：存储用户的基本信息，包括用户名、密码、姓名、班级名和电话号码。

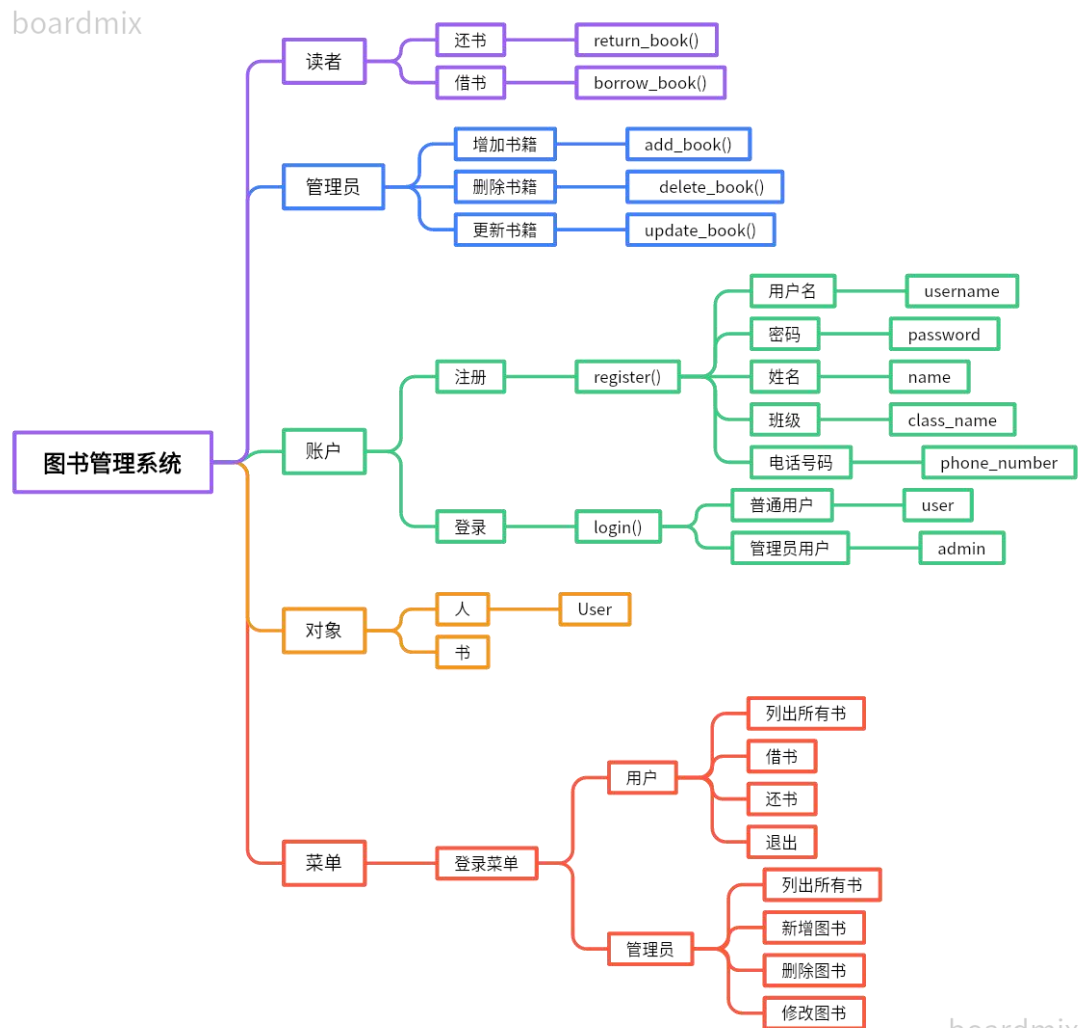
数据结构：每一行代表一个用户的信息，不同字段使用逗号分隔。文件第一行为表头，包括"UID"、"PSK"、"Name"、"cid"和"Phone"等字段。

books.csv 文件：

文件用途：存储图书馆中书籍的基本信息，包括书名、作者和数量。

数据结构：每一行代表一本书的信息，不同字段使用逗号分隔。文件第一行为表头，包括"Title"、"Author"和"Num"等字段。

二 系统功能结构



2.1 文件架构

系统的文件架构包括以下文件：

main.py: 主程序模块

login.py: 用户登录和注册模块

book.py: 图书管理模块

users.csv: 存储用户信息的 CSV 文件

books.csv: 存储图书信息的 CSV 文件

2.2 模块说明

2.2.1 main 模块

(1) check_file_exists()

```
1.     def check_file_exists():
2.         user_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)),
    'users.csv')
3.         books_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)),
    'books.csv')
4.         if not os.path.isfile(user_csv_path):
5.             with open(user_csv_path, 'w', newline='') as f:
6.                 writer = csv.writer(f)
7.                 writer.writerow(["UID", "PSK", "Name", "cid", "Phone"])
8.         if not os.path.isfile(books_csv_path):
9.             with open(books_csv_path, 'w', newline='') as f:
10.                 writer = csv.writer(f)
11.                 writer.writerow(["Title", "Author", "Num"])
```

检查用户信息和图书信息的 CSV 文件是否存在，如果不存在，则创建对应的 CSV 文件并写入表头。

三 系统实现

3.1 模块说明

3.1.1 book 模块

(1) list_books():

```
1.     def list_books():
2.         with open(books_csv_path, 'r') as f:
3.             reader = csv.reader(f)
4.             for row in reader:
5.                 print(row)
```

打开图书 CSV 文件，以只读方式读取并打印所有书籍的信息。

(2) add_book():

```
1.     def add_book():
2.         book_name = input("请输入书名: ")
3.         book_author = input("请输入作者: ")
4.         book_quantity = input("请输入数量: ")
5.         with open(books_csv_path, 'a', newline='') as f:
6.             writer = csv.writer(f)
7.             writer.writerow([book_name, book_author, book_quantity])
```

用户输入书名、作者和数量，将这些信息追加到图书 CSV 文件中。

(3) delete_book():

```
1.     def delete_book():
2.         book_name = input("请输入要删除的书名:")
3.         lines = []
4.         with open(books_csv_path, 'r') as f:
5.             reader = csv.reader(f)
6.             lines = [row for row in reader if row[0] != book_name]
7.         with open(books_csv_path, 'w', newline='') as f:
8.             writer = csv.writer(f)
9.             writer.writerows(lines)
```

用户输入要删除的书名，删除图书 CSV 文件中对应的书籍信息。

(4) update_book():

```
1.     def update_book():
2.         book_name = input("请输入要更新的书名:")
3.         lines = []
4.         with open(books_csv_path, 'r') as f:
5.             reader = csv.reader(f)
6.             lines = [row for row in reader if row[0] != book_name]
```

```

7.     book_author = input("请输入新的作者:")
8.     book_quantity = input("请输入新的数量:")
9.     lines.append([book_name, book_author, book_quantity])
10.    with open(books_csv_path, 'w', newline='') as f:
11.        writer = csv.writer(f)
12.        writer.writerows(lines)

```

用户输入要更新的书名，然后输入新的作者和数量，更新图书 CSV 文件中对应书籍的信息。

(5) borrow_book():

```

1.     def borrow_book():
2.         book_name = input("请输入要借阅的书名:")
3.         lines = []
4.         with open(books_csv_path, 'r') as f:
5.             reader = csv.reader(f)
6.             lines = [row for row in reader]
7.             for row in lines:
8.                 if row[0] == book_name:
9.                     if int(row[2]) > 0:
10.                        row[2] = str(int(row[2]) - 1)
11.                        print("借书成功!")
12.                    else:
13.                        print("这本书已经被借完了。")
14.                        break
15.         with open(books_csv_path, 'w', newline='') as f:
16.             writer = csv.writer(f)
17.             writer.writerows(lines)

```

用户输入要借阅的书名，从图书 CSV 文件中找到对应书籍，如果数量大于 0，则借阅成功并减少数量；否则，提示该书已被借完。

3.1.2 login 模块

(1) register():

- 获取用户输入的用户名、密码、姓名、班级和电话号码。
- 创建新用户对象，并将其信息追加到用户 CSV 文件中。
- 返回新用户对象。

```

1.     new_user = User(username, password, name, class_name, phone_number)
2.     with open(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv'), 'a', newline='') as f:
3.         writer = csv.writer(f)

```



```
4.         writer.writerow([new_user.username, new_user.password, new_user.name, new_user.class_name, new_user.phone_number])
5.         return new_user
```

(2) login():

- 获取用户输入的用户名和密码。
- 如果用户名为 'admin' 且密码为 '123456', 返回 'admin'。
- 打开用户 CSV 文件, 遍历文件内容, 如果输入的用户名和密码正确,

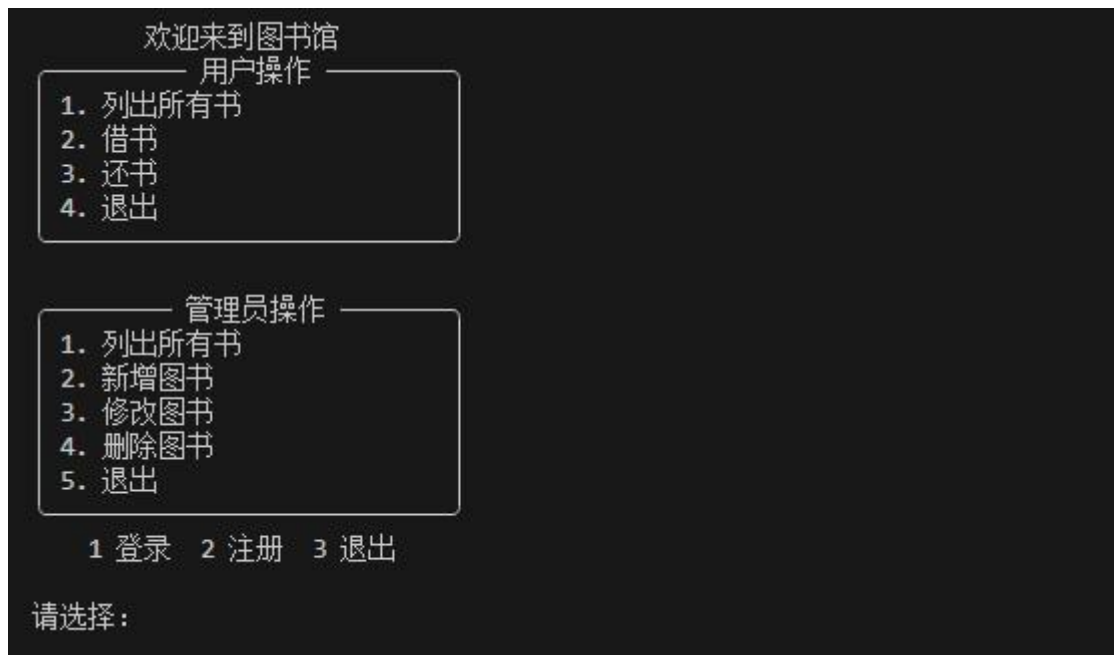
返回 'user'。

```
1.         with open(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv'), 'r') as f:
2.             reader = csv.reader(f)
3.             for row in reader:
4.                 if row[0] == username and row[1] == password:
5.                     return 'user'
```

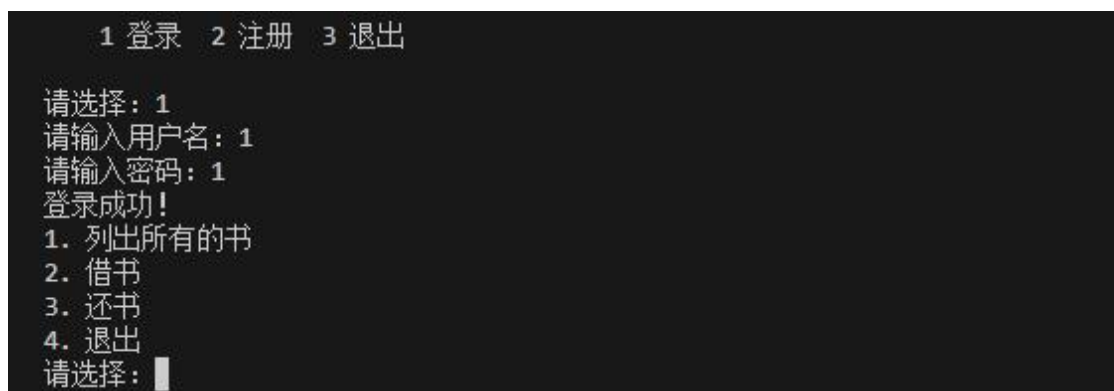
- 如果用户名和密码不正确, 返回 None。

四 系统测试

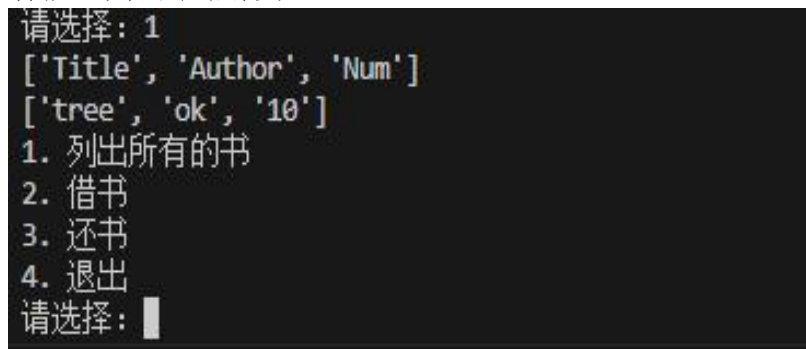
登录界面



普通用户界面



功能 1 列出书籍展示



功能 2 借书展示

```

请选择：2
请输入要借阅的书名：tree
借书成功！
1. 列出所有的书
2. 借书
3. 还书
4. 退出
请选择：1
['Title', 'Author', 'Num']
['tree', 'ok', '9']
1. 列出所有的书
2. 借书
3. 还书
4. 退出
请选择：

```

管理员界面

```

1 登录 2 注册 3 退出

请选择：1
请输入用户名：admin
请输入密码：123456
管理员登录成功！
1. 列出所有的书
2. 新增图书
3. 修改图书
4. 删除图书
5. 退出
请选择：

```

功能 2 新增图书展示

```

请选择：2
请输入书名：tree
请输入作者：ok
请输入数量：41
1. 列出所有的书
2. 新增图书
3. 修改图书
4. 删除图书
5. 退出
请选择：1
['Title', 'Author', 'Num']
['tree', 'ok', '9']
['tree', 'ok', '41']

```

五 结论

在完成本次任务中，我们成功地设计并实现了一个简单的图书管理系统，该系统包括用户注册、登录以及图书的增加、删除、更新、借阅和归还等功能。通过这个项目，我们深入理解了 Python 中文件操作、类的使用以及 CSV 文件的读写等方面的知识。我们学会了如何组织和管理代码，提高了问题解决能力和编码实践经验。

在已实现功能方面，系统能够准确处理用户的注册和登录请求，保证用户信息的安全性。图书管理方面，系统支持书籍的增加、删除、更新以及借阅和归还等基本操作，有效管理图书馆的图书信息。

然而，系统仍存在一些需要改进和完善的地方。例如，对于输入的合法性和错误处理方面可以加强，系统还可以添加更多的辅助功能，提升用户体验。未来，我计划进一步优化系统代码，提高系统的稳定性和可靠性。通过这个项目，我不仅学到了具体的技术知识，也提高了对软件开发流程的理解和把控能力。

附录:

全部代码

main.py:

```
1. import login
2. import book as book
3. import os
4. import csv
5. def check_file_exists():
6.     user_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv')
7.     books_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'books.csv')
8.     if not os.path.isfile(user_csv_path):
9.         with open(user_csv_path, 'w', newline='') as f:
10.             writer = csv.writer(f)
11.             writer.writerow(["UID", "PSK", "Name", "cid", "Phone"])
12.     if not os.path.isfile(books_csv_path):
13.         with open(books_csv_path, 'w', newline='') as f:
14.             writer = csv.writer(f)
15.             writer.writerow(["Title", "Author", "Num"])
16. def main():
17.     check_file_exists()
18.     while True:
19.         print(
20.             """
21.             欢迎来到图书馆
22.             ┌────────── 用户操作 ─────────┐
23.             | 1. 列出所有书                |
24.             | 2. 借书                      |
25.             | 3. 还书                      |
26.             | 4. 退出                      |
27.             └──────────────────────────┘
28.
29.             ┌────────── 管理员操作 ─────────┐
30.             | 1. 列出所有书                |
31.             | 2. 新增图书                  |
32.             | 3. 修改图书                  |
33.             | 4. 删除图书                  |
```

```

34. | 5. 退出 |
35. |-----|
36. 1 登录 2 注册 3 退出
37. ""
38. )
39.     choice = input("请选择: ")
40.     if choice == '1':
41.         user_type = login.User.login()
42.         if user_type == 'user':
43.             print("登录成功!")
44.             while True:
45.                 print("1. 列出所有的书")
46.                 print("2. 借书")
47.                 print("3. 还书")
48.                 print("4. 退出")
49.                 choice = input("请选择: ")
50.                 if choice == '1':
51.                     book.list_books()
52.                 elif choice == '2':
53.                     book.borrow_book()
54.                 elif choice == '3':
55.                     book.return_book()
56.                 elif choice == '4':
57.                     break
58.             elif user_type == 'admin':
59.                 print("管理员登录成功!")
60.                 while True:
61.                     print("1. 列出所有的书")
62.                     print("2. 新增图书")
63.                     print("3. 修改图书")
64.                     print("4. 删除图书")
65.                     print("5. 退出")
66.                     choice = input("请选择: ")
67.                     if choice == '1':
68.                         book.list_books()
69.                     elif choice == '2':
70.                         book.add_book()
71.                     elif choice == '3':
72.                         book.update_book()
73.                     elif choice == '4':
74.                         book.delete_book()
75.                     elif choice == '5':

```

```

76.                 break
77.             else:
78.                 print("登录失败！")
79.             elif choice == '2':
80.                 user = login.User.register()
81.             elif choice == '3':
82.                 break
83.
84. if __name__ == "__main__":
85.     main()

```

login.py:

```

1. import csv
2. import os
3. class User:
4.     def __init__(self, username, password, name, class_name, phone_number):
5.         self.username = username
6.         self.password = password
7.         self.name = name
8.         self.class_name = class_name
9.         self.phone_number = phone_number
10.    def register():
11.        username = input("请输入用户名: ")
12.        password = input("请输入密码: ")
13.        name = input("请输入姓名: ")
14.        class_name = input("请输入班级: ")
15.        phone_number = input("请输入电话号码: ")
16.        new_user = User(username, password, name, class_name, phone_number)
17.        with open(os.path.join(os.path.dirname(os.path.abspath(__file__)), 'users.csv'), 'a', newline='') as f:
18.            writer = csv.writer(f)
19.            writer.writerow([new_user.username, new_user.password, new_user.name, new_user.class_name, new_user.phone_number])
20.        return new_user
21.    def login():
22.        username = input("请输入用户名: ")
23.        password = input("请输入密码: ")
24.        if username == 'admin' and password == '123456':
25.            return 'admin'

```

```

26.         with open(os.path.join(os.path.dirname(os.path.abspath(__f
ile__)), 'users.csv'), 'r') as f:
27.             reader = csv.reader(f)
28.             for row in reader:
29.                 if row[0] == username and row[1] == password:
30.                     return 'user'
31.             return None

```

Book.py:

```

1. import csv
2. import os
3. books_csv_path = os.path.join(os.path.dirname(os.path.abspath(__file__
), 'books.csv')
4. def list_books():
5.     with open(books_csv_path, 'r') as f:
6.         reader = csv.reader(f)
7.         for row in reader:
8.             print(row)
9. def add_book():
10.     book_name = input("请输入书名: ")
11.     book_author = input("请输入作者: ")
12.     book_quantity = input("请输入数量: ")
13.     with open(books_csv_path, 'a', newline='') as f:
14.         writer = csv.writer(f)
15.         writer.writerow([book_name, book_author, book_quantity])
16. def delete_book():
17.     book_name = input("请输入要删除的书名: ")
18.     lines = []
19.     # 以只读的方式打开 books_csv_path 文件，以读取的方式
20.     with open(books_csv_path, 'r') as f:
21.         reader = csv.reader(f)
22.         lines = [row for row in reader if row[0] != book_name]
23.     with open(books_csv_path, 'w', newline='') as f:
24.         writer = csv.writer(f)
25.         writer.writerows(lines)
26. def update_book():
27.     book_name = input("请输入要更新的书名: ")
28.     lines = []
29.     with open(books_csv_path, 'r') as f:
30.         reader = csv.reader(f)
31.         lines = [row for row in reader if row[0] != book_name]
32.     book_author = input("请输入新的作者: ")

```



```

33.     book_quantity = input("请输入新的数量: ")
34.     lines.append([book_name, book_author, book_quantity])
35.     with open(books_csv_path, 'w', newline='') as f:
36.         writer = csv.writer(f)
37.         writer.writerows(lines)
38.     def borrow_book():
39.         book_name = input("请输入要借阅的书名: ")
40.         lines = []
41.         with open(books_csv_path, 'r') as f:
42.             reader = csv.reader(f)
43.             lines = [row for row in reader]
44.         for row in lines:
45.             if row[0] == book_name:
46.                 if int(row[2]) > 0:
47.                     row[2] = str(int(row[2]) - 1)
48.                     print("借书成功! ")
49.                 else:
50.                     print("这本书已经被借完了。")
51.                 break
52.         with open(books_csv_path, 'w', newline='') as f:
53.             writer = csv.writer(f)
54.             writer.writerows(lines)
55.     def return_book():
56.         book_name = input("请输入要归还的书名: ")
57.         lines = []
58.         with open(books_csv_path, 'r') as f:
59.             reader = csv.reader(f)
60.             lines = [row for row in reader]
61.         for row in lines:
62.             if row[0] == book_name:
63.                 row[2] = str(int(row[2]) + 1)
64.                 print("还书成功! ")
65.                 break
66.         with open(books_csv_path, 'w', newline='') as f:
67.             writer = csv.writer(f)
68.             writer.writerows(lines)

```