**中国计量大学信息工程学院人工智能系**

**实验报告**

实验课程： python 实验名称： 实验五

班 级： 智能3 学 号： 2300305322

姓 名： 李昊洋 实验日期： 2023/11/29

1. **实验目的**

初步学习pyhton数据库的操作

初步学习python网络应用

**二、实验内容**

1、 使用Python自带的SQLite3数据库创建一个student.db数据库文件，用这个数据库文件完成本班学生信息系统的构建，主要功能有：增加学生信息，删除学生信息，修改学生信息，显示学生信息。学生信息内容可以有学号，姓名，性别，年龄，电话，Python课成绩等等。

2. requests和bs4是最常用的爬虫库。利用requests和bs4，写一个脚本程序，从豆瓣官网爬取电影排名250的名字。浏览器访问地址为<https://movie.douban.com/top250>

3. 如何利用requests和bs4库，用类似的方法，爬取某个城市天气信息。

提示：

可以参考这个爬虫库使用方法的讲解视频

https://www.bilibili.com/video/BV1j4411c7ny

天气来源网站：https://www.tianqi.com/hangzhou/30

参考资料：

https://blog.csdn.net/m0\_47258632/article/details/125982902

注意：如果选择使用lxml解析器但提示lxml相关的错误，需要安装xml，方法是pip install lxml

爬取的结果请保存为电子表格输出。

**三、实验环境**

Windows，python3.8

**实验成绩： 主管教师签名：**

**四、实验数据记录（源程序或算法设计思想）**

T1

1. **import** sqlite3
2. conn = sqlite3.connect('student.db')
3. cur = conn.cursor()
5. **class** operate\_sys:
6. **def** \_\_init\_\_(self):
7. **pass**
8. **def** addStudent(self,name,sex,age,phnum,pyscore,single):
9. **try**:
10. cur.execute("insert into student (name,sex,age,phnum,pyscore,single) values (?,?,?,?,?,?)",(name,sex,age,phnum,pyscore,single))
11. **print**("添加成功")
12. **except**:
13. **print**("输入有误！")
14. **def** delStudent(self,id):
15. **try**:
16. cur.execute("delete from student where id = ?",(id,))
17. **print**("删除成功!")
18. **except**:
19. **print**("删除过程中出现异常，请重新确认！")
20. **def** reviseStudent(self,id,reitem,redata):
21. **try**:
22. cur.execute(f"update student set {reitem} = ? where id = ? ",(redata,id))
23. **print**("修改成功!")
24. **except**:
25. **print**("输入有误！")
27. **def** showStudent(self,id):
28. **try**:
29. cur.execute("select \* from student where id = ?",(id,))
30. a = cur.fetchall()[0]
31. **print**("id:",a[0],"\nname:",a[1],"\nsex:",a[2],"\nage:",a[3],"\nphnum:",a[4],"\npyscore:",a[5],"\nsingle:",a[6],"\n")
32. **except**:
33. **print**("出现错误！")
35. **def** showmanu(self):
36. **print**("输入1————增加学生新信息")
37. **print**("输入2————删除学生信息")
38. **print**("输入3————修改学生信息")
39. **print**("输入4————显示学生信息")
40. **print**("输入0————保存并退出系统")
41. **return** int(input())
43. op = operate\_sys()
44. **if** \_\_name\_\_ == '\_\_main\_\_':
45. **while** True:
47. flag=op.showmanu()
48. **if** flag == 1:
49. **print**("输入学生姓名、性别、年龄、电话、python成绩、单身状况（空格隔开）：")
50. a = input().split()
51. op.addStudent(a[0],a[1],a[2],a[3],a[4],a[5])
52. **elif** flag == 2:
53. **print**("输入要删除的学生id：")
54. a = int(input())
55. op.delStudent(a)
56. **elif** flag == 3:
57. **print**("输入要修改信息的学生的id：")
58. a = int(input())
59. **print**("正在查询该学生信息......")
60. op.showStudent(a)
61. **try**:
62. **print**("输入需要修改的项目名，修改后的值（空格隔开）：")
63. b = input().split()
64. op.reviseStudent(a,b[0],b[1])
65. **except**:
66. **print**("输入格式错误")
67. **elif** flag == 4:
68. **print**("输入需要查询的学生id：")
69. a = int(input())
70. op.showStudent(a)
71. **elif** flag == 0:
72. **print**("退出系统成功")
73. **break**
74. **else**:
75. **print**("输入有误，重新输入！")
76. **continue**
77. conn.commit()
78. conn.close

T2

1. **import** requests
2. **from** bs4 **import** BeautifulSoup
4. **class** Douban:
5. **def** \_\_init\_\_(self):
6. self.URL = "https://movie.douban.com/top250"
7. self.headnum = []
8. **for** num\_i **in** range(0, 251, 25):
9. self.headnum.append(num\_i)
10. self.header = {'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36'}
12. **def** get\_top250(self):
13. **for** start **in** self.headnum:
14. start = str(start)
15. html = requests.get(self.URL, params={'start': start},headers=self.header)
16. soup = BeautifulSoup(html.text, "html.parser")
17. name = soup.select('#content > div > div.article > ol > li > div > div.info > div.hd > a > span:nth-child(1)')
18. **for** name\_i **in** name:
19. **print**(name\_i.get\_text())
21. **if** \_\_name\_\_ == "\_\_main\_\_":
22. db=Douban()
23. db.get\_top250()

T3

1. **import** requests
2. **from** bs4 **import** BeautifulSoup
4. **class** Tianqi:
5. **def** \_\_init\_\_(self):
6. self.URL = "https://www.tianqi.com/hangzhou/30"
8. self.header = {'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36'}
10. **def** getwather(self):
11. html = requests.get(self.URL,headers=self.header)
12. soup = BeautifulSoup(html.text,"html.parser")#
13. daydate = soup.select('body > div.w1100.newday40\_top > div.inleft > ul.weaul > li > a > div.weaul\_q.weaul\_q > span.fl')
14. weatherdate = soup.select('body > div.w1100.newday40\_top > div.inleft > ul.weaul > li > a > div:nth-child(3)')
15. **for** i **in** range(0, 30):
16. **print**(daydate[i].get\_text(), " ", weatherdate[i].get\_text())
18. **if** \_\_name\_\_ == "\_\_main\_\_":
19. tq=Tianqi()
20. tq.getwather()

**五、实验数据分析及结论（程序运行结果及分析）**

T1通过代码实现了对学生数据库的读取写入修改等

T2运行成功爬取了豆瓣top250的电影，是通过爬取每页信息后进入下一页再重复爬取直到爬取完所有元素

T3成功爬取了杭州接下来三十天（包括今天）的天气以及对应的日期并按照日期输出，但目前网站似乎处于不可用状态

**六、实验中的问题及心得体会**

感受到了使用数据库的处理数据强大功能，体会到python爬虫爬取网页的对数据获取的提供的巨大帮助和便利