

**面向对象程序设计上机实验报告**

实验题目4： 数据结构的综合运用

**学院名称 智能与计算学部**

**专 业 软件工程**

**学生姓名 王杨森**

**学 号 3020244116**

**年 级 2020级**

**班 级 软件工程3班**

**时 间 2022年 04月28日**

1. **实验目的**
2. 熟悉Eclipse/MyEclipse；
3. 熟悉Java中各种类的综合使用；
4. **实验内容**

现在所有学生的名单students.txt，其中每行为一个学生信息，包括学号、姓名、班级，以tab符号分割(\t)，学院要求所有同学把自己一寸照片发给辅导员，图片命名规则为 “学号\_姓名.jpg”，每个照片文件不得超过128K（124\*1024 byte）现在存在下列问题：1 有的同学没有交照片；有的同学照片超过128K。请用编程的方式帮助辅导员解决如下问题：

1. 找出那些同学的照片没有交照片；
2. 找出那些同学照片文件超过128K了。
3. **程序实现**

**public** Set<String> copyToTargetDirAndReturnNoExist(String studentListFile,String srcDir,String target) **throws** Exception {

Set<String> noExistSet = **new** HashSet<>(); // 不存在的学生集合

List<String> existIdList = getExistsIdList(srcDir); // 已经存在的学生

List<String[]> nameList = getNameList(studentListFile); // 学生姓名

**for** (String[] strings : nameList) {

String id = strings[0];

String name = strings[1];

String classes = strings[2];

File targetDir = **new** File(target, classes);

**if** (!targetDir.exists()) { // 如果还没这个班级目录，创建之

targetDir.mkdirs();

}

**if** (!existIdList.contains(id)) {

noExistSet.add(id); //如果没有这个学生的照片，添加到不存在的集合中

} **else** {

String src = srcDir + id + ".jpg";

String dst = target + classes + "/" + id + "\_" + name + ".jpg";

*copyFile*(src, dst); // 文件拷贝

}

}

**return** noExistSet;

}

/\*\*

\* 从文件中读取学生列表到List中，List的元素为String数组

\*/

**public** List<String[]> getNameList(String fileName) **throws** IOException {

List<String[]> res = **new** ArrayList<>();

**try** (Reader reader = **new** FileReader(fileName);

LineNumberReader lineReader = **new** LineNumberReader(reader)){

String line = lineReader.readLine();

**while**(line != **null**) {

res.add(line.split("\t"));

line = lineReader.readLine();

}

}

**return** res;

}

/\*\*

\* 获取已经上传了照片的学生名单

\* \*/

**public** List<String> getExistsIdList(String filename) {

List<String> res = **new** ArrayList<>();

File file = **new** File(filename);

File[] lists = file.listFiles();

**assert** lists != **null**;

**for** (File thisFile : lists) {//运用正则表达式进行字符串匹配

**if** (thisFile.isFile() && thisFile.getName().matches("[0-9]{10}.jpg")) {

res.add(thisFile.getName().substring(0, 10));

}

}

**return** res;

}

/\*\*

\* 复制IO流

\*

\* **@param** in

\* **@param** out

\* **@throws** IOException

\*/

**public** **static** **void** copyIO(InputStream in, OutputStream out) **throws** IOException {

**byte**[] buf = **new** **byte**[*CHUNK\_SIZE*];

**int** len = in.read(buf);

**while**(len != -1) {

out.write(buf, 0, len);

len = in.read(buf);

}

}

/\*\*

\* 复制文件

\* **@param** src

\* **@param** dst

\* **@throws** IOException

\* \*/

**public** **static** **void** copyFile(String src, String dst) **throws** IOException {

InputStream in = **null**;

OutputStream out = **null**;

**try** {

in = **new** FileInputStream(src);

out = **new** FileOutputStream(dst);

*copyIO*(in, out);

} **finally** {

*close*(in);

*close*(out);

}

}

/\*\*

\* 关闭一个输入 输出流

\*

\* **@param** inout

\*/

**public** **static** **void** close(Closeable inout) {

**if**(inout != **null**) {

**try** {

inout.close();

} **catch** (IOException e) {

e.printStackTrace();

}

}

}

1. **实验结果**

测试代码：

**public** **static** **void** main(String[] args) {

**try** {

/\*\*

\* 现在所有学生的名单students.txt（为了避免乱码问题，文件中不包含汉字），其中每行为一个学生信息，包括学号、姓名、班级，以tab符号分割(\t)，

\* 学院要求所有同学把自己一寸照片发给辅导员，图片命名规则为 “学号.jpg”

\*

\* 现在存在下列问题。请用编程的方式帮助辅导员解决如下问题：

\* 1) 找出那些同学的照片没有交照片；

\* 2) 在目标目录下每个班级建立一个字目录，把上交的同学的照片，统一按 学号\_姓名.jpg 方式拷贝到各自班级目录下，原来的文件不要删除.

\*\*/

// 实际测试中，文件存放位置可能改变，学生数量也可能改变

// 其中每行为一个学生信息，包括学号、姓名、班级，以tab符号分割(\t)，

String studentListFileName="D:/北小洋/21222/面向对象程序设计/OOP-Homework/src/com/huawei/classroom/student/h14/students.txt";

// 学生照片存放的目录，不会包含子目录，本目录的pic给出了存放图片的目录示例，实际测试中，pic 位置可能改变，其下面文件数量也会改变

String picDir="D:/北小洋/21222/面向对象程序设计/OOP-Homework/src/com/huawei/classroom/student/h14/pic/";

String targetDir="D:/北小洋/21222/面向对象程序设计/OOP-Homework/src/com/huawei/classroom/student/h14/target/";

MyTools tools = **new** MyTools();

// nopicsIds返回所有没有上交照片的学生ID（）

Set<String> nopicsIds = tools.copyToTargetDirAndReturnNoExist(studentListFileName,picDir,targetDir);

Set<String> test=**new** HashSet<String>();

test.add("3016218088");

test.add("3016218009");

**if**(test.equals(nopicsIds)) {

System.***out***.println("如果target下面也像这个例子一样按班级整理好了学生文件，那么就做对了");

}

} **catch** (Exception e) {

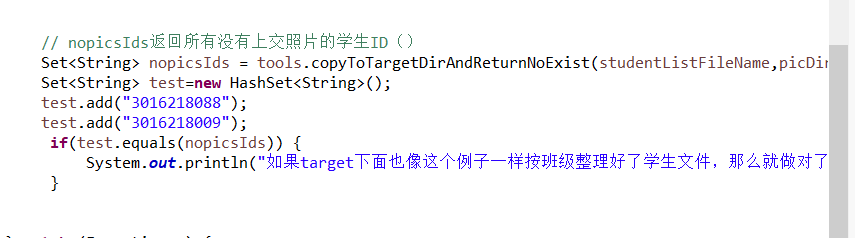
// **TODO** Auto-generated catch block

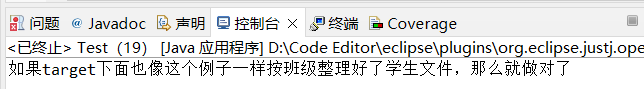
e.printStackTrace();

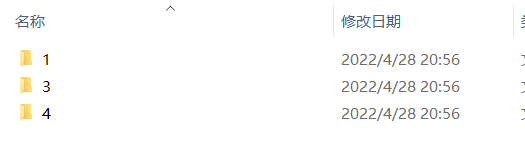
}

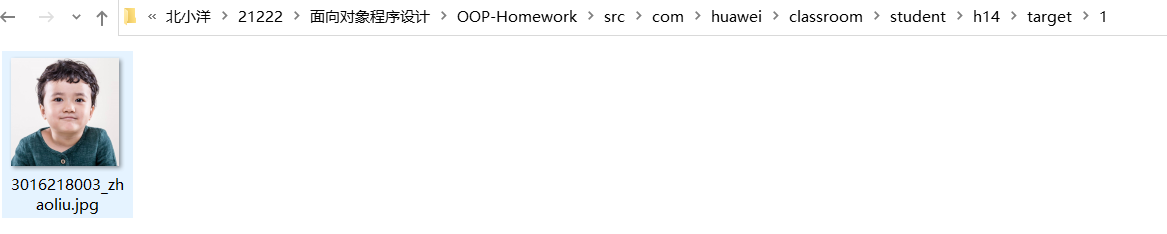
}

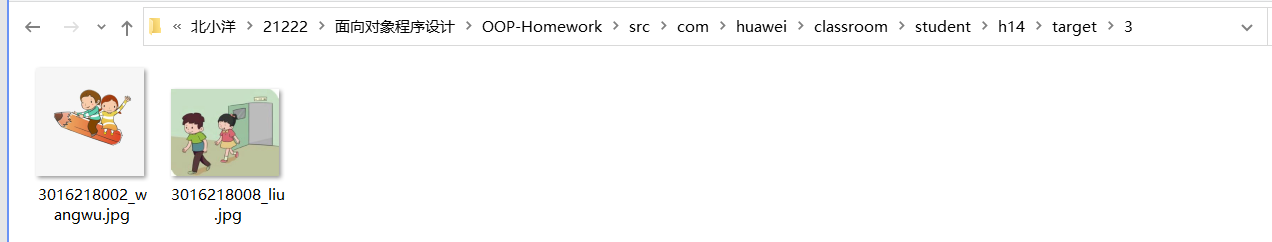
测试结果：













可以看出，可以找出没有上传照片的学生列表，且将已上传的照片进行分类。

1. **实验中遇到的问题及解决方法**

遇到的主要问题为创建文件目录时起初选用mkdir()，但在进行调试时发现会有问题，查询资料应该采取mkdirs()，并且在创建目录时不应重复创建，先查看当前目录下有无目标目录，若没有再创建。同样也要先判断当前学生是否已经上传照片，因此需要先有一个集合存储已经上传过照片的学生名单，这里可以采用读取文件的方式获取。