

学号	0121710880415
----	---------------

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	颜道江
指导教师	许毅

日期 2018 年 11 月 12 日

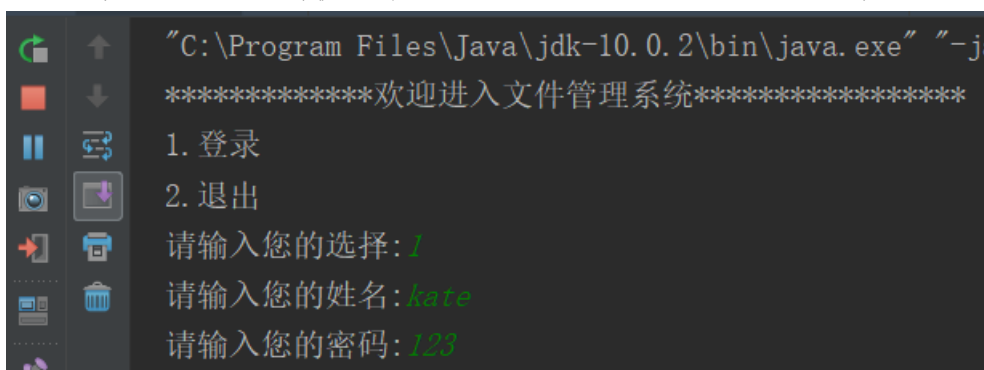
1. 继承与多态

1.1 实验目的

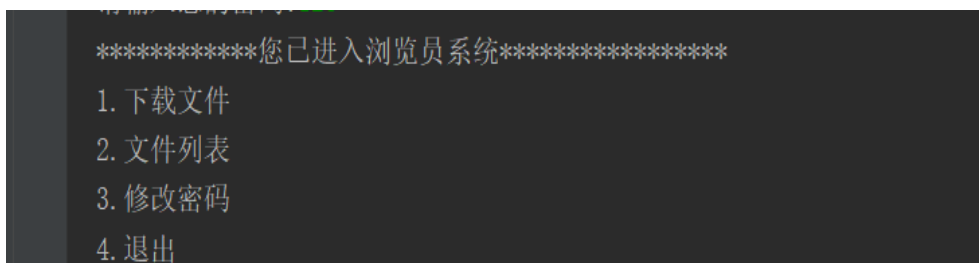
掌握类的定义,对象的创建,对象的属性的引用和方法的调用:熟悉 Java 中的继承机制,方法的重载与覆盖;掌握多态,抽象类,借口的使用。

1.2 系统功能描述

1. 首先是用户输入自己的用户名和口令进行登录;在登录的过程中根据用户的输入到哈希表中去查找是否存在这个用户,如果存在就根据用户的身份创建相关的对象,显示对应身份的菜单,否则给用户返回相应的提示信息;



2. 如果用户的经过登录验证,同时用户的身份为浏览员,就显示浏览员的菜单:由于此时还没有涉及到文件相关的操作,因此在浏览员的菜单中相关的操作只是打印相关的提示信息;



3. 如果经过登录验证,同时身份确认为档案录入人员,就显示档案录入人员的菜单选项。由于档案录入人员的操作在浏览人员的基础上就增加了档案的上传的功能

*****您已进入档案录入人员系统*****

1. 上传文件
2. 下载文件
3. 文件列表
4. 修改密码
5. 退出

4.如果经过登录验证同时将身份确认为档案管理人员，那么菜单中就有下面的选项

*****您已进入管理员系统*****

1. 修改用户
2. 删除用户
3. 新增用户
4. 用户列表
5. 下载文件
6. 文件列表
7. 修改密码
8. 退出

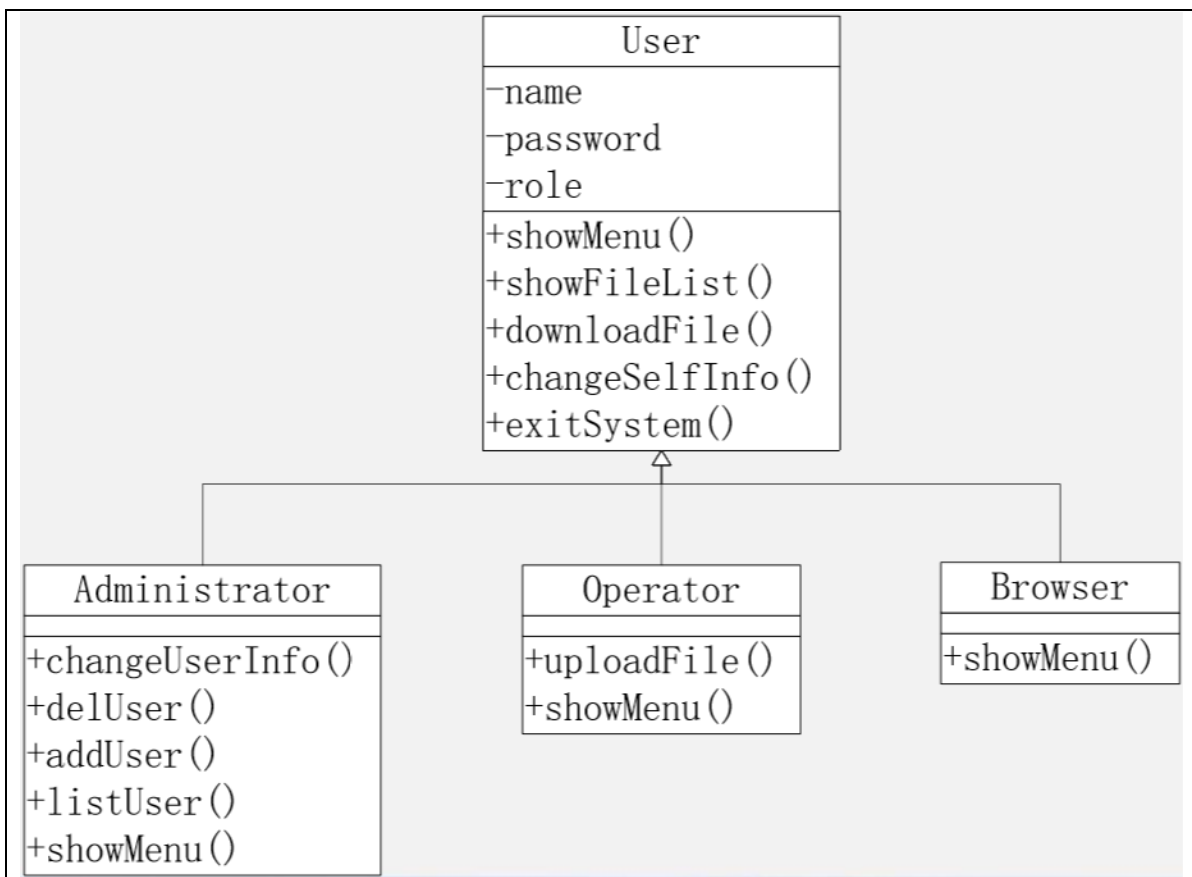
在这部分的操作中拥有最多的功能，除了能够对文件进行操作同时还能够对系统中的用户进行相关的操作，例如有修改用户，删除用户，显示用户的列表等。在对用户列表进行显示的时候是通过对哈希表中的相应的元素进行枚举，然后将这些元素一一输出，使用户列表能够按照相应的格式进行输出。

1.3 模块设计

首先本次的实验是为下面的多次的实验进行框架的搭建，在实验的过程中我选择的是按照老师的提示进行的设计。首先实现一个抽象的 User 类，然后由这个类派生出 Brower 类，Operator 类和 Administrator 类。他们之间满足下面的关系：

1. User 类为父类，操作员，浏览员和系统管理员都是他的子类；
2. 子类继承了父类的相关的属性和方法，例如：文件下载，菜单显示等；
3. 同时子类中增加了一些自己的个性化的方法，例如文件上传；
4. 子类对菜单显示方法进行覆盖；

具体的依赖关系如下图：



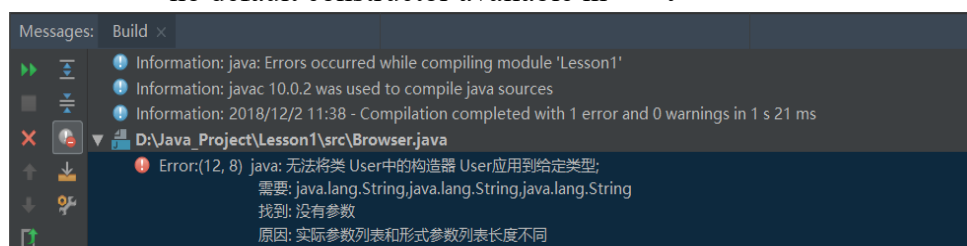
在编码实现的过程中，首先是在工程目录中新建了这 4 个类，另外增加了教学资料中提供的 **dataprocessing** 类，(这个类利用哈希表模拟了系统中相关信息的存储)。然后在编码实现的过程中完善了个各类的相关的功能，本次的实验中由于父类中相关的方法已经给出，因此只需要实现各个子类中自己的个性化的方法，以及对菜单显示方法的覆盖。

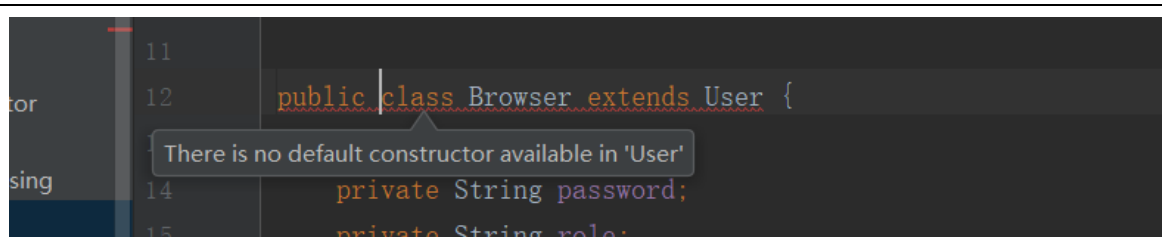
另外在实验中还需要设计一个主类用于处理登录。然后创建一个 **User** 对象 **temp**，通过用户名和密码的匹配，利用多态使 **temp** 向上转型，指向一个具体的子类，然后通过子类的引用给父类的 **temp** 赋值，之后便通过了登录验证能够显示正确的菜单选项。

1.4 开发难点与体会

● 解决子类的构造函数问题

问题描述：在创建了一个子类后，在子类的构造函数处编译器报错显示：**there is no default constructor available in *****。





原因：父类中没有无参的构造函数，子类在继承的时候，构造函数必须要显示调用父类的构造函数，同时进行相关的参数的传递。

解决方案：在子类中定义带参构造函数，同时调用父类的构造函数。

● 字符串的比较

问题描述：在开始设计的阶段进行登录的时候的口令验证中我使用的“==”进行的字符串的比较，导致不能登录成功。

原因：在 Java 中使用==比较相当于是比较的两个字符串对象的地址，即使他们的内容相同也会返回 false,因为他们的内容不同。

解决办法：Java 中必须使用 string1.equals(string2)来进行判断，这样比较才是进行的字符串的内容的比较。

● 用户列表的设计

问题描述：在管理员中需要进行用户列表的显示，只能显示系统中一个用户的相关信息，不能正确的显示其他用户的信息

原因：哈希表中相关的元素是一个无序的状态，可以调用相关的方法进行哈希表中元素的列举，通过上网查询，我获取到了解决这一问题的方法

Enumeration（列举）

public interface Enumeration<E>实现 Enumeration 接口的对象，它生成一系列元素，一次生成一个。连续调用 nextElement 方法将返回一系列的连续元素。

例如，要输出 Vector<E> v 的所有元素，可使用以下方法：

```
for (Enumeration<E> e = v.elements(); e.hasMoreElements();) {
```

```
    System.out.println(e.nextElement());
```

这些方法主要通过向量的元素、哈希表的键以及哈希表中的值进行枚举。枚举也用于将输入流指定到 SequenceInputStream 中。

解决方法：通过应用找到的方法用到程序中的具体的体现如下：

```
//枚举哈希表中的值，用来输出系统中的人员信息
for(Enumeration<User> e=DataProcessing.getAllUser();e.hasMoreElements();) {
    User user=e.nextElement();
    System.out.println(user.getName()+"\t\t"+user.getPassword()+"\t\t\t"+user.getRole());
}
```

1.5 实验总结

这次的实验是这门课程的第一次实验，本次的实验中总体来说实现还是比较的顺利，在浏览员和操作员的类的设计中基本没有遇到什么太大的困难，主要的问题就集中在管理员和主类的编码问题上。同时这部分的问题都与对哈希表的相关的操作不熟悉有一定的关系。同时在编码的过程中我发现自己在继承和多态方面的相关的构造函数的设计等等问题还不是特别的熟悉。

同时本次的实验也是这学期的 Java 的第一次正式的上机实验，在这次的实验中我也获得了一定的经验，例如编码的过程中要注意函数的返回值，注意相关的属性要用根据他的功能进行封装，充分的体现面向对象的特点。