

学号	0121710880414
----	---------------

武汉理工大学

《面向对象与多线程综合实验》报告

学 院	计算机科学与技术学院
专 业	软件工程
班 级	计算机类 m1704
姓 名	穆逸诚
指导教师	许毅

日期_____

1.继承与多态

1.1 实验目的

掌握类的定义、对象的创建、对象的属性的引用和方法的调用；熟悉 Java 中的继承机制，方法的重载与覆盖；掌握多态、抽象类、接口的使用。

1.2 系统功能与描述

第一次实验主要为运用继承与多态的思想编写一个程序，实现档案管理系统中的用户管理模块。通过编写 Administrator 类、Operator 类与 Browser 类将用户分为了系统管理人员、档案录入人员，档案浏览人员，不同类别的人员可以进行不同的操作。本次实验主要实现了以下功能：

1.实现用户的模拟登录过程。通过用户输入，获取用户名和口令；与事先记录在哈希表中的用户信息进行对比，通过口令验证后才能使用系统；

```
public class DataProcessing {  
    static Hashtable<String, User> users;  
    static {  
        users = new Hashtable<String, User>();  
        users.put("jack", new Operator("jack", "123", "operator"));  
        users.put("rose", new Browser("rose", "123", "browser"));  
        users.put("kate", new Administrator("kate", "123", "administrator"));  
    }  
}
```

（用户信息—哈希表）

2.Administrator 类：展示用户列表、添加用户、修改用户、删除用户与显示菜单等功能；

3.Operator 类：上传文件、显示菜单等功能；

4.Browser 类：显示菜单等功能；

其中，在 User 类（用户类）中实现了封装，即通过 setter、getter 方法实现了属性的访问，并以继承的方式实现了上述类。showMenu 方法（显示菜单方法）则运用了多态的思想，实现用户菜单项的展示，即根据用户角色不同，系统自动调用对应 showMenu 方法。

1.3 模块设计

本次实验由于未利用到数据库，因此在 `DataProcessing` 类中使用了哈希表模块以提供测试数据。此次实验的其它模块有：

1.修改用户信息。在 `Administrator` 类中通过 `DataProcessing` 类中的 `search` 方法比对原有信息与输入的信息，匹配成功后再运用 `DataProcessing.update` 方法进行修改，修改前利用字符串的 `equals` 方法检查两次输入的新密码是否一致；

```
public void changeUserInfo() {
    String newName, newPassword, checkPassword, newRole;
    Scanner scan = new Scanner(System.in);
    System.out.print("请输入用户名: ");

    ...

    if(DataProcessing.search(name, password) == null) {
        System.out.println("您输入的信息有误");
    }
    System.out.print("请输入新用户名: ");

    ...

    newRole = scan.nextLine();
    if(newPassword.equals(checkPassword)) {
        if(DataProcessing.update(newName, newPassword, newRole)) {
            System.out.println("更改成功!");
        }
        else {
            System.out.println("更改失败!");
        }
    }
    else {
        System.out.println("两次填入的密码不一致!");
    }
    scan.close();
}
```

2.添加用户信息。在 Administrator 类中通过 DataProcessing 类中的 insert 方法添加新用户；

```
void addUser() {
    Scanner scan = new Scanner(System.in);
    System.out.println("请输入新增用户用户名: ");
    String name = scan.nextLine();
    System.out.println("请输入新增用户密码: ");
    String password = scan.nextLine();
    System.out.println("请输入新增用户身份: ");
    String role = scan.nextLine();
    if(DataProcessing.insert(name, password, role)) {
        System.out.println("新增用户成功! ");
    }
    else {
        System.out.println("新增用户失败! ");
    }
    scan.close();
}
```

3.删除用户信息。在 Administrator 类中通过 DataProcessing 类中的 delete 方法删除用户；

```
void delUser() {
    Scanner scan = new Scanner(System.in);
    String name = scan.nextLine();
    if(DataProcessing.delete(name)) {
        System.out.println("删除成功! ");
    }
    else {
        System.out.println("删除失败! ");
    }
    scan.close();
}
```

4.显示用户信息。在 Administrator 类中，通过 while 循环从 DataProcessing 类中调用显示用户信息；

```
void listUser() {
    Enumeration<User> ele = DataProcessing.users.elements();
    while(ele.hasMoreElements()) {
        User a = (User)ele.nextElement();
        System.out.println("姓名: "+a.getName()+"密码: "+a.getPassword()+...
    }
}
```

5.显示菜单。在 Administrator 类、Operator 类与 Browser 类中通过多态的思想，让系统自动调用对应 showMenu 方法以显示对应的菜单。

```
public void showMenu() {
    System.out.println("*****系统管理人员*****\n");
    System.out.println("1、展示用户列表");
    System.out.println("2、新增用户");
    ...

    System.out.println("请输入功能的序号: ");
    Scanner scan = new Scanner(System.in);
    int n = scan.nextInt();
    switch(n) {
        case 1: listUser();
            break;
        case 2: addUser();
            break;
        ...

        System.out.println("输入序号有误! ");
        break;
    }
    scan.close();
}
```

(Administrator 类中菜单的伪代码)

```

public void showMenu() {
    System.out.println("*****档案浏览人员*****\n");
    System.out.println("1、文件列表");
    System.out.println("2、下载文件");
    System.out.println("3、修改密码");
    System.out.println("请输入功能的序号: ");
    Scanner scan = new Scanner(System.in);
    int n = scan.nextInt();
    switch (n) {
        case 1:
            showFileList();
            break;

        ...

    }
    scan.close();
}

```

(Operator 类中菜单的伪代码)

```

public void showMenu() {
    System.out.println("*****档案浏览人员*****\n");
    System.out.println("1、文件列表");
    System.out.println("2、下载文件");
    System.out.println("3、修改密码");
    System.out.println("请输入功能的序号: ");
    Scanner scan = new Scanner(System.in);
    int n = scan.nextInt();
    switch (n) {
        case 1:
            showFileList();
            break;

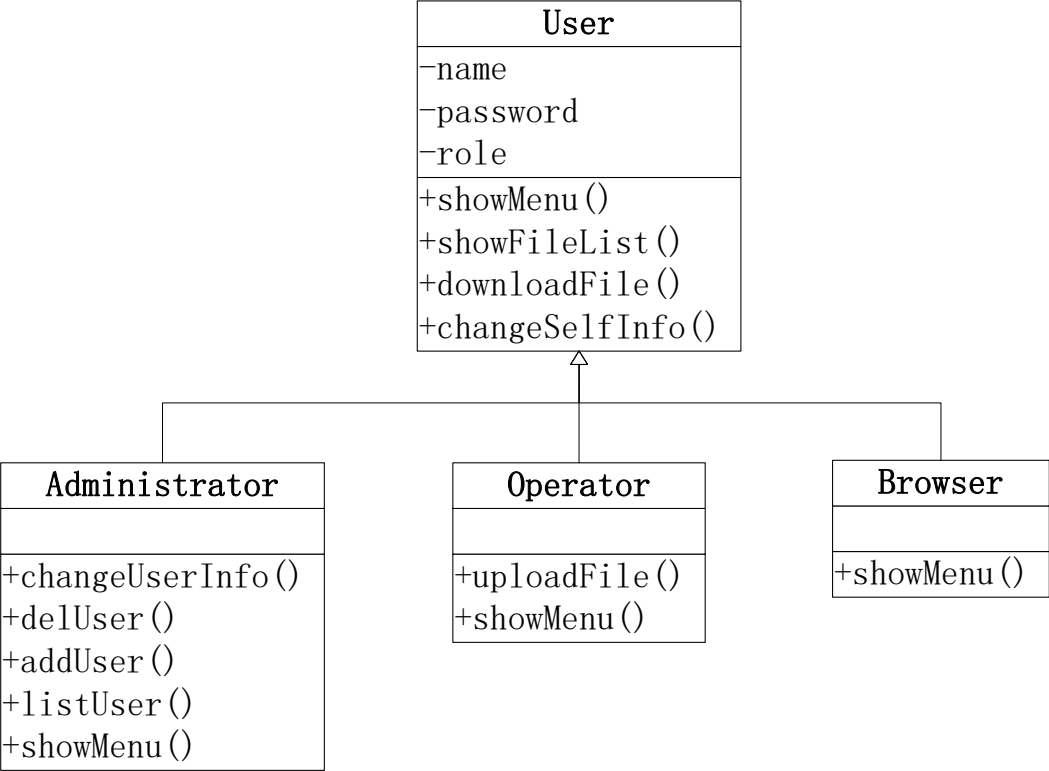
        ...

    }
    scan.close();
}

```

(Browser 类中菜单的伪代码)

本次实验的模块整体结构图如下图所示：



本次实验部分模块的实际运行效果截图：

```
*****欢迎进入档案管理系统*****
1. 登陆
2. 退出
请输入选择的命令： 1
请输入用户名
kate
请输入密码
123
*****系统管理人员*****
1、展示用户列表
2、新增用户
3、修改用户
4、删除用户
5、文件列表
6、下载文件
7、修改密码
请输入功能的序号：
2
请输入新增用户用户名：
qwe
请输入新增用户密码：
123
请输入新增用户身份：
Administrator
新增用户成功！
```

1.4 开发难点与体会

遇到的困难：第一次编写 JAVA 语言综合实验，难免遇到各种小问题，如对 JAVA 的各种方法不熟悉，在编写构造函数时忘记继承父类，next 函数的使用不熟悉等。最大的难点在于对程序的整体架构没有非常清晰的认识，花费了较长时间设计如何设计架构，编写程序。

解决的过程：通过上网查找相关资料和查阅相关书籍，学习了父类构造方法的继承操作，掌握了 next()、nextInt()与 nextLine()的区别，更重要的是设计出了程序的主要结构，有了较为清晰的编程思路。

1.父类构造方法的继承。下图为没有继承父类时的错误构造方法：

```
public class Administrator extends User {  
    Administrator(String name, String password, String role){  
        setName(name);  
        setPassword(password);  
        setRole(role);  
    }  
}
```

更改后的正确代码：

```
public class Administrator extends User {  
    Administrator(String name, String password, String role){  
        super(name, password, role);  
    }  
}
```

体会：此类错误为继承操作方面的失误，学习继承操作为 JAVA 的基础，这个错误让我掌握了利用 super()方法进行继承。

2.next 方法的使用。下图为错误使用 next 方法（使用 nextLine()方法）截图：

```
System.out.println("请输入用户名");  
name = scan.nextLine();  
System.out.println("请输入密码");  
password = scan.nextLine();
```

此时使用 nextLine()将产生错误：


```

*****欢迎进入档案管理系统*****

1. 登陆
2. 退出

请输入选择的命令: 1
请输入用户名
请输入密码
kate
Exception in thread "main" java.lang.NullPointerException
    at Main.main(Main.java:19)

```

通过网上查找，掌握了 next 函数的用法：

Java基础之Scanner类中next()与nextLine()方法的区别

java中使用Scanner类实现数据输入十分简单方便，Scanner类中next()与nextLine()都可以实现字符串String的获取，所以我们会纠结二者之间的区别。

其实next()与nextLine()区别很明确：

next() 方法遇见第一个有效字符（非空格，非换行符）时，开始扫描，当遇见第一个分隔符或结束符(空格或换行符)时，结束扫描，获取扫描到的内容，即获得第一个扫描到的不含空格、换行符的单个字符串。

使用nextLine()时，则可以扫描到一行内容并作为一个字符串而被获取到。

将 nextLine()改为 next()后，程序可正确运行：

```

*****欢迎进入档案管理系统*****

1. 登陆
2. 退出

请输入选择的命令: 1
请输入用户名
kate
请输入密码
123
*****系统管理人员*****

1、展示用户列表
2、新增用户
3、修改用户
4、删除用户
5、文件列表
6、下载文件
7、修改密码
请输入功能的序号:

```

体会：这个错误让我明白了 `next` 方法的相关操作，总结如下：`next()` 方法遇见第一个有效字符（非空格，非换行符）时，开始扫描，当遇见第一个分隔符或结束符(空格或换行符)时，结束扫描，获取扫描到的内容，即获得第一个扫描到的不含空格、换行符的单个字符串。使用 `nextLine()`时，则可以扫描到一行内容并作为一个字符串而被获取到。

1.5 实验总结

本次实验为 JAVA 面向对象与多线程综合实验的第一次迭代开发，主要围绕 JAVA 的封装、继承与多态展开，实验难度于我而言较为适中，难点主要在于对整体程序的架构设计与对封装、继承与多态这种面向对象编程思想的理解及运用。

实验过程中，我遇到了各种各样的问题、困难，通过查阅相关书籍、上网查找相关资料等途径一一解决，学习了利用 `super` 方法进行继承操作，掌握了 `next` 方法的正确使用，明白了 `next()`、`nextInt()`与 `nextLine()`之间的区别，对封装、继承与多态这种面向对象编程思想有了进一步的理解，最重要之处在于对程序的设计、架构有了一个大致思路，这对日后迭代开发有着深远的意义。

总体来讲，本次实验我收获颇丰，学到了关于 JAVA 程序设计的不少知识，解决了编程过程中的许多问题，对面向对象编程思想与程序架构有了更深远的思考。

2.异常处理

2.1 实验目的

理解异常的基本概念；了解 Java 异常的层次结构；熟悉并掌握 Java 异常的捕获处理方法。

2.2 系统功能与描述

第二次实验主要是根据新提供的 DataProcessing 类（因还未讲 SQL，此类模拟异常出现情况，以一定概率随机产生异常），在所编写的 Administrator 类、Operator 类和 Browser 类中增加异常处理功能：

1.在 Administrator 类里，将异常处理代码添加至展示用户列表、添加用户、修改用户、删除用户与显示菜单等功能中，并在父类 User 中通过 Math 类产生随机数以随机产生异常；

2.在 Operator 类里，将异常处理代码添加至 switch 方法的各个 case（选择）中，产生异常的原理同上；

3.在 Browser 类里，将异常处理代码添加至 switch 方法的各个 case（选择）中，产生异常的原理同上；

2.3 模块设计

本次实验主要是在各个类中添加异常处理的方法，因此在各处添加了 try-catch 模块、throw(s)模块。具体载入的模块如下：

1.Administrator 类中，在展示用户列表、添加用户、修改用户、删除用户与显示菜单等功能里添加了 try-catch 模块。下图为匹配账户时的 try-catch 模块：

```
try{
    DataProcessing.search(name, password);
}catch(SQLException a){
    System.out.println("您输入的用户名或密码错误");
    a.printStackTrace();
}
```

2.Operator 类中，在菜单的选项里添加了 try-catch 模块。下图为部分模块：

```
case 1:
    try {
        showFileList();
    }
    catch(SQLException e){e.printStackTrace();break;}
case 2:
    try{
        uploadFile("as");
    }
    catch(IOException a) {a.printStackTrace();
break;}
```

3.Browser 类中，在菜单的选项里添加了 try-catch 模块。下图为部分模块：

```
case 1:
    try {
        showFileList();
    } catch (SQLException a) {a.printStackTrace();break;
}

case 2:
    try {
        downloadFile("df");
    } catch (IOException o) {
        o.printStackTrace();break;
    }
break;
```

4.在 DataProcessing 类中，运用了 throw(s)抛出异常并且使用随机数制造异常：

```
public static User searchUser(String name) throws SQLException{
    if ( !connectToDB )
        throw new SQLException( "Not Connected to Database" );
    double ranValue=Math.random();
    if (ranValue>0.8)
        throw new SQLException( "Error in excecuting Query" );

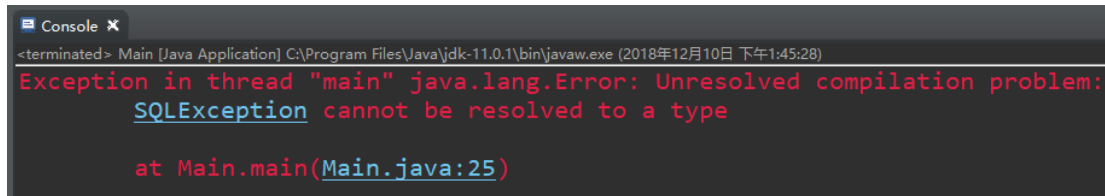
    if (users.containsKey(name)) {
        return users.get(name);
    }
    return null;
}
```

5.随机数在 User 类（父类）中产生：

```
double ranValue=Math.random();
if (ranValue>0.9)
    throw new IOException( "Error
```

2.4 开发难点与体会

遇到的困难：本次实验主要围绕异常处理展开，然而相关知识只在最近的 JAVA 课程与面向对象多线程课程中初次涉猎，对于这方面的知识储备还不够，在编程过程中多个地方出现错误，如 try-catch 格式不正确，未引入异常包，为正确使用 throw 抛出异常等。



```
<terminated> Main [Java Application] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (2018年12月10日 下午1:45:28)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    SQLException cannot be resolved to a type

    at Main.main(Main.java:25)
```

（未导入 java.sql.SQLException 包）



```
24         }
25     catch(SQLException g)
26     {
27         g.printStackTrace();
    }

<terminated> Main [Java Application] C:\Program Files\Java\jdk-11.0.1\bin\javaw.exe (2018年12月10日 下午1:55:47)
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
    Syntax error on token "{", delete this token
    Syntax error on token ")", { expected after this token

    at Main.main(Main.java:25)
```

（try-catch 方法格式不正确）

解决的过程：通过查阅书籍，上网查找相关材料等途径，找到了关于异常处理的相关资料，知道了异常处理的方法，并解决了以上问题：

①try-catch 用法:



try-catch 语句由一个 try 块和其后所跟的一个或多个 catch 子句（为不同的异常指定处理程序）构成。此语句会采用下列形式之一：

```
try try-block
catch (exception-declaration-1) catch-block-1
catch (exception-declaration-2) catch-block-2
...
try try-block catch catch-block
```

其中：

- try-block 包含应引发异常的代码段。
- exception-declaration, exception-declaration-1, exception-declaration-2 异常对象声明。
- catch-block, catch-block-1, catch-block-2 包含异常处理程序。

备注

try-block 包含可能导致异常的保护代码块。该块一直执行到引发异常或成功完成为止。例如，下列转换 null 对象的尝试引发 NullPointerException 异常：

```
object o2 = null;
try
{
    int i2 = (int) o2; // Error
}
```

什么时候来的，告诉我

昵称：流浪的猫
圆龄：13年3个月
粉丝：3
关注：0
[+加关注](#)

< 2018年12月 >

日	一	二	三	四	五	六
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

搜索

常用链接

- [我的随笔](#)
- [我的评论](#)

体会：通过对 try-catch 相关方法的学习，我学会了这种异常处理的方式，了解了使用这种方法的格式：

```
try {
    // 可能会发生异常的程序代码
} catch (Type1 id1) {
    // 捕获并处理 try 抛出的异常类型 Type1
} catch (Type2 id2) {
    // 捕获并处理 try 抛出的异常类型 Type2
} finally {
    // 无论是否发生异常，都将执行的语句块
}
```

此外，我也注意到在使用这种方法时需要导入相关异常的包（例如 java.sql.SQLException 包）。

②throw(s)用法:

原 Java中throws和throw的用法

2017年12月16日 21:18:30 子夕听雨 阅读数: 128

版权声明：版权归作者所有，转载请注明出处 <https://blog.csdn.net/ibllol365/article/details/78822123>

```
1 public class Test_Exception {
2     public static void main(String[] args){
3         int a = 1;
4         int b = 1;
5         add(a,b);
6     }
7
8     public static void add(int a,int b) throws RuntimeException{
9         if (a == b)
10             throw new RuntimeException("Error!!");
11     }
12 }
```

1.throws

throws关键字写在函数名和形参列表后面以及函数体的前面，用于声明该函数将要抛出指定类型的异常，比如上文实例代码中，add函数将会抛出一个RuntimeException异常。

2.throw

throw关键字用于抛出一个特定的异常，如在实例代码的add函数中，如果a等于b，就会抛出一个带有“Error”提示信息的RuntimeException异常。

体会：通过对 throw 相关方法的学习，我又学会了一种新的异常处理的方式，并了解了 throw(s)方法与 try-catch 方法的区别：

- 1、throw 是语句抛出一个异常，throws 是方法抛出一个异常；
- 2、throw 要么和 try-catch-finally 语句配套使用，要么与 throws 配套使用。但 throws 可以单独使用，然后再由处理异常的方法捕获。

以上错误经过修改后，程序可正确抛出异常，实现异常处理功能：

```
PS D:\学习文档\Java\面向对象多线程实验\Java多线程实验\out
****欢迎进入档案系统****
1.登陆
2.退出
*****
1
请输入用户名
请输入口令
kate
123
java.sql.SQLException: Not Connected to Database
    at DataProcessing.search(DataProcessing.java:48)
    at Main.main(Main.java:22)
退出系统
```

(出现异常)

```
PS D:\学习文档\Java\面向对象多线程实验\Java
****欢迎进入档案系统****
1. 登陆
2. 退出
*****
1
请输入用户名
请输入口令
kate
123
***欢迎进入管理人员系统***
1、展示用户列表
2、新增用户
3、修改用户
4、删除用户
5、文件列表
6、下载文件
7、修改密码
请输入选择命令：
1
姓名: jack 密码: 123 身份: operator
姓名: rose 密码: 123 身份: browser
姓名: kate 密码: 123 身份: administrator
退出系统
```

(未出现异常)

2.5 实验总结

本次实验为 JAVA 面向对象与多线程综合实验的第二次迭代开发，主要围绕 JAVA 的异常处理展开，实验难度于我而言较为适中，难点主要在于对程序进行异常处理设计与异常处理方法的使用。

实验过程中，我遇到了各种各样的问题、困难，通过查阅相关书籍、上网查找相关资料等途径一一解决，学习了 try-catch 和 throw(s)等异常处理的方法，掌握了异常处理方法的格式，明白了 try-catch 与 throw(s)之间的区别，对异常处理在 Java 程序中的重要性有了进一步的理解，最重要之处在于对程序的异常处理设计有了更好的认识，这对日后迭代开发有着深远的意义。

总体来讲，本次实验我收获颇丰，学到了关于 JAVA 程序设计的不少知识，解决了编程过程中的许多问题，对面向对象编程思想与程序异常处理有了更多的见解与认知。

3.输入输出流

3.1 实验目的

了解 Java 中 I/O 流的概念和种类；掌握字节流处理和字符流处理，包括 File 类，InputStream/OutputStream 及其子类，Reader/Writer 及其子类；熟练掌握文件的顺序处理，随机访问处理；熟悉对象串行化的概念和方法。

3.2 系统功能与描述

第二次实验主要是根据新提供的 DataProcessing 类（因还未讲 SQL，此类模拟异常出现情况，以一定概率随机产生异常），在所编写的 Administrator 类、Operator 类和 Browser 类中增加异常处理功能：

1.在 Administrator 类里，将异常处理代码添加至展示用户列表、添加用户、修改用户、删除用户与显示菜单等功能中，并在父类 User 中通过 Math 类产生随机数以随机产生异常；

2.在 Operator 类里，将异常处理代码添加至 switch 方法的各个 case（选择）中，产生异常的原理同上；

3.在 Browser 类里，将异常处理代码添加至 switch 方法的各个 case（选择）中，产生异常的原理同上；

3.3 模块设计

本次的实验中增加了一个 Doc 类，这个类在教学资料中进行了给出，在实际编码的过程中我直接使用了给出的代码。本次设计到的几个操作如下：

文件上传：

- 1、本地模拟上传：在本地模拟文件的上传，将文件从系统的任意位置写入到 指定的位置，即在指定的位置创建一个文件将要上传的文件保存到该目录下；
- 2、文件信息的写入：将上传的文件的信息保存到档案系统中，此处即将文件 插入到哈希表中。

文件下载：

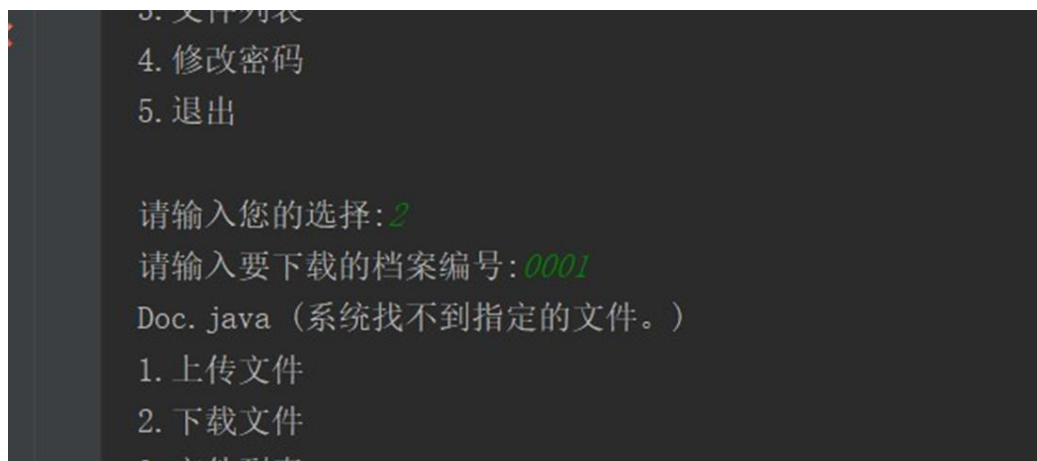
- 1、在系统（哈希表）中进行查找，如果再到输入的信息说明这个文件能够下载，如果没有返回 False；
- 2、文件下载：如果在哈希表中找到文件的相关信息就进行下载，下载过程即按照流的方式从系统中读入文件，然后将文件写到指定的位置。

文件列表：

类似之前用户列表的操作，对系统中的文件信息利用哈希表的相关的函数进行文件信息的枚举，然后按照指定的格式进行文件信息的输出。

3.4 开发难点与体会

错误信息一：



```
3. 文件列表
4. 修改密码
5. 退出

请输入您的选择:2
请输入要下载的档案编号:0001
Doc. java (系统找不到指定的文件。)
1. 上传文件
2. 下载文件
3. 文件列表
```

分析：

首先通过按照流的方式进行文件的读写，在这部分的操作中遇到的第一个问题就是通过输入文件的编号，只能确定在系统中（本次实验为哈希表）中好到了这个文件，但是不能正确的进行文件的下载。

处理办法：

要处理这个问题首先要定位到产生问题的地方。程序运行的过程中无法进行文件的下载，涉及到的类应该是父类中的下载函数的问题，或者是在操

作员的位置产生了异常。

在处理的过程中我将 `Operator` 类中的一个异常处理部分进行了反馈信息的重写：

```
    } catch (IOException e) {  
        System.out.println("捕获到了一个异常");  
    }  
}
```

然后通过运行确定了抛出异常的位置：

```
请输入您的选择:2  
请输入要下载的档案编号:0001  
捕获到了一个异常
```

通过本次的运行确定到了抛出异常的位置为 `User` 类中的文件下载出现了问题，因此在这个类的构建文件的位置设置了一个断点进行调试。

```
@  
public FileInputStream( @NotNull File file) throws FileNotFoundException { file: "Doc. java"  
    String name = (file != null ? file.getPath() : null); name: "Doc. java"  
    SecurityManager security = System.getSecurityManager(); security: null  
    if (security != null) {  
        security.checkRead(name); security: null  
    }  
    if (name == null) { name: "Doc. java"  
        throw new NullPointerException();  
    }  
    if (file.isInvalid()) { file: "Doc. java"  
        throw new FileNotFoundException("Invalid file path");  
    }  
}
```

最终通过调试却定这是 `FileNotFoundException` 抛出的异常。然后我对中断的地方进行了检查，发现是文件的路径设置的问题。通过对保存文件的目录的绝对路径进行设置，最后解决了问题。

体会：

这次在编写的过程中我遇到了一个小的问题就是文件路径的问题，虽然该问题并非特别严重，然而在调试过程中却让我花费了大量的时间来发现它

通过这次的调试我获得了以下的经验：

- 1、如何分析产生异常的位置，然后设置断点逐步的进行调试
- 2、另外的一个经验就是在编写相关的异常处理的时候要尽可能的将 try 语句中的部分变得更加的简单，这样有利于程序中出现异常的时候快速的查找到是程序的哪一位置产生的异常。然后进行调试解决，提高效率。

错误信息二：

```
5. 退出

请输入您的选择:2
请输入要下载的档案编号:0001
D:\Java_Project\Lesson3\File\downloadfile (拒绝访问。)
```

原因：

在这里我自己通过调试并没有解决这个问题，这里出现的问题我分析仍然是文件路径的问题，但是检查仍然没有找到问题，于是通过上网查找了相关资料：

```
at org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.7
```

出现此问题原因在于：

在创建输出流的时候，只写了文件的名称，如果要实现上传则必须是文件的路径；才能正常上传；如果有遇到类况请看下自己的路径是否正确； 代码中丢了path，因此报了“拒绝访问”的错误。

```
String path = request.getServletContext().getRealPath("/attr/");
File folder = new File(path);
if(!folder.exists()){
    folder.mkdirs();
}
String fileName = myfile.getOriginalFilename();
if(!fileName.isEmpty()){
    String ext = FilenameUtils.getExtension(fileName);
    String newFileName = new Date().getTime()+"_"+new Random().nextInt(1000000000)+ext;
    myfile.transferTo(new File(path+File.separator+newFileName));
}
```

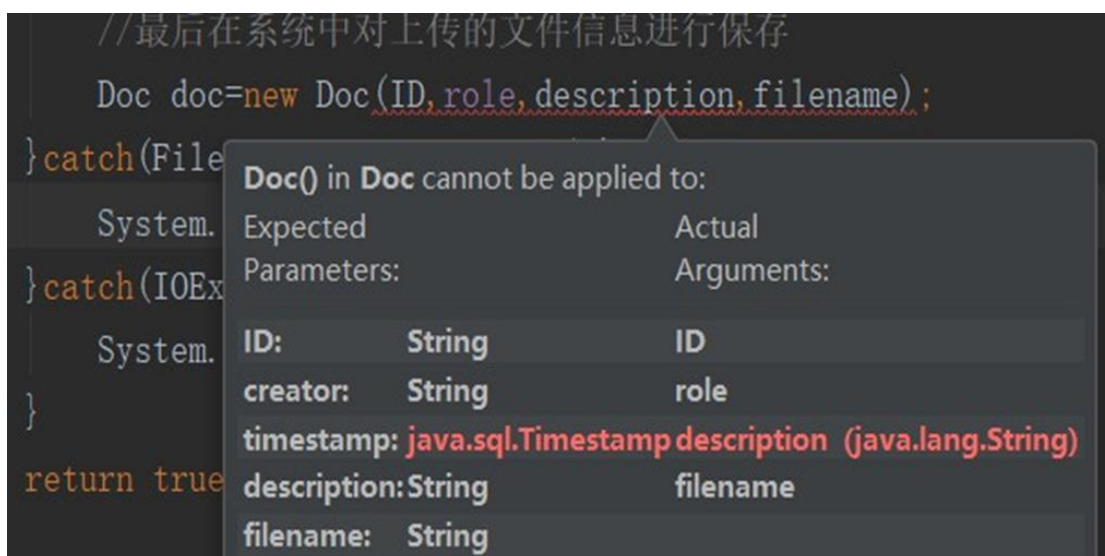
https://blog.csdn.net/Sky_r1

网上的一篇博客分析依然是路径的问题，于是我直接将有关路径的部分代码删除进行了重写，解决问题。

体会：

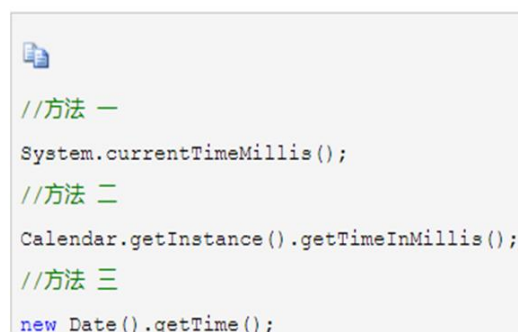
虽然对于这个问题直到解决我也没弄清楚原因，只能大致判断是文件的路径的问题，但是经过重写还是解决了问题。这也算一种经验吧，遇到了问题，反复检查也不能解决，如果代码较简单就直接进行重写。

错误信息三：



在创建一个新的文件对象时，他的构造函数中有一个参数为时间戳，这是我在之前并没有碰到过的，于是查阅了网上Java 时间戳的相关信息。

获取当前时间戳



通过上面地址的博客的阅读，我通过创建了一个 Date 对象完成了对时间信息的写入：

```
//最后在系统中对上传的文件信息进行保存  
Date date=new Date();  
Timestamp timestamp=new Timestamp(date.getTime());  
DataProcessing.insertDoc(ID,name,timestamp,description,uploadfile.getName());
```

3.5 实验总结

本次的实验内容主要围绕输入输出流展开，关于这部分的内容在 Java 程序课上并没有进行太多的讲述，只是在学习 C++面向对象时对流的概念有一定的了解。通过这次实验，我了解 Java 中 I/O 流的概念和种类；掌握字节流处理和字符流处理，包括 File 类，InputStream/OutputStream 及其子类，Reader/Writer 及其子类；熟练掌握文件的顺序处理，随机访问处理；熟悉对象串行化的概念和方法。

虽然本次的实验并不是太难，但是还是遇到了不小的困难。首先是我在编写程序的时候犯了几个低级的错误就是关于文件路径的问题，实验中我遇到了无法找到文件的异常和文件拒绝访问的异常。并且有一个问题我经过调试并没有解决直接通过相关部分的代码的重写得到的实现。同时在本次的实验中文件的读写中我使用的是课件上提供的一个办法来获取到文件中的全部的内容并将其进行模拟下载相关的操作。

最后通过本次的实验获得几个方面的收获就是程序的调试和程序相关异常的查找定位。另外，通过本次的实验我知道了 java 中的时间戳的用法，以及如何获取到时间并且进行时间的转换，让时间按照指定的格式进行输出。总体而言，本次实验我收获颇丰。

4.GUI 设计

4.1 实验目的

了解 Java 图形界面程序的基本结构;掌握 Java 布局管理和常用组件的使用;掌握 Java 事件处理机制。

4.2 系统功能与描述

第四次实验主要是给档案管理系统通过 Java GUI，添加图形用户界面。本次实验主要设计了系统界面的布局，并添加了了菜单、按钮、文本框、密码框、下拉列表、文件对话框等组件，实现了系统界面化。

4.3 模块设计

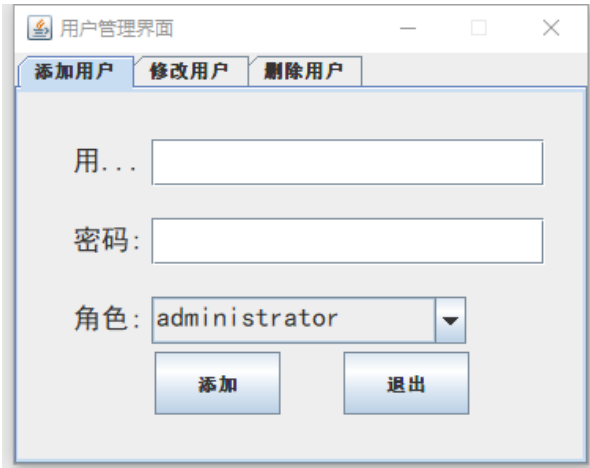
①LoginWindow 类：实现了第一个登录窗口；



②MenuWindow 类：构建了界面的主框架；



③PasswordWindow 类：实现了密码修改界面；



用户管理界面

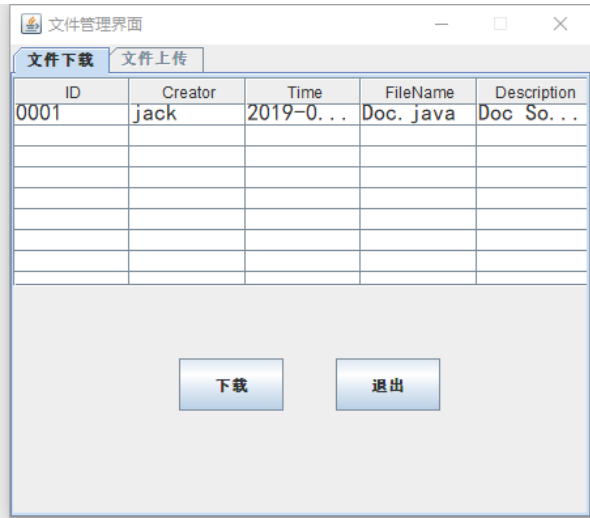
添加用户 修改用户 删除用户

用...

密码:

角色: administrator

④UpDownloadWindow 类：实现了文件上传下载界面；



文件管理界面

文件下载 文件上传

ID	Creator	Time	FileName	Description
0001	jack	2019-0...	Doc. java	Doc So...

⑤UserWindow 类：实现了用户管理界面；



密码修改界面

账号:

旧密码:

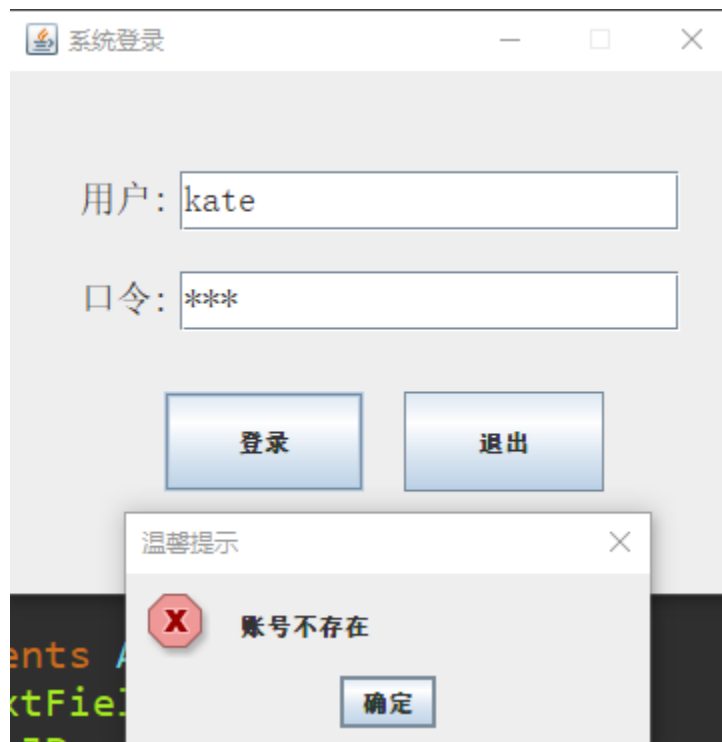
新密码:

重复密码:

角色:

4.4 开发难点与体会

遇到的问题：输入正确信息后无法进入主界面，提示账号不存在



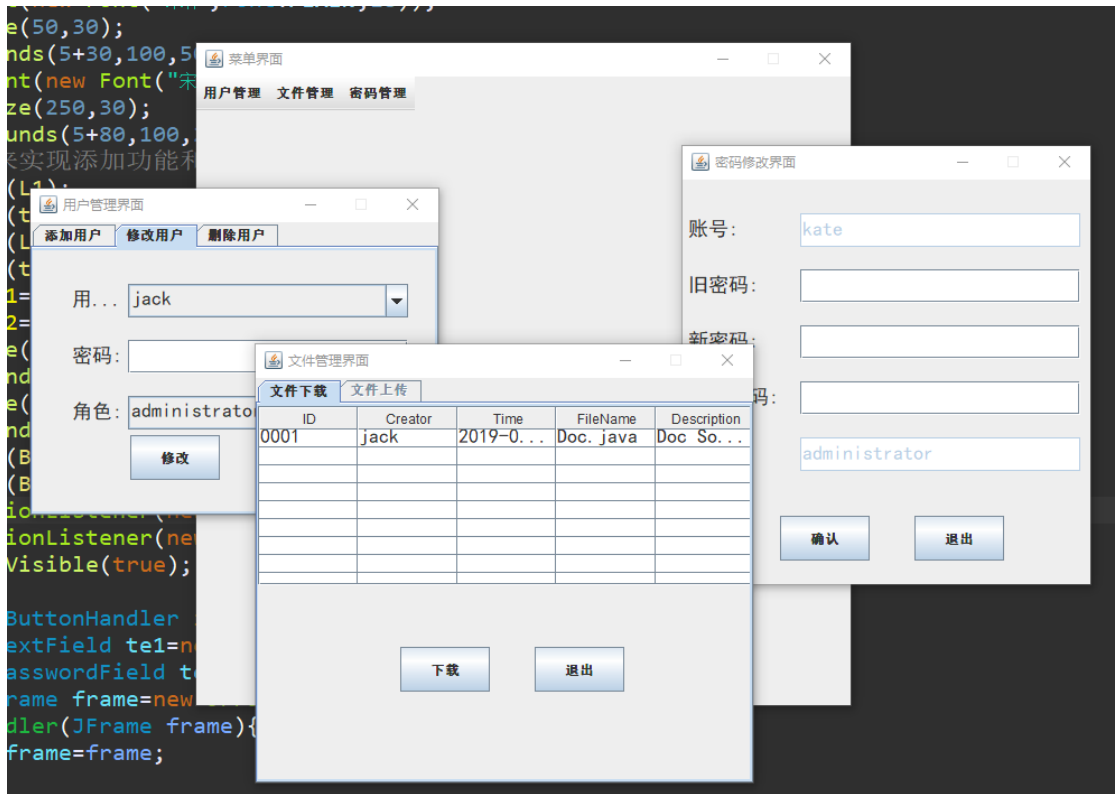
解决的方法：这个错误产生时，程序并未报错。经过长时间的仔细检查，发现登陆按钮的监听有问题

```
JButton B1=new JButton("登录");
JButton B2=new JButton("退出");
B1.setSize(100,50);
B1.setBounds((380-100-5)/2-60,(300-50-30)/2+50,100,50);
B2.setSize(100,50);
B2.setBounds((380-100-5)/2+60,(300-50-30)/2+50,100,50);
frame.add(B1);
frame.add(B2);
B1.addActionListener(new ButtonHandler(frame));
B2.addActionListener(new ButtonHandler(frame));
```

应将相应部分改为

```
JButton B1=new JButton("登录");
JButton B2=new JButton("退出");
B1.setSize(100,50);
B1.setBounds((380-100-5)/2-60,(300-50-30)/2+50,100,50);
B2.setSize(100,50);
B2.setBounds((380-100-5)/2+60,(300-50-30)/2+50,100,50);
frame.add(B1);
frame.add(B2);
B1.addActionListener(new ButtonHandler(frame,te1,te2));
B2.addActionListener(new ButtonHandler(frame));
```

更改后，程序可正常运行



4.5 实验总结

本次实验主要围绕 GUI 设计展开，难度不大，但需要添加的内容较多，涉及到很多 GUI 方面的操作，相对复杂。本次实验中，我主要将前面课程所编写的档案管理系统改编为图形用户界面。通过构思，对比，我选用了合适的布局，综合使用了菜单、按钮、文本框、密码框、下拉列表、文件对话框等组件，实现了良好的人机接口。

在实验过程中，我遇到了许多问题，如，好在通过查阅相关书籍与上网查找相关资料等手段，将上述问题逐一解决。

通过这次实验，我学习了有关 Java GUI 设计的很多知识，进一步了解到了 Java 编程的强大，让程序有了界面，在编程过程中也自己解决了很多问题。总而言之，本次实验我收获不少，对面向对象编程思想与程序 GUI 设计有了更多的见解与认知。

5.JDBC 数据库操作

5.1 实验目的

理解 JDBC 的特点，结构，应用模型；了解 JDBC 驱动程序的类型；掌握通过 JDBC 访问数据库执行 SQL 语句的方法。

5.2 系统功能与描述

本实验将档案管理系统中的数据管理方式改写为数据库存储，实现原来程序中关于用户、文档信息的数据库操作。

利用 LoginWindow()类来实现初始用户登陆界面的功能实现,用 MenuWindow()类来实现根据用户的输入密码账号的不同来弹出第二个界面，显示不同的功能。PasswordWindow()用以实现用户密码的修改，UpDownloadWindow()类用以实现上传下载的功能，UserWindow()类用以实现用户界面的相关操作，而所有的数据库操作封装在 DataProcessing 类中，并保留原来接口，这样可保持原有程序不做大的变化。

5.3 模块设计

1、在类 LoginWindow()中添加数据库驱动类 URL 等信息，创建驱动用于后继连接数据库管理软件

```
import ...  
/serial/  
public class LoginWindow extends JFrame{  
    private static String driverName="com.mysql.jdbc.Driver";// 加载数据库驱动类  
    private static String url="jdbc:mysql://localhost:3306/document";// 声明数据库的URL  
    private static String user="root";// 数据库用户  
    private static String psw="12345678";  
    ...  
}
```

(默认 connectedToDatabase 为 false)

2、DataProcessing 类中添加连接数据库的函数

当连接成功后 赋值为 true

```
        connection = DriverManager.getConnection(url,user,password);  
        connectedToDatabase = true;  
    }  
}
```

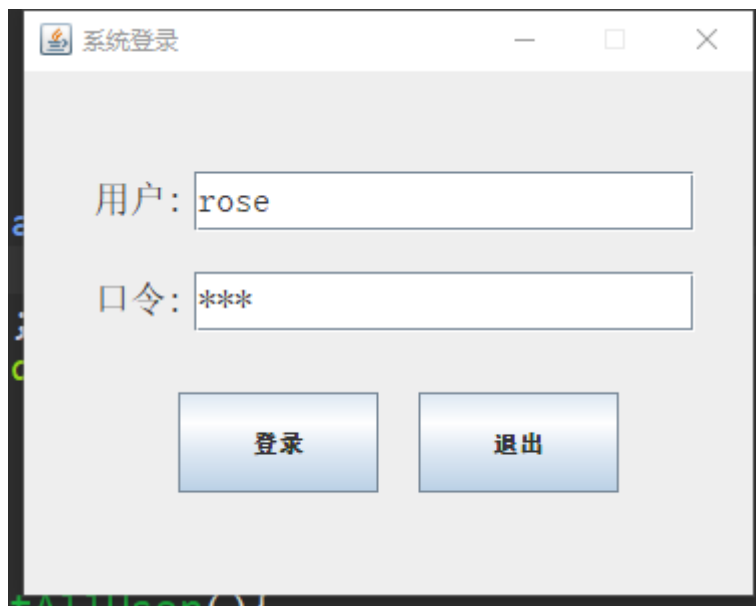
插入文件函数，对文件的 DOC 信息要返回数据库

```
public static boolean insertUser(String name, String password, String role) throws SQLException {
    User temp;
    if (!connectedToDatabase)
        throw new SQLException("Not Connected to Database");
    if (role.equals("administrator"))
        temp = new Administrator(name, password, role);
    else if (role.equals("browser"))
        temp = new Browser(name, password, role);
    else if (role.equals("operator"))
        temp = new Operator(name, password, role);
    statement = connection.createStatement();
    String sql = "insert into user_info (username,password,role) values ('"+name+"','"+password+"','"+role+"')";
    if (statement.executeUpdate(sql) > 0) {
        return true;
    }
    return false;
}
```

创建新用户，同样的需要返回到数据库，还要考虑到可能出现的异常，例如没有连接上数据库。

5.4 开发难点与体会

问题一：在调试过程中，曾多次出现卡死在登陆界面

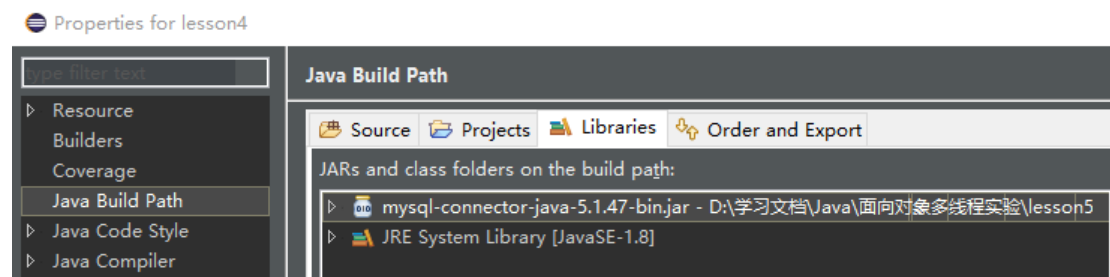


分析问题在与数据库连接失败 `connectedToDatabase` 变量为 `false`, 抛出异常，导致后继函数无法运行。

解决方案：除在 `LoginWindow` 缺少对应的加载数据库驱动等必要信息外，

```
private static String driverName="com.mysql.jdbc.Driver";// 加载数据库驱动类
private static String url="jdbc:mysql://localhost:3306/document";// 声明数据库的URL
private static String user="root";// 数据库用户
private static String psw="12345678";
```

在 Eclipse 的 Project Properties 中，设置 Java Build Path，导入相应驱动：mysql-connector-java-5.1.47-bin.jar

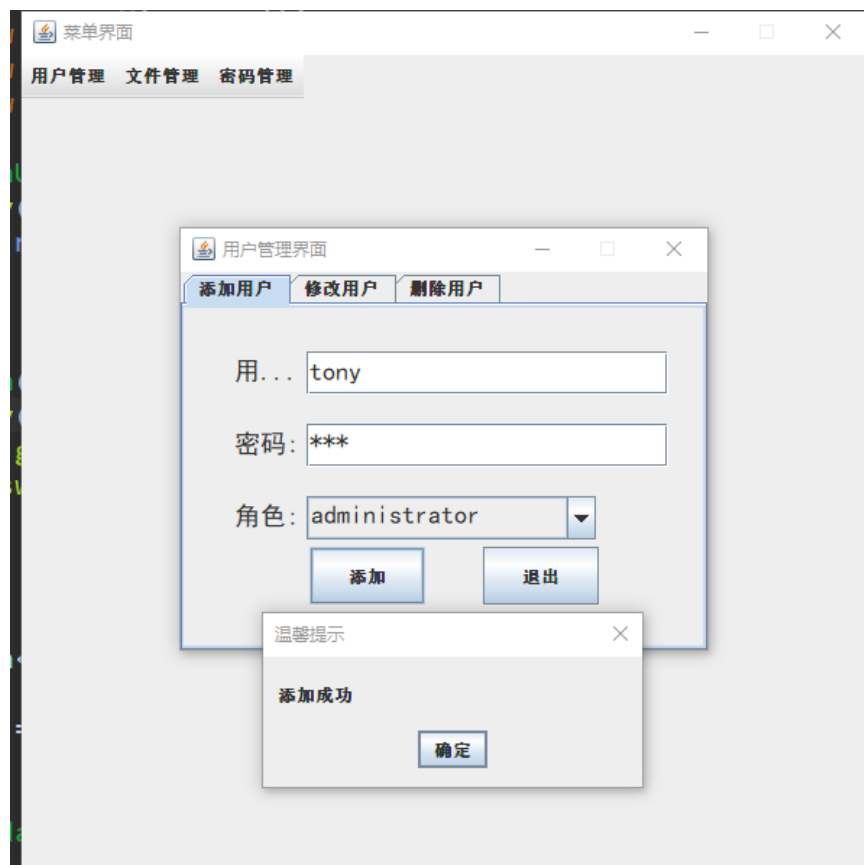


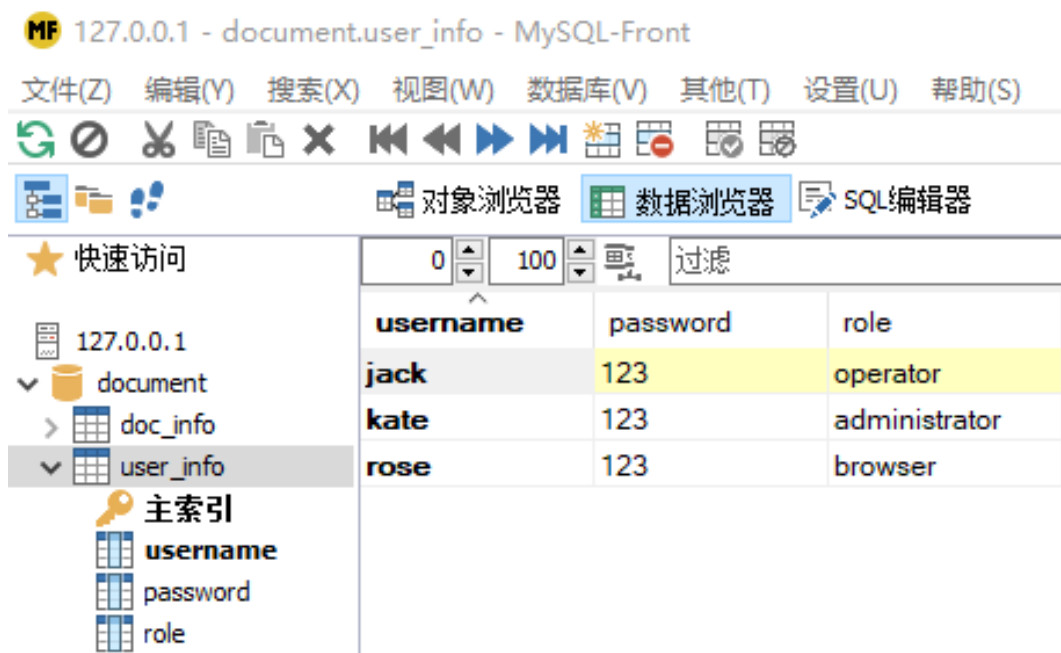
另外在开发 GUI 的经验里 从用户的角度思考

```
try {  
    DataProcessing.connectedToDatabase(url,user,psw,driverName);  
} catch (Exception e1) {  
    JOptionPane.showMessageDialog( parentComponent: null, message: "数据库连接失  
    e1.printStackTrace();  
}
```

需要在 LoginWindow 类的 try catch 语句中添加对应的异常信息输出框 从用户的角度出发考虑，在登陆界面无法运行，应当有对应的图形界面输出数据库连接失败。

问题二：用户管理界面中添加的新用户无法添加到数据库中





解决方案:

```
resultSet.CONCUR_READ_ONLY);
String sql = "update user_info set role = '"+role+"' where username='"+name+"'";
String sql1 = "update user_info set password = '"+password+"' where username='"+name+"'";
int count = statement.executeUpdate(sql);
int count1 = statement.executeUpdate(sql1);
```

在 DataProcessing 的 update 和 insertUser 需要添加编写正确的 String 语句才可返回到数据库中 结果集以 ResultSet 接口的一个对象的形式返回

```
statement = connection.createStatement();
String sql = "insert into user_info (username,password,role) values ('"+name+"','"+password+"';
if((statement.executeUpdate(sql))>0){
    return true;
```

5.5 实验总结

本次实验主要围绕 JDBC 数据库操作展开，实验难度相对较大，涉及到的新知识较多。在这次实验中，我主要遇到了所安装的 MySQL 8.0.13 无法正常启动、多次卡死在登陆界面、用户管理界面中添加的新用户无法添加到数据库管理软件 Navicat for MySQL 中等问题，好在通过查阅有关书籍与上网查找相关资料等途径，将问题一一解决。

通过这次实验，我对 JDBC 数据库有了一定的了解，学习到了数据库的相关操作，建立了程序对应的数据库并通过 JDBC 成功访问了数据库，执行了 SQL 语句。数据库作为当今热门的计算机方面内容，学好数据库是必要的，相信今后也会多处用到数据库。总体而言，本次实验收获颇丰。

6.网络编程

6.1 实验目的

了解 Java 网络编程基础知识；掌握 java.net 包中关于网络的基本类及其属性和方法；掌握基于 Socket 的客户和服务端编程方法。

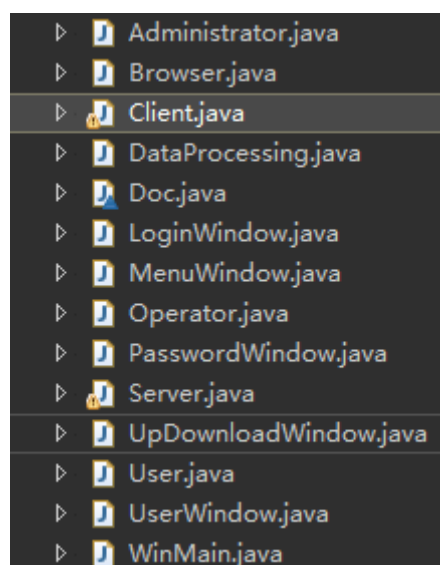
6.2 系统功能与描述

将原有系统从单机版升级为客户机——服务器模式（模拟），通过 Socket 编程，实现了档案通过客户机操作，从服务端进行上传，下载等功能。

本次实验新增了 Client 类与 Server 类，用于启动客户端与服务器，并实时呈现服务器当前状态，实现了系统的网络化。

6.3 模块设计

第六次实验在系统原有的功能上新增了 Client 类与 Server 类



①Client 类：用于进行客户端的相关操作

```
private static JFrame Jframe;  
private static DataOutputStream output;  
private static DataInputStream input;  
private String message = "";  
private static String user_name="";  
private static String user_password="";  
private String chatServer;  
private static Socket client;
```

②Server 类：用于服务器的处理与显示操作状态

```
public class Server{
    private static DataOutputStream output;
    private static DataInputStream input;
    private ServerSocket server;
    private Socket connection;
    public Server(){
        super();
    }
    public void runServer() throws SQLException{
```

6.4 开发难点与体会

遇到的问题一：报错显示无法连接到数据库

```
CommunicationsException: Communications link failure

...
y to the server was 0 milliseconds ago. The driver has
l.reflect.NativeConstructorAccessorImpl.newInstance0(Na
l.reflect.NativeConstructorAccessorImpl.newInstance(Na
l.reflect.DelegatingConstructorAccessorImpl.newInstance
reflect.Constructor.newInstance(Constructor.java:490)
handleNewInstance(Util.java:425)
or.createCommunicationsException(SQLError.java:990)
IO.<init>(MysqlIO.java:342)
tionImpl.coreConnect(ConnectionImpl.java:2197)
```

解决方法：经故障排查，发现是由于未启动数据库服务而导致的

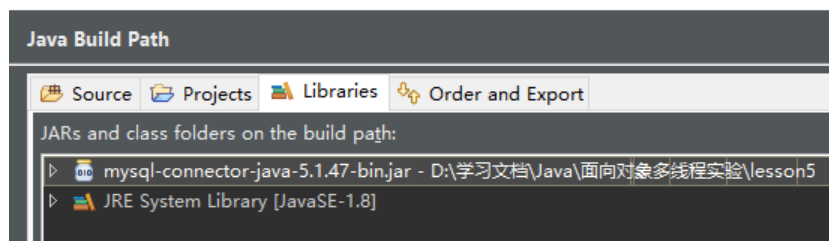
服务				
MySQL-MU				
名称	描述	状态	启动类型	
Microsoft Store 安装服务	为 M...		手动	
Microsoft Windows SMS ...	根据...		手动(触发...	
MySQL-MU			手动	

启动数据库后，程序正常运行

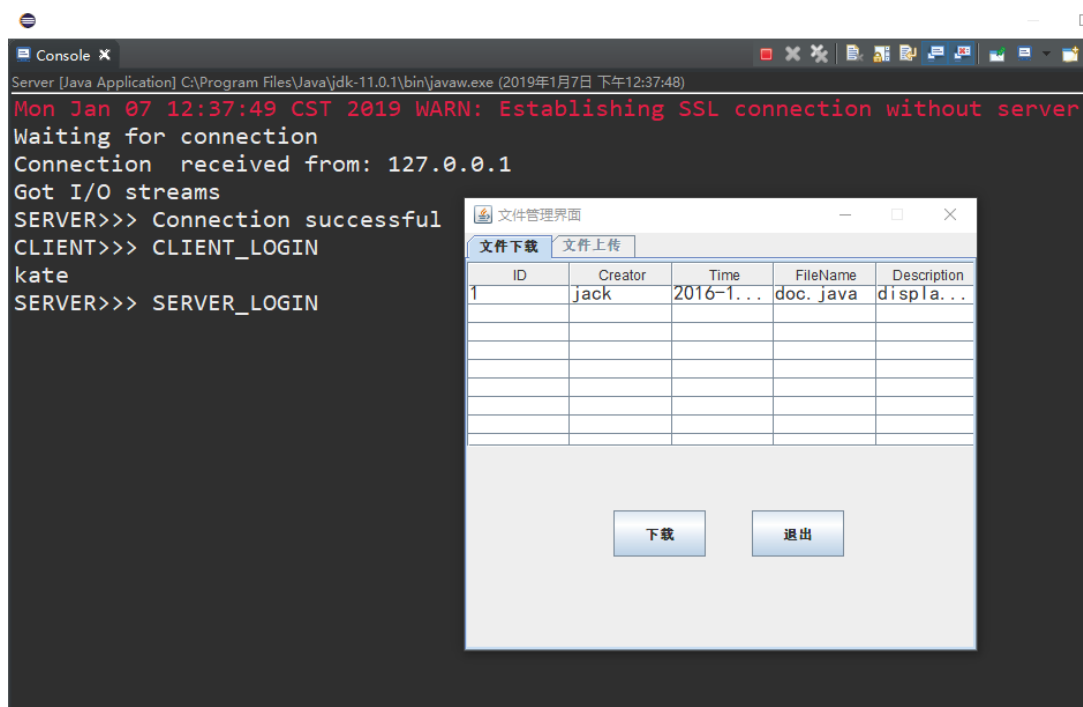
遇到的问题二：报错显示连接数据库错误

```
ConnectException: Connection refused: connect
/java.net.PlainSocketImpl.connect0(Native Method)
/java.net.PlainSocketImpl.socketConnect(PlainSock
/java.net.AbstractPlainSocketImpl.doConnect(Abstr
/java.net.AbstractPlainSocketImpl.connectToAdres
/java.net.AbstractPlainSocketImpl.connect(Abstrac
/java.net.SocksSocketImpl.connect(SocksSocketImpl
/java.net.Socket.connect(Socket.java:591)
```


解决方法：经检查，发现需在 Eclipse 的 Project Properties 中，设置 Java Build Path，导入相应驱动：mysql-connector-java-5.1.47-bin.jar



导入后，程序正常运行



6.5 实验总结

本次实验主要围绕网络编程展开，实验难度相对较大，涉及知识面广。由于本次实验所涉及到的 Socket 网络编程内容少在 Java 课程上提及，需要自己查阅相关书籍并上网查找大量相关资料，才能对该部分内容有一定了解。之后，需要对程序进行多次反复调试，方可完成本次实验。

在本次实验中，我学习了 Java 网络编程基础知识，认识了 java.net 包中关于网络的基本类及其属性和方法，并对基于 Socket 的客户和服务端编程方法有了一定的了解。实验过程中也遇到了许多问题，通过细心检查也都一一解决。总体来讲，本次实验收获很大。

7.多线程

7.1 实验目的

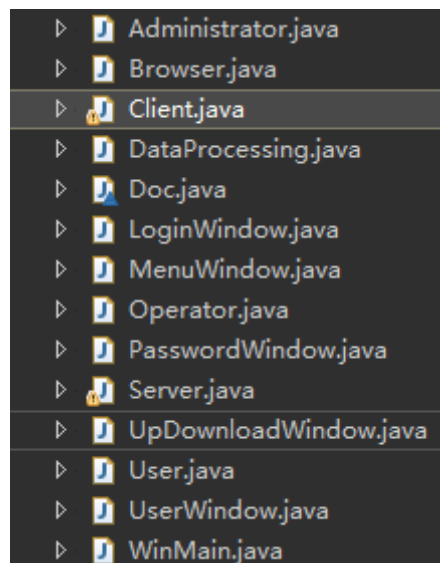
了解线程的基本概念；掌握线程的构造，调度，控制策略；掌握多线程互斥和同步处理方法。

7.2 系统功能与描述

本次实验主要目的在于实现程序的多线程运行功能，因此我将档案管理系统改写成多线程运行模式，使得程序可以同时运行多个客户端，并允许多个客户端同时进行各种操作。

7.3 模块设计

本次实验在原有类的基础上，对部分类添加了多线程操作



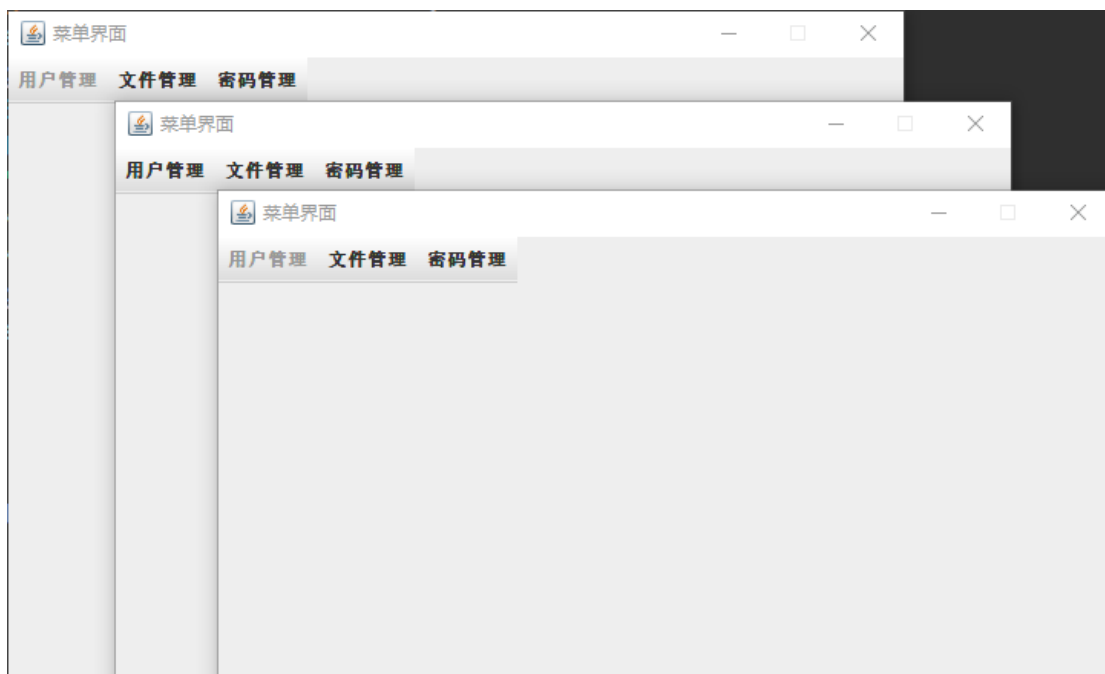
```
SwingUtilities.invokeLater(  
    new Runnable()  
    {  
        public void run()  
        {  
            System.out.println(messageToDisplay);  
        }  
    }  
);  
}
```

7.4 开发难点与体会

开发难点：添加多线程后，程序执行错误，无法同时打开多个窗口

```
Constructor.newInstance(Constructor.java:490)
newInstance(Util.java:425)
ol.getInstance(ConnectionImpl.java:386)
ngDriver.connect(NonRegisteringDriver.java:330)
ager.getConnection(DriverManager.java:677)
ager.getConnection(DriverManager.java:228)
ectToDatabase(DataProcessing.java:22)
```

解决方法：调试程序的多线程部分，发现语法错误，修改后，程序可正常运行



```
Waiting for connection
Connection received from: 127.0.0.1
Got I/O streams
Thread1 come in...
Connection successful!
Thread1:
CLIENT>>>CLIENT_LOGIN
kate
SERVER>>>SERVER_LOGIN
```

7.5 实验总结

本次实验是本学期 Java 综合实验的最后一次，实验主要围绕多线程展开，实验难度适中，主要编程过程为在系统代码的相应地方添加多线程的功能，实现程序的多开。

在本次实验中，由于添加多线程的位置较多，难免出现语法错误等失误，同时在调试过程中也遇到了许多其他问题，好在通过细心检查，问题全部统一解决，并在解决问题的过程中，学习到了许多平时未注意的知识，因而在在我看来，程序的调试过程也是对自己 Java 知识的查漏补缺。

通过本次实验，我进一步掌握了 Java 多线程编程的思想与操作，了解线程的基本概念，学习了线程的构造，调度，控制策略，并明白了多线程互斥和同步处理方法。总体而言，本次实验收获颇丰。

完成了 Java 综合实验七次课程后，我对 Java 的面向对象编程思想有了全新的认识与理解，体会到了 Java 编程与 C++编程的不同之处，并将 Java 课程所学到的知识付诸实践，最后完成了本次综合实验编程，不仅有了程序编写完整的满足感，更学习到了很多 Java 课程中学习不到的知识，如：JDBC 数据库操作，网络编程等。

通过这七次实验课，我最大的收获便是我加深了对面向对象编程思想的理解。Java 与 C++一样，是面向对象的编程语言，与面向过程的 C 语言不同。在这之前，我喜欢将所有操作全部集成于主函数中，这样做虽然看似效率高，实则是在调试过程中时常因为错误关联位置太多，导致修改程序变得非常复杂。有了面向对象的编程思想，我在编程时注重程序的继承性、多态性与封装性，使得程序界面简洁，条理清晰，也让调试过程变得简单。

总体来讲，通过这七次实验课，我学习到了很多新知识，体会到了 Java 编程的乐趣所在。希望以后有更多的机会编写程序，以来多多磨练我的编程能力！

《面向对象与多线程综合实验》成绩评定表

班级：

姓名：

学号：

序号	评分项目	满分	实得分
1	学习态度认真、遵守纪律	10	
2	迭代开发进度合理，提交结果正确	30	
3	软件功能完善、界面美观、运行正确	20	
4	代码规范、注释清晰、可读性好	10	
5	系统演示汇报情况良好、回答问题正确	10	
6	报告规范、文理通顺、描述清晰	20	
		总得分	

评语：

指导教师签名：

2018 年 月 日