注意：如果查询结果为空的话，要输出提示“查询结果为空”!或者其他提示信息。

如果经过考虑，有的函数需要有返回值或者需要增加参数，就赶紧在群里提出来！尤其是各个函数之间有调用关系的…

登录：

LoginServlet();(可以在servlet里先询问是管理员身份还是读者身份，然后分情况去调用下面的两个service)

AdminLoginService();

ReaderLoginService();

一、管理员：

1.1书籍管理

(由下面的几个service组成)

* + 1. 按条件（id，ISBN，书名，作者，类别，出版社）查询书

QueryBookServlet();

QueryBookService();

1.1.2增加书籍（已存在的书增加amount，不存在的新增一类，如果该书的作者在原作者表中不存在，会自动增加一个作者）

AddBookServlet();

AddBookService()

1.1.3修改书籍信息（id不可以修改）

先查询，然后展示相应的书籍信息，要求输入书籍id，然后给出要修改的项目选择，循环修改，注意容错（修改作者时如果修改之前的作者没有写过其他书籍，就把他删除）

ModifyBookServlet();(通过调用多个Service来完整的完成上面描述的流程)

ModifyBookService();(函数里不用调用查询书籍的service，在ModifyBookServlet会调用。

在这个service开始就直接让用户输入要修改的书籍id即可)

1.1.4删除书籍（假删除，分为删除一类书或减少一类书的amount，若当前书籍有书被借阅，禁止删除）

先查询然后展示相应的书籍信息，要求输入书籍id，然后删除

DeleteBookServlet();(通过调用多个Service来完整的完成上面描述的流程)

DeleteBookService(函数里不用调用查询书籍的service，在DeleteBookServlet里会调用。在这里开始就直接让用户输入要删除的书籍id以及其他信息即可);

* 1. 读者管理

1.2.1按条件（id，姓名，性别，所有）查询读者，已被删除（假删除）的读者不会被检索到

QueryReaderServlet();

QueryReaderService();

1.2.2增加读者（可以是一批，也可以是一个）

AddReaderServlet();

AddReaderService();

1.2.3修改读者信息（id禁止修改）

先查询，然后展示相应的读者信息，要求输入读者id，然后给出要修改的项目选择，循环修改，注意容错.

ModifyReaderServlet();(通过调用多个Service来完整的完成上面描述的流程)

ModifyReaderService();(函数里不用调用查询读者的service，在ModifyReaderServlet里会调用。在这里开始就直接让用户输入要修改的读者id以及其他信息即可)

1.2.4删除读者（假删除，如果当前读者有书未还，禁止删除）

DeleteReaderServlet();(通过调用多个Service来完整的完成上面描述的流程)

DeleteReaderService();(函数里不用调用查询读者的service，在DeleteReaderServlet里会调用。在这里开始就直接让用户输入要删除的读者id)

1.3作者管理

1.3.1按照（id，名字）查询作者

QueryAuthorServlet();

QueryAuthorService();

1.3.2查询作者写的书籍(按照id或姓名)

QueryBooksByAuthor();

QueryBooksByAuthor();

1.4借阅管理

1.4.1根据条件（一类书的id，读者id）查询当前借阅信息

QueryBorrowRecordServlet();(需要调用QueryBookService(),QueryReaderService()，以适当的逻辑组织。也可以查询所有的借阅信息。)

QueryBorrowRecordService() ;(在这之前需要向管理员展示有哪些书，有哪些读者，这些放在QueryBorrowRecordServlet里调用其他Service来展示。这个Service直接让管理员输入一类书的id或者读者id，然后进行相应操作。)

1.4.2根据条件（一类书的id，读者id）查询借阅历史

QueryBorrowHistoryServlet();(需要调用QueryBookService(),QueryReaderService()，以适当的逻辑组织。也可以查询所有的借阅历史。)

QueryBorrowHistoryService();(在这之前需要向管理员展示有哪些书，有哪些读者，这些放在QueryExtendedBookServlet里调用其他Service来展示。这个Service直接让管理员输入一类书的id或者读者id，然后进行相应操作。)

1.4．3根据一类书的id，读者id）查询超期借阅

QueryExtendedBookServlet();(需要调用QueryBookService(),QueryReaderService()，以适当的逻辑组织)

QueryExtendedBookService();(在这之前需要向管理员展示有哪些书，有哪些读者，这些放在QueryExtendedBookServlet里调用其他Service来展示。这个Service直接让管理员输入一类书的id或者读者id，然后进行相应操作。)

1.4.4删除借阅历史和借阅记录（写一个函数，但是流程里不调用，供测试程序用）

DeleteBorrowRecordServlet();(需要调用QueryBookService(),QueryReaderService()，以适当的逻辑组织)

DeleteBorrowRecordService();(在这之前需要展示借阅历史和借阅记录，这些放在DeleteBorrowRecordServlet里调用。这个Service直接让管理员输入借阅历史id或者借阅记录id，然后进行相应操作。

)

二、读者：

1.1

1．1.1按条件（id，书名，作者，类别，出版社）查询书

QueryBookServlet();

QueryBookService();

1.2借书

先查到书，显示书的所有信息，输入id，判断该书的num是否大于1，若大于，可借阅，判断该读者是否有未还的书籍，未还书不能借书，借书的数量不能大于最大借书数量.

BorrowBookServlet(int readerId);(调用QueryBookService()以及下面的service)

BorrowBookService(int readerId);(readerId是当前登录的读者的id号，主函数会将这个id传给这个servlet)

1.3还书

先显示当前借阅，让读者输入借书记录的id进行还书，借书记录真删除一条，借阅历史生成一条，馆藏数量增加

ReturnBookServlet(int readerId);(调用QueryCurReaderBorRecordService (int readerIfd)用来展示当前读者的借书记录，以及用下面的ReturnBookService()完成剩余操作。)

ReturnBookService();

1.4当前借阅

QueryCurReaderBorRecordServlet(int readerId); (readerId是当前登录的读者的id号，主函数会将这个id传给这个servlet)

QueryCurReaderBorRecordService(int readerId);(展示当前读者所有当前借书记录。

显示当前借阅书籍，超期图书，罚款金额，

计算一下可借书的数量

)

1.5借阅历史

1.5.1查询

QueryCurReaderBorHistoryServlet(int readerId); (readerId是当前登录的读者的id号，主函数会将这个id传给这个servlet)

QueryCurReaderBorHistoryService(int readerId);(展示当前读者的所有借阅历史)

1.5.2删除（假删除）

DeleteCurReaderHistoryServlet(int readerId);(调用QueryCurReaderBorHistoryService(int readerId)调用之前，读者也可以选择不查询借阅历史而直接输入借阅记录id号进行删除;)

DeleteCurReaderHistoryService(int readerId);(一定要保证登录的读者只能删除自己的借阅历史。当读者输入一个要删除的借阅历史的id后，要确认这条借阅历史是当前读者的，才可以执行删除操作)

1.6续借

RenewBookServlet(int readerId);(调用QueryCurReaderBorRecordService(int readerId),下面的额service)

RenewBookService(int readerId);(得到续借的记录id,完成续借功能)

三、不登录

UnsignedReaderServlet();(调用QueryBookService()，加上其他输出语句)

调用

四、读者注册

SignUpServlet();

SignUpService();

通用方法：

输出书籍信息（showBooksInfo(List<Book>);）

输出读者信息（showReaderInfo(List<Reader>);）

输出借阅历史（showBorHistoryInfo(List<borrowHistory>)）

输出借阅记录（show BorRecordInfo(list<borrowRecord>)）

根据list中的属性展示对应的信息。