

**课程 实 验 报 告**

**课程名称： Python程序设计**

**专业班级： 自实1901**

**学 号： U201916457**

**姓 名： 张皓然**

**指导教师： 卢仁智**

**报告日期： 2020年11月7日**

**人工智能与自动化学院**

# 题目2

## 1题目

用Python设计一个简易的图书管理系统，功能包括但不限于以下：

1. 查询图书

2. 增加图书

3. 借阅图书

4. 归还图书

5. 退出系统

## 2设计思路与代码实现

### 2.1制作流程

1.设计图形界面

2.设计简单的算法

3.完成代码编写

## 3程序介绍

本程序设计了多个界面，来分别实现题目要求的不同功能。

|  |
| --- |
|  |
| 图1. 图书管理系统的主界面 |

在主界面中，我们可以通过左侧五个按钮，选择不同的功能。选择“退出系统”则会直接结束程序。

|  |
| --- |
|  |
| 图2. 搜索图书界面 |

在书界面点击搜索图书，我们便会进入搜索界面（Search System）。该界面支持三种搜索模式，分别为直接搜索图书，搜索作者，以及搜索类别。以下为这三种模式的介绍。

|  |
| --- |
|  |
| 图3. 搜索图书功能 |

下方文本框显示的信息分别是：书名，ISBN码，作者，类别，馆藏数量。

|  |
| --- |
|  |
| 图4. 搜索作者的功能 |

搜索作者可以检索出图书馆内该作者的所有作品信息，非常方便。

|  |
| --- |
|  |
| 图5. 搜索类别的功能 |

搜索类别的功能可以检索出图书馆内该类别的所有作品信息，非常非常方便。

此外，我们可以在书界面点击添加新书，进入添加新书的界面。

|  |
| --- |
|  |
| 图6. 添加新书的界面 |

如果新书的ISBN码在图书馆中检索不到，则添加新的一行。如果ISBN码在图书馆中已有，则对应馆藏书目的数目加1.

|  |
| --- |
|  |
| 图7. 添加一本传世巨作 |

随后，我们可以在查找图书的界面中找到这本书。

|  |
| --- |
|  |
| 图8. 在搜索界面中找到刚刚添加的图书 |

如果我们此时再返回到添加图书的界面，再添加一次该图书，如图，

|  |
| --- |
|  |
| 图9. 添加同一本书 |

此时系统会将相同的书叠加在一起，我们可以在搜索界面中查询。

|  |
| --- |
|  |
| 图10. 检查添加的情况 |

此外，本系统支持借阅图书功能和归还图书功能，这两个功能的实现基本相同。借阅成功后，馆藏书目减一。归还成功后，馆藏书目加一。

|  |
| --- |
|  |
| 图11. 借阅图书的展示 |

|  |
| --- |
|  |
| 图12. 归还图书的展示 |

## 实验结果

程序功能丰富，满足题目要求。

## 感悟

为更好的完成最后一次Python的作业，我在网上自学了Python的Tkinter库，惊喜的发现Python设计界面非常简单。恰好C语言课程设计刚刚结束，与C语言相比，Python确实更加简单易用，学习成本更低。一个C语言的程序，我们做了几个月，但如果用Python实现相同的功能，可能一个月就可以完成。在计算机算力逐渐增强的今天，Python解放了我们研究编程语言本身的时间，不必再花时间去思考复杂的指针，从而让我们把更多的时间花费在思考算法，梳理逻辑的上面，这确实是功德一件。

## 附录（代码）

1. #  Library Manager
2. #  function\_1: search for a book
3. #  function\_2: append a new book
4. #  function\_3: borrow a book
5. #  function\_4: return the book
6. #  function\_5: exit the system
7. #  struct: Cur\_book
8. #  1. book name
9. #  2. ISBN code
10. #  3. author
11. #  4. class of the book
12. #  5. current sum
13. from tkinter import \*
14. import pandas as pd

17. def quit(root):
18. root.destroy()

21. def open\_library():
22. data = pd.read\_csv('book.csv')
23. bookList = data.values.tolist()
24. **return** bookList

27. def search\_sys():
28. search = Tk()
29. search.title('Search System')
30. search.geometry('400x400')
32. txt = Text(search)
33. txt.place(relx=0.05, rely=0.6, relwidth=0.9, relheight=0.3)
34. txt.insert(END, '搜索图书（书名，ISBN码均可）\n搜索作者可找到该作者所有书籍\n搜索类别（小说，诗歌，散文，历史，传记）')
35. inp = Entry(search)
36. inp.place(relx=0.25, rely=0.1, width=200)
38. btn1 = Button(search, text='搜索图书', command=lambda: f\_search\_one(inp, txt))
39. btn1.place(relx=0.40, rely=0.2, relwidth=0.20, relheight=0.1)
41. btn2 = Button(
42. search,
43. text='搜索作者',
44. command=lambda: f\_search\_author(
45. inp,
46. txt))
47. btn2.place(relx=0.40, rely=0.3, relwidth=0.20, relheight=0.1)
49. btn3 = Button(
50. search,
51. text='搜索类别',
52. command=lambda: f\_search\_class(
53. inp,
54. txt))
55. btn3.place(relx=0.40, rely=0.4, relwidth=0.20, relheight=0.1)

58. def f\_search\_one(inp, txt):
59. aim\_book = inp.get()
60. txt.**delete**(1.0, 10.0)
61. **if** len(aim\_book) == 0:  # 如果未输入
62. txt.insert(END, '未找到！')
63. **return** 0
64. result = []
65. library = open\_library()
67. **for** item in library:
68. item[1] = str(item[1])  # ISBN码一开始并不是字符串，改为字符串，方便比较
69. **if** aim\_book in [item[0], item[1]]:
70. result.append(item)
71. **break**
72. **if** len(result) == 0:  # 如果未找到结果
73. txt.insert(END, '未找到！')
74. **return** 0
75. txt.insert(END, result)
76. inp.**delete**(0, END)

79. def f\_search\_author(inp, txt):  # 搜索作者
80. aim\_book = inp.get()
81. txt.**delete**(1.0, 10.0)
82. **if** len(aim\_book) == 0:  # 如果未输入
83. txt.insert(END, '未找到！')
84. **return** 0
85. result = []
87. library = open\_library()
89. **for** item in library:
90. **if** aim\_book in item[2]:
91. result.append(item)
93. **if** len(result) == 0:  # 如果未找到结果
94. txt.insert(END, '未找到！')
95. **return** 0
97. **for** item in result:
98. txt.insert(END, item)
99. txt.insert(END, '\n')
101. inp.**delete**(0, END)

104. def f\_search\_class(inp, txt):
105. aim\_book = inp.get()
106. txt.**delete**(1.0, 10.0)
107. **if** len(aim\_book) == 0:  # 如果未输入
108. txt.insert(END, '未找到！')
109. **return** 0
110. result = []
112. library = open\_library()
114. **for** item in library:
115. **if** aim\_book in item[3]:
116. result.append(item)
118. **if** len(result) == 0:  # 如果未找到结果
119. txt.insert(END, '未找到！')
120. **return** 0
122. **for** item in result:
123. txt.insert(END, item)
124. txt.insert(END, '\n')
126. inp.**delete**(0, END)

129. def append\_sys():
130. append = Tk()
131. append.title('Append System')
132. append.geometry('400x400')
134. txt = Text(append)
135. txt.place(relx=0.05, rely=0.6, relwidth=0.9, relheight=0.3)
136. txt.insert(END, '添加图书，需写明书名，ISBN码，作者，分类\n若馆藏内已有此书，则增加馆藏数量')
138. lb1 = Label(append, text='书名')
139. lb2 = Label(append, text='ISBN')
140. lb3 = Label(append, text='作者')
141. lb4 = Label(append, text='分类')
142. lb5 = Label(append, text='添加数目')
144. lb5.place(relx=0.1, rely=0.325 + 0.055, relwidth=0.15, relheight=0.075)
145. lb4.place(relx=0.1, rely=0.25 + 0.055, relwidth=0.15, relheight=0.075)
146. lb3.place(relx=0.1, rely=0.175 + 0.055, relwidth=0.15, relheight=0.075)
147. lb2.place(relx=0.1, rely=0.1 + 0.055, relwidth=0.15, relheight=0.075)
148. lb1.place(relx=0.1, rely=0.025 + 0.055, relwidth=0.15, relheight=0.075)
150. inp1 = Entry(append)
151. inp1.place(relx=0.25, rely=0.1, width=200)
153. inp2 = Entry(append)
154. inp2.place(relx=0.25, rely=0.175, width=200)
156. inp3 = Entry(append)
157. inp3.place(relx=0.25, rely=0.25, width=200)
159. inp4 = Entry(append)
160. inp4.place(relx=0.25, rely=0.325, width=200)
162. inp5 = Entry(append)
163. inp5.place(relx=0.25, rely=0.4, width=200)
165. btn1 = Button(
166. append,
167. text='添加图书',
168. command=lambda: append\_book(
169. inp1,
170. inp2,
171. inp3,
172. inp4,
173. inp5,
174. txt))
175. btn1.place(relx=0.40, rely=0.5, relwidth=0.20, relheight=0.1)

178. def append\_book(inp1, inp2, inp3, inp4, inp5, txt):
179. book\_name = str(inp1.get())
180. ISBN = str(inp2.get())
181. author = inp3.get()
182. Class = inp4.get()
183. sum = **int**(inp5.get())
185. library = open\_library()
186. row = 0
187. data = library
189. **for** item in library:
190. **if** ISBN in str(item[1]):
191. sum\_old = data[row][4]
192. del data[row]  # 删除重复行的数据
193. sum += sum\_old
194. row += 1
195. sum = str(sum)
196. data.append([book\_name, ISBN, author, Class, sum])  # 更新数据
198. book\_list = []
199. ISBN\_list = []
200. author\_list = []
201. Class\_list = []
202. sum\_list = []
204. **for** item in data:
205. book\_list.append(str(item[0]))
206. ISBN\_list.append(str(item[1]))
207. author\_list.append(str(item[2]))
208. Class\_list.append(str(item[3]))
209. sum\_list.append(str(item[4]))
211. new\_data = {
212. 'bookname': book\_list,
213. 'ISBN': ISBN\_list,
214. 'author': author\_list,
215. 'class': Class\_list,
216. 'sum': sum\_list}
217. df = pd.DataFrame(new\_data)
218. df.to\_csv('book.csv', index=False, sep=',')
219. txt.**delete**(0.0, 8.0)
220. txt.insert(END, '添加成功')
222. inp1.**delete**(0, END)
223. inp2.**delete**(0, END)
224. inp3.**delete**(0, END)
225. inp4.**delete**(0, END)
226. inp5.**delete**(0, END)

229. def borrow\_sys():
230. borrow = Tk()
231. borrow.title('Borrow System')
232. borrow.geometry('400x400')
234. lb1 = Label(borrow, text= '书名')
235. lb1.place(relx=0.1, rely=0.175, relwidth=0.15, relheight=0.1)
237. txt = Text(borrow)
238. txt.place(relx=0.05, rely=0.6, relwidth=0.9, relheight=0.3)
239. txt.insert(END, '借阅图书')
241. inp1 = Entry(borrow)
242. inp1.place(relx=0.25, rely=0.2, width=200)
244. btn1 = Button(
245. borrow,
246. text='借阅图书',
247. command=lambda: borrow\_book(inp1, txt))
248. btn1.place(relx=0.40, rely=0.5, relwidth=0.20, relheight=0.1)

251. def borrow\_book(inp1, txt):
252. txt.**delete**(0.0, 9.0)
253. book\_name = str(inp1.get())
255. library = open\_library()
256. row = 0
257. data = library
259. mark = 0
261. **for** item in library:
262. **if** book\_name in str(item[0]):
263. mark = 1
264. result = item
265. sum\_old = data[row][4]
266. del data[row]  # 删除原始的数据
267. sum = sum\_old - 1
268. **if** sum < 0:
269. txt.insert(END, '借阅失败')
270. **return** 0
271. **break**
272. row += 1
274. **if** mark == 0:
275. txt.insert(END, '借阅失败')
276. **return** 0
278. sum = str(sum)
279. data.append([book\_name, str(result[1]), str(result[2]), str(result[3]), sum])  # 更新数据
281. book\_list = []
282. ISBN\_list = []
283. author\_list = []
284. Class\_list = []
285. sum\_list = []
287. **for** item in data:
288. book\_list.append(str(item[0]))
289. ISBN\_list.append(str(item[1]))
290. author\_list.append(str(item[2]))
291. Class\_list.append(str(item[3]))
292. sum\_list.append(str(item[4]))
294. new\_data = {
295. 'bookname': book\_list,
296. 'ISBN': ISBN\_list,
297. 'author': author\_list,
298. 'class': Class\_list,
299. 'sum': sum\_list}
300. df = pd.DataFrame(new\_data)
301. df.to\_csv('book.csv', index=False, sep=',')
302. txt.**delete**(0.0, 8.0)
303. txt.insert(END, '借阅成功')
305. inp1.**delete**(0, END)

308. def return\_sys():
309. ret = Tk()
310. ret.title('Return System')
311. ret.geometry('400x400')
313. lb1 = Label(ret, text='书名')
314. lb1.place(relx=0.1, rely=0.175, relwidth=0.15, relheight=0.1)
316. txt = Text(ret)
317. txt.place(relx=0.05, rely=0.6, relwidth=0.9, relheight=0.3)
318. txt.insert(END, '归还图书')
320. inp1 = Entry(ret)
321. inp1.place(relx=0.25, rely=0.2, width=200)
323. btn1 = Button(
324. ret,
325. text='归还图书',
326. command=lambda: return\_book(inp1, txt))
327. btn1.place(relx=0.40, rely=0.5, relwidth=0.20, relheight=0.1)

330. def return\_book(inp1, txt):
331. txt.**delete**(0.0, 9.0)
332. book\_name = str(inp1.get())
334. library = open\_library()
335. row = 0
336. data = library
338. mark = 0
340. **for** item in library:
341. **if** book\_name in str(item[0]):
342. mark = 1
343. result = item
344. sum\_old = data[row][4]
345. del data[row]  # 删除原始的数据
346. sum = sum\_old + 1
347. **break**
348. row += 1
350. **if** mark == 0:
351. txt.insert(END, '归还失败')
352. **return** 0
354. sum = str(sum)
355. data.append([book\_name, str(result[1]), str(result[2]), str(result[3]), sum])  # 更新数据
357. book\_list = []
358. ISBN\_list = []
359. author\_list = []
360. Class\_list = []
361. sum\_list = []
363. **for** item in data:
364. book\_list.append(str(item[0]))
365. ISBN\_list.append(str(item[1]))
366. author\_list.append(str(item[2]))
367. Class\_list.append(str(item[3]))
368. sum\_list.append(str(item[4]))
370. new\_data = {
371. 'bookname': book\_list,
372. 'ISBN': ISBN\_list,
373. 'author': author\_list,
374. 'class': Class\_list,
375. 'sum': sum\_list}
376. df = pd.DataFrame(new\_data)
377. df.to\_csv('book.csv', index=False, sep=',')
378. txt.**delete**(0.0, 8.0)
379. txt.insert(END, '归还成功')
381. inp1.**delete**(0, END)

384. def main\_sys():
385. root = Tk()
386. root.title('Library Manager')
387. root.geometry('400x400')
389. lb1 = Label(root, text='designed by Zhang Haoran')
390. lb1.place(relx=0, rely=0.85, relwidth=0.8, relheight=0.2)
392. lb2 = Label(root, text='图书管理系统', font=('华文新宋', 20))
393. lb2.place(relx=0.25, rely=0.3, relwidth=0.8, relheight=0.3)
395. btn1 = Button(root, text='搜索图书', command=search\_sys)
396. btn1.place(relx=0, rely=0, relwidth=0.15, relheight=0.2)
398. btn2 = Button(root, text='添加新书', command=append\_sys)
399. btn2.place(relx=0, rely=0.2, relwidth=0.15, relheight=0.2)
401. btn3 = Button(root, text='借阅图书', command=borrow\_sys)
402. btn3.place(relx=0, rely=0.4, relwidth=0.15, relheight=0.2)
404. btn4 = Button(root, text='归还图书', command=return\_sys)
405. btn4.place(relx=0, rely=0.6, relwidth=0.15, relheight=0.2)
407. btn4 = Button(root, text='退出系统', command=lambda r=root: quit(r))
408. btn4.place(relx=0, rely=0.8, relwidth=0.15, relheight=0.2)
409. root.mainloop()

412. main\_sys()