# 实验一 文件操作(2)

**1、文件复制**

**实验要求：使用BufferedReader及BufferedWriter实现**

查看文件复制窗体UserMonitorFrame，该类中已添加相应的控件，实现窗体布局，请编写文件复制功能方法copyFileUtil(String source,String target)，完成将source路径代表的文件复制到target代表的文件中。

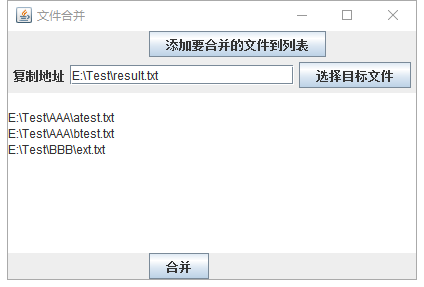


核心代码见上次内容

**2、合并多个txt文件**

**实验目的：使用BufferedReader及BufferedWriter重构上节课代码**

在项目中创建类MergeFilesFrame，点击“添加要合并的文件到列表”就可以打开一个文件选择对话框，选择好一个文件后，该文件被添加到下面列表中，用户选择合并文件后输出的目标文件路径。



单击合并按钮，则系统将会调用方法public void writeFiles(List<File> files,String fileName)，用来将用户选择的要合并的文件列表List<File> files，合并后写入到文件fileName中。

核心代码同上次内容。

**3、使用PrinWriter或者PrintStream，将一组Student对象数据写入文件student.txt中。**

public class Demo {

public static void main(String[] args) {

ArrayList<Student> list = new ArrayList<Student>();

list.add(new Student("student1",21));

list.add(new Student("student2",21));

list.add(new Student("student3",21));

list.add(new Student("student4",21));

list.add(new Student("student5",21));

list.add(new Student("student6",21));

//写Student对象数据到文件中

}

}

**4、使用ObjectOutputStream，将一组Student对象数据写入文件student.txt中。**

public class ObjectOutputStreamDemo {

public static void main(String[] args) {

ArrayList<Student> list = new ArrayList<Student>();

list.add(new Student("student1",21));

list.add(new Student("student2",21));

list.add(new Student("student3",21));

list.add(new Student("student4",21));

list.add(new Student("student5",21));

list.add(new Student("student6",21));

//写Student对象数据到文件中

}

}