|  |  |
| --- | --- |
| **学号** |  |

****

**《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

# 一、要求

# 1 实验目的

运用Java语言，以迭代方式逐步编程实现一个小型档案管理系统。由此了解软件开发的一般过程，深入理解面向对象语言的基本概念和基本原理，理解和掌握继承与多态、异常处理、输入输出流、GUI设计、JDBC数据库操作、网络编程、多线程等技术；熟练掌握在Java语言环境下，上述技术的具体实现方法。

# 2 系统功能与描述

[请描述你已编程实现的系统功能。如果有改进、完善或者自己设计的扩展功能，请重点描述]

# 3 模块设计(完整源码请见光盘)

[请给出3个或以上模块设计，要求运用文字、流程图、或伪代码等写出模块的详细过程性描述，包括模块使用的数据结构、模块的接口（对系统外部的接口、对系统内部其它模块的接口、以及模块输入数据、输出数据、用户界面），并且给出模块运行的实际效果截图]

# 4 开发难点与体会

[请描述开发中你所遇到的具体困难、解决办法和体会]

# 5 实验总结

[请给出你对本次实验的感想和总结]

[另：整份实验报告小4号中文宋体、英文Times New Roman打印，不含封面封底不少于4页]

|  |
| --- |
| 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；      1. 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；     1. 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     .  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    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抱错显示：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  相关的信息，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    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：   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。 |

|  |
| --- |
| 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系统功能描述 在第一次功能实现的基础上进行迭代开发，对可能产生异常的部分进行异常的捕获和处理，是程序在遇到一些异常的时候能够正确的处理不会造成程序的崩溃。 2.3模块设计 本次为第二次的实验主要对于异常进行处理，实验中通过数学函数随机产生一些异常，因此要对所有产生可能产生异常的部分进行检查，避免程序的崩溃。  在这部分的处理过程中我主要按照异常的种分类对异常进行处理   1. 其他Expception子类   这类异常是由于程序外部的问题引起的异常，这类异常在语法上要求必须进行处理程序才能够正常的运行。  本次的实验在第一次的实验的基础上进行迭代，同时使用了教学资料中提供的相关部分的代码。在运行程序之前直接通过编译器的提示对这类异常进行处理使程序能够正常的运行不会产生语法上的错误；   1. RuintimeException及其所有子类：   这类异常的特点是在语法上不强制要求程序员进行处理，因此编译器也不会直接产生提示。  在处理这种类型的异常的时候首先我是对几个比较明显的会产生异常的位置进行了处理，然后就是通过不断的运行程序，找出其他会产生异常的代码然后使用相关的方法进行处理。 2.4开发难点与体会 说明本次实验中的异常采用的是数学随机函数随机产生的异常对实际情况的模拟。开发的过程中主要出现了下面的错误：   * 错误信息     原因：  根据编译器的提示显示出来的问题是存在未处理的异常，这是在编译之前编译器提示的异常，属于程序的外部问题引发的异常，是要求程序员必须要进行处理的异常。  具体到程序当中这是由于无法连接到数据库等原因引起的    处理办法：  在可能产生异常的位置使用try…catch…finally结构进行异常的捕获和处理。   * 错误信息   C:\Users\13738\AppData\Roaming\Tencent\Users\1373836603\TIM\WinTemp\RichOle\0YU]$5VLKCI@RV5HI6NG(5G.png  原因：  这里出现的结果同一次报错的原因是一样的，都是由于代码中一些必须处理的异常没有进行合适的处理。  解决办法：  这里的解决办法同样采用的是java中的try…catch结构对异常进行捕获处理。   * 错误信息     原因：  这是在编写代码的过程中编译器自动产生的提示，首先我认为可能是这里的异常的对象同上面的异常的对象命名时同名产生的问题，但是将异常对象的变量名修改之后依然有错误。  将报错信息复制到网上找到原因在try这个语句块中并不会产生SQLException异常，因此在这里用catch去捕获这个异常时没有意义的。  解决办法：对后面的异常语句的捕获进行删除，不用catch 去捕获没有意义的异常。   * 错误信息   D:\TIM\MobileFile\Image\RXIHZ@PD7$E(}O}4_I{UJN5.png  原因：  这是在程序运行时出现的错误，根据显示的结果这是由于没有正确的连接到数据库，由于这里还没有添加数据库，产生异常也是由于数学函数随机产生的异常。同时可以根据返回的信息定位到产生异常的位置为主函数中的38行。因此在主函数的38行增加了异常的处理的代码。  在增加了异常处理部分的代码之后依然存在的问题是，程序能够正确的抛出异常，但是对异常缺乏很好的处理，显示了相关的异常信息之后程序依然终止没有实际的意义。    在此进行改进的过程中使用了完整的try…catch…finally语句。    此时程序能够正确的处理异常，同时程序并不会中断，而是可以继续接受用户进行的下一次的输入，同时产生下面相应的菜单。 2.5实验总结 本次的实验比较的简单，在第一次的基础上增加了异常处理的语句块，对模拟产生的异常进行了捕获处理。首先在本次的实验中直接利用编译器的报错对所有的程序外部产生的异常进行捕获处理是程序能够达到能够编译的阶段。  然后由于本次的异常时通过数学随机函数进行的模拟，因此通过反复的运行调试可以确定程序运行时的异常，通过阅读给的相应的代码确定可能产生异常的位置进行捕获处理。同时在本次的实验中收获的经验是我们在检查异常的时候并不是进行的漫无目的的检查，在处理相应的部分的时候应该分析可能那些位置会产生异常，会产生什么异常，然后有目的的检查，不需要进行没有意义的一些检查。增加没有意义的还可能引起语法上的错误。  同时记录一下本次实验中用到的产生异常的随机函数：  Math.Random  调用这个Math.Random()函数能够返回带正号的double值，该值大于等于0.0且小于1.0，即取值范围是[0.0,1.0)的左闭右开区间，返回值是一个伪随机选择的数，在该范围内（近似）均匀分布。 |

|  |
| --- |
| 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系统功能描述 在第三次的实验中利用流的方式进行文件的读写，首先本次的实验中增加了一个Doc类，主要作为获取文件的文件名，文件拥有者，文件的属性等等信息的初始化和文件信息的返回。  本次的实验中主要进行操作的就是操作员的上传和下载的模拟，另外就是要进行文件列表的显示，在执行相关的菜单操作的时候进行文件的相关的信息的输出。  档案模块的操作可以概括如下：   1. 档案查询:根据系统中已经有的数据，浏览已有的档案信息 2. 档案下载：根据相关用户的权限，实现文件的下载功能，同时将下载的文件保存到指定的目录中 3. 档案上传：档案录入人员专用，实现档案的上传，本次实验在本地进行数据库的模拟，从指定的目录实现文件的上传。  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的信息就进行下载，下载过程及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   D:\TIM\MobileFile\Image\}SAVLKU(~R5]W(G@XC%5OPV.png  分析：  首先通过按照流的方式进行文件的的读写，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。    然后通过运行确定了抛出异常的位置：  D:\TIM\MobileFile\Image\TPY2P7LV1550SWP{4U}@FHN.png  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。  D:\TIM\MobileFile\Image\%_~91T9KM0)~R}LBB3WF1`S.png  D:\TIM\MobileFile\Image\DBP2TTTQ1PVBA(9O`F~DCAC.png  最终通过调试却定这是FileNotFound抛出的异常。然后我对中断的地方进行了检查，查到是文件的路径设置的问题。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的问题，虽然这个问题并不是特别的严重，但是在调试的过程中却花了我大量的时间才定位到了问题。  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：   D:\TIM\MobileFile\Image\VZ059CFU%Z~}CLMHGQ298LY.png  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一下    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，解决而问题。  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   D:\TIM\MobileFile\Image\ZHNCC@GXO88{3_1~{Y~U~HK.png  D:\TIM\MobileFile\IMG_7107.PNG  这里我实现了文件的下载但是通过检查下载的文件我发现的问题是，我下载保存的文件不是完整的，这是部分文件。从文件的行号来看是缺少了9行，更加具体的是文件从2k变成了1k。  原因：  通过对文件大小的具体的变化我查找到死在进行文件的读写的时候使用的字节数组的大小为byte[1024],所以文件下载下来就变成了1k。  解决办法：    最后我的解决办法是根据课件上提供的方法，在文件进行读取的时候，对指定的文件读取文件的全部字节。   * 错误信息： D:\TIM\MobileFile\Image\GRR_7VCJNR0C$Y%LVYM%2L6.png   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时间戳的相关的信息。    <https://www.cnblogs.com/zhujiabin/p/6168671.html?utm_source=itdadao&utm_medium=referral>  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   3.5实验总结 本次的实验是关于输入输出流相关的内容，关于这部分的内容在Java程序课上并没有进行太多的讲述，只是在大一的C++面向对象中对流的概念有一定的了解。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。 |

|  |
| --- |
| 4.GUI图形用户界面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系统功能描述 本次的实验需要进行实现的是Gui图形界面需要进行实现有：登录界面，操作员界面，管理员界面，浏览员界面。功能实现完成后的效果图如下：     4.3模块设计 本次实验的模块的设计和实验一的基本相同就是将控制台的输出更改为GUI图形界面，因此在编码的过程中我是针对前几次的实验进行的修改，修改的部分主要包括主函数中的登录界面的编码，还有系统中三种身份的人员的代码的编写。  运行程序首先通过主登录界面接受用户输入的用户名和口令然后针对输入的口令进行检查，确认系统中是否存在这个用户然后根据身份显示下一个界面的用户的操作菜单。  相关的具体的功能的实现通过对对应的事件的监听来实现。 4.4开发难点与体会  * 解决Java Swing中中文乱码的问题     解决办法：  通过上网查找相关的资料我了解到之所以产生这个问题可能是编译器的原因而不是编码过程中的问题，应该是编译器的编码的问题。我根据网上给出的办法顺利的解决了资格问题，详细的解决办法如下：     * 不同类型的对象同名造成的问题：       解决办法：  这里根据编译器产生的提示直接却定了问题的所在是因为产生两个不同的对象，但是这两个对象的命名时相同的。结果调试发现是因为在进项异常处理的时候同样将异常对象的命名为e产生的报错。通过将异常对象变量更改为ex顺利的解决了这个问题。   * 空指针问题：       解决办法：  这里通过调试确定问题是因为产生了空指针的问题，经过调试确定是在之前的事件的监听造成了frame为空。   * 对话框的显示问题     这里出现的问题不是编写代码上的问题但是这类问题的解决也需要花费大量的时间，这种问题就是类似上面的的显示界面的对齐问题在程序中虽然能够进行显示但是显示出来的确实没有完全对齐的状态。  解决办法：  这类问题在本次的实验中产生的很多，包括尺寸问题，显示位置问题，布局问题等等，对此目前我也没有找到很好的解决办法只能有耐心的一步一步的不断尝试进行更改，直到找到合适的尺寸和显示。 4.5实验总结 本次的实验内容是GUI图形界面设计，由于之前不论是C++还是其他的程序设计我们写出来的都是控制台程序，对于GUI图形界面是一个全新的内容。由于之前没有接触过相关的内容同时本次的实验中需要进行编写的代码的量也比较的大，因此本次的实验我也是花了较多的时间，花了较大的功夫才对一些基本的内容进行了实现，同时在我编写的程序中还存在着一些瑕疵还有待进一步的改进。  通过本次的实验我也有了不少的收获。首先这是我第一次编写图形界面的程序虽然在编写的过程中遇到了不少的困难，但是最终都通过查找相关的资料进行了解决，最终能够完成这个界面自己也有一定的成就感。第二就是通过本次的实验对于java中Swing的有了一个基本的了解。知道如何在一个界面中添加相关的组件同时对相关的事件进行处理实现自己的功能。对于这些组件的相关的协调组织在后面的学习中也还要进行更加深入的学习争取让做出来的界面也更加的完善美观。最后通过本次的实验我也对Java有了更加深入的理解，对了编写这类程序java给我的感受就是虽然实现的时候并不是特别的复杂但是需要进行的编码的工作可能比较的大。由于没有其他方面的相关经验这里我也无法进行横向的比较，只能在通过后面的学习进行深入的理解。 |

|  |
| --- |
| 5.JDBC数据库操作面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系统功能描述 本次的实验将原先的哈希表中的文件信息和用户的信息组织，通过实现编写的数据库脚本文件输入实验的数据库中的两个表doc\_info和user\_info。然后通过编写Java代码实现数据库的相关的信息的增删查改等操作。改进的之前的档案管理系统的数据信息保存在哈希表中，每次运行程序之后不能动态的更新信息的缺点。  数据库中两个表的组织情况如下：    主索引分别为文件Id和用户的Username。      上面的截图为两个表中的数据的具体组织 5.3模块设计 本次的实验中为了最大限度的利用之前说编写的代码保留了有关数据操作的Dataprocessing这个类以及类中的所有接口，如图：    只是对相关的方法由之前的哈希表的操作更改为了数据库的操作。然后在主函数中进行数据库的连接与断开实现对程序的控制。  下图分别为数据库的连接与断开：     5.4开发难点与体会 由于数据库相关的课程我们还没有进行学习，数据库相关的操作对于我们来说还是新的东西，需要从开始的数据库的下载和安装进行。在这次的实验中我碰到的最大难点是数据库的安装，最开始的数据库的连接，后面数据库的几个基本的操作增删查改的编程实现到时没有太大的困难，熟悉了一个之后操作的基本就是套路。   * 数据库的安装     进行数据库的操作之前首先是要下载安装数据库同时还要安装Java相关的驱动。  这是目前的最新的版本，但是通过实验最新的版本我发现在程序中无法实现数据库的连接。为了结局这个问题我只好按照课件上给的相关的方法重新安装了一个较低版本的数据库。  重新安装了数据库之后能够使用教学资料中的提供的Mysql\_font软件，也成功将文件输入。     * 数据驱动错误   安装好了数据库并输入了sql文件之后使用了教学资料中提供的sqlteat文件进行了数据库的连接测试但是这次又遇到了新的问题    解决办法：  实验中sqltest这个类中抛出的异常时数据驱动问题，通过于都相关的资料了解到可能是重新安装了低版本的数据库但是依旧是使用的之前的数据驱动文件。  为了解决这个问题我又从mysql的官网上下载了较低版本的驱动文件，再次进行测试解决了这一问题。   * 数据库错误     解决办法：  为了确顶是测试程序的问题还是数据库本身的问题，首先我排查了数据库连接是需要用到的用户名和密码都没有错误。  然后我数据库入手进行了相关的检查。为了方便检查我使用了开发环境中集成的数据库配置的先关的工具进行了测试，通过连接测试相关的功能的检查发现数据库并没有存在什么问题，能够成功的进行连接，测试的结果显示如下。  确定了实验中的数据亏没有什么问题之后我检查了程序相关的部分，发现是连接相关的url在编写时发生了错误。  D:\TIM\MobileFile\Image\D_$SX)G2H4IN(8NZR]4C@JC.png   * 警告解决   D:\TIM\MobileFile\Image\6SQCSA3LQK}Q19Z44H7OST0.png  结局了上面的两个问题成功的连接到了数据库但是数据库抛出了一个警告，为了由于警告的内容太长，我直接百度解决了这个问题。  解决办法：声明数据库的url中增加参数。     * 转型错误     这是在dataprocessing类中出现的一个错误，由于这个类为抽象类不能进行实例化，于是我才用了迂回的办法进行了解决根据身份进行实例化。   5.5实验总结 本次的实验是关于JDBC数据库的相关的操作，通过这次的实验，首先是在自己的电脑上安装好数据库下载相关的驱动。虽然在之前并没有学习过数据库的相关的知识，但是在编写相关部分的代码的时候并没有遇到太大的困难。一是因为本次的相关的操作比较的简单，主要涉及到的就是数据库的基本的内容，数据库的相关的增删查；另外一个原因就是我在这部分中使用的方法相对的单一，主要通过静态sql语句进行实现。  通过本次的实验也是对数据库有了一个基本的了解，在后面的学习中好需要对数据库进行更加深入的学习。  **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的客户和服务器编程方法。  **6.2 系统功能与描述**  编写程序，将前面课程所编写的档案管理系统从单机版改编成为客户机/服务器模式，实现档案文件在客户机和服务器之间的上传、下载。  **6.3 模块设计(完整源码请见光盘)**  本次的实验的模块设计包含两个部分，分别是基于Socket的客户端和服务器，在与运行程序的时候首先运行服务器程序，然后运行客户端程序，通过客户端与服务端的连接实现数据的传输：   1. 服务端的设计   主要实现了客户端对特地昂的端口进行监听，等待客户端发送来的信息；在这个程序中客户端发送的为文件信息，服务端受到文件信息之后对文件进行存储   1. 客户端的设计   客户端根据IP地址和端口号连接服务器端，从磁盘中读取文件的信息，发送到服务器端，然后接收服务器端返回的信息，最后结束会话。  整个过程可以用下面的流程图来进行描述：    本次的实验中，根据前面的几次的程序进行迭代，通过客户端实现了文件的下载，由于时间有限数据库部分使用的之前的代码为进行修改，将数据库的管理放置在本地。  **6.4 开发难点与体会**   * 服务端的连接问题   D:\TIM\MobileFile\IMG_7431.PNG  解决办法：  这里根据编译的提示是连接问题，我根据下面的调试的步骤进行了解决：   1. 首先我检查的服务器端的程序的是否运行，确认进行连接之前的服务器程序是运行的状态； 2. 对于传输的端口进行检查，需要确定使用的端口不是系统运行时会占用的，通知需要进行确认的是服务端的端口和客户端的端口嗜血要进行匹配的   通过按照上面的步骤进行检查最终确定是我的服务端和客户端的程序的端口使用没有进行匹配；   * 空指针问题   D:\TIM\MobileFile\IMG_7552(20190101-171808).jpg  解决办法：  这里通过编译器的显示提示的是一个空指针的问题，需要对此进行解决首先是要找到具体是哪一个对象的值是空的然后进行解决；  在调试的过程中具体使用的二分法的方式进行的调试，通过编译器的调试的功能确定程序的哪一个对象的值是null然后对齐进行赋值就能得到解决。   * 数据库信息写入问题：     在文件的上传的过程中实现了文件的上传，能够在系统的指定的位置找到上传的文件但是文件的信息并没有在数据库中进行显示，通过相关的检查得到是由于没有在服务端进行数据库的相关的操作。  这里为了简单期间我对功能进行了简化，直接将数据库的相关的操作放在了本地进行实现，具体的实现的步骤如下：   1. 在客户端进行文件的信息的上传； 2. 上传完成之后直接在客户端对数据库的信息进行更新实现了数据库的信息写入。   通过这样的设计虽然能够完成对于文件信息的写入但是对于文件的保存时候的文件的命名会带来一定的影响。 6.5实验总结 本次的实验是关于物联网络编程的部分，具体来说是要实现文件的上传功能，将原来的文件上传功能从本地改为数据库的上传，就是通过网络实现文件信息在客户端和服务端的通信。由于这部分的内容在之前完全没有接触过因此本次在编程的时候不知道从何处进行下手，在编写代码的时候也遇到了不少的问题。在实际的操作的过程中通过上网查找相关的资料阅读老师给的示例程序最终对文件上传这一简单的功能进行了实现。按照文件流的方式在两个程序之间实现了文件信息的传输。  **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**  编写程序，将前面课程所编写的档案管理系统的服务器端改写为多线程方式，使得其能同时处理多个客户端的请求。  **7.3 模块设计(完整源码请见光盘)**  本次的实验为多线程设计，在具体的设计过程中主要针对服务端进行修改，将服务器端的程序修改为能够同时处理多个客户端请求。设计的过程中主要对多线程的始终状态进行设计：    在线程的具体的实现的过程中本实验中才用的是继承Thread类的方式实现。  **7.4 开发难点与体会**   * 连接问题解决     解决办法：  通过报错的信息我们可以大概的分析出来这部分的问题因该是由于网络编程引起的而不是由于多线程的实现的问题，因为从报错的信息我们可以看出错误的内容与Thread并没有太大的关系。  从这个角度来看可能是数据库的问题或者是网络连接的问题，通过调试最终确定是数据库的连接部分除了问题，通过多数据库部分的修改成功的解决了这个问题。   * 多线程没有具体实现的问题     解决办法：  这里通过提示的信息可以看出来我们的多线程并没有具体的实现，在进行连接的时候无法实现多个客户端的登录，这里可能是关于登录的问题或者是服务端的多线程的处理问题。  通过调试确定是线程的start()方法那你可能出现了问题，导致不能实现多线程，通过重新编写那部分的程序成功解决了问题。   * 服务问题     解决办法：  这里通过上完查询这个异常的问题了解到：  导致“Connection reset”的原因是服务器端因为某种原因关闭了Connection，而客户端依然在读写数据，此时服务器会返回复位标志“RST”，然后此时客户端就会提示“java.net.SocketException: Connection reset”。  具体的解决办法如下：  1.出错后重试  2. 客户端和服务器统一使用TCP长连接  3. 客户端和服务器统一使用TCP短连接 7.5实验总结 本次的实验是文件管理系统的最后一次的实验，是关于多线程的，通过本次的实验的改进文件系统的功能已经比较的完善了，能够同时处理多个客户端的请求实现文件的上传和下载，具有了实际的意义。本次的实验中重点运用的是关于线程的两种实现的方法和线程处理的基本的过程。其中包括了线程发的5中状态的相关的详细的运用，关于这部分的实验，虽然通过查找相关的资料实现了，但是自己自运用的方面还不是特别的熟练，线程的调度方面掌握的不是特别的好，在后面的学习和运用中还要加强这方面的熟练掌握。 |

**《面向对象与多线程综合实验》成绩评定表**

班级：　计算机类m1704 姓名：颜道江　学号：0121710880415

|  |  |  |  |
| --- | --- | --- | --- |
| 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：

　　　　　　　　　　　　　　 　　2019 年 01 月 07 日