1. 输入选项容错(给出一系列固定的选项，但是用户输入了其他的数据时也要保证不能出错。而且输入的错误数据可以是多种形式的！)这里要注意，所有输入一个int型的数据的地方都改成直接调用GetIntFromCin(也就是现在的项目里的MainDealFault)方法。(调用方法待会会发到群里，该函数的实现也在项目里，有什么疑问，自己测试解决或者在群里问)。输入double型数据的 cin >> double 不用改变。(这个虽然也会出现同样的问题，但是先放过)
2. 提示信息友好。给用户给出选项的时候一定要将选项分隔开，像下面这样：
3. 添加读者
4. 删除读者
5. …

还有就是有的输出之间需要分隔开来，不要全部粘在一起，很难看，这个自己在调试的时候多注意，按自己的要求添加换行符。

3.不合理输入判断也要正确容错。这个就是说用户输入的数据格式上没有问题，但是是错误的。如果你按照用户输入的这个数据进行操作，可能会导致文件之间的关联被破坏掉(比如说借出去的书籍数量和馆藏数量，总数量之间的相对关系错误，还有其他类似这种的情况也需要进行考虑)。

4.输入数据的确认。修改或增加操作中，用户输入一个数据后要再向用户展示一下，让用户确认后才能进行接下去的操作。如果用户取消这个操作就让他重新输入一个数据。不过也不仅仅是修改或删除操作，如果其他函数中有这种用户输完数据，就会根据这个数据对相应的文件进行改写的功能，都需要让用户再次确认一下是否输入正确了。

5.测试的时候一定要多循环输入几次，以检查在循环过程中会出现什么问题。

6.Servlet层调用多个service时，要注意逻辑的整合。

下面这个函数写的很好，你们可以参考。

这个servlet是用来删除读者的。需要调用QueryReaderService() 和 DeleteReaderService()。

调用QueryReaderService()是为了获取管理员想删除读者的id号，但是在进入DeleteReaderServlet（）之前可能管理员就已经知晓了想删除读者的id号，因此在调用QueryReaderService()之前先询问管理员是否知晓读者的id。如果知晓，直接去调用DeleteReaderService()，否则先调用QueryReaderService()，再调用DeleteReaderService()。

1. void DeleteReaderServlet()
2. {
3. while (1)
4. {
5. cout << endl << "您要删除一个读者吗？按1确定，其他键退出" << endl;
6. string key;
7. getline(cin, key);
8. if (key == "1")//1代表继续删除操作
9. {
10. cout << endl << "您是否知晓要删除的读者的id？" << endl << "若您知道，请按1继续删除，否则请按其他键查找该用户的id" << endl;
11. getline(cin, key);
12. if (key == "1")//1代表继续删除操作
13. {
14. cout << endl << "请输入你要删除的读者的id" << endl;
15. int tempId;//临时存放要删除的读者的id
16. cin >> tempId;
17. while (!cin)
18. {
19. cout << endl << "输入不合法！您只能输入数字！请重新输入！";
20. cin.clear();
21. //cin.ignore(numeric\_limits<streamsize>::max(), '\n');
22. cin >> tempId;
23. }
24. cin.ignore(numeric\_limits<streamsize>::max(), '\n');//吃掉换行符
25. DeleteReaderService(tempId);
26. cout << endl << "id为" << tempId << "的读者已删除" << endl;
27. }
28. else//其他键代表用户不清楚要删除的读者的id，需要先进行查找操作
29. {
30. QueryReaderService();//查找读者
31. //继续删除，与if中进行的操作相同
32. cout << endl << "请输入你要删除的读者的id" << endl;
33. int tempId;//临时存放要删除的读者的id
34. cin >> tempId;
35. while (!cin)
36. {
37. cout << endl << "输入不合法！您只能输入数字！请重新输入！";
38. cin.clear();
39. //cin.ignore(numeric\_limits<streamsize>::max(), '\n');
40. cin >> tempId;
41. }
42. cin.ignore(numeric\_limits<streamsize>::max(), '\n');//吃掉换行符
43. DeleteReaderService(tempId);
44. cout << endl << "id为" << tempId << "的读者已删除" << endl;
45. }
46. }
47. else//其他按键退出该servlet
48. {
49. cout << endl << "删除读者程序已退出！" << endl;
50. return;
51. }
52. }
53. }