

Lab 09 Assignment

班级：202111

学号：19241027

姓名：胡峻诚

Question1 如果准备按字节读取一个文件的内容，应当使用 `FileInputStream` 流还是 `FileReader` 流，为什么？
简答

应该选择 `FileInputStream`

- `FileInputStream` 流的 `read` 方法是按字节从文件中读取的
- `FileReader` 的 `read` 是按字符读取的

Question2 **简答**

1. 请写出程序的输出

```

1  import java.io.File;
2  import java.io.FileInputStream;
3  import java.io.FileOutputStream;
4  import java.io.IOException;
5
6  public class FileInputStreamTest {
7      public static void main(String[] args) {
8          File f = new File("pathname: \"hello.txt\");
9          byte[] a = "abcd".getBytes();
10         try {
11             FileOutputStream out = new FileOutputStream(f);
12             out.write(a);
13             out.close();
14             FileInputStream in = new FileInputStream(f);
15             byte[] tom = new byte[3];
16             //Part I
17             int m = in.read(tom, off: 0, len: 3);
18             System.out.println(m);//3
19             String s = new String(tom, offset: 0, length: 3);
20             System.out.println(s);//abc
21             //Part II
22             m = in.read(tom, off: 0, len: 3);
23             System.out.println(m);//1
24             s = new String(tom, offset: 0, length: 3);
25             System.out.println(s);//dbc
26         } catch (IOException e) {
27         }
28     }
29 }
    
```

运行结果：
 3
 abc
 1
 dbc

1 3
 2 abc
 3 1
 4 dbc

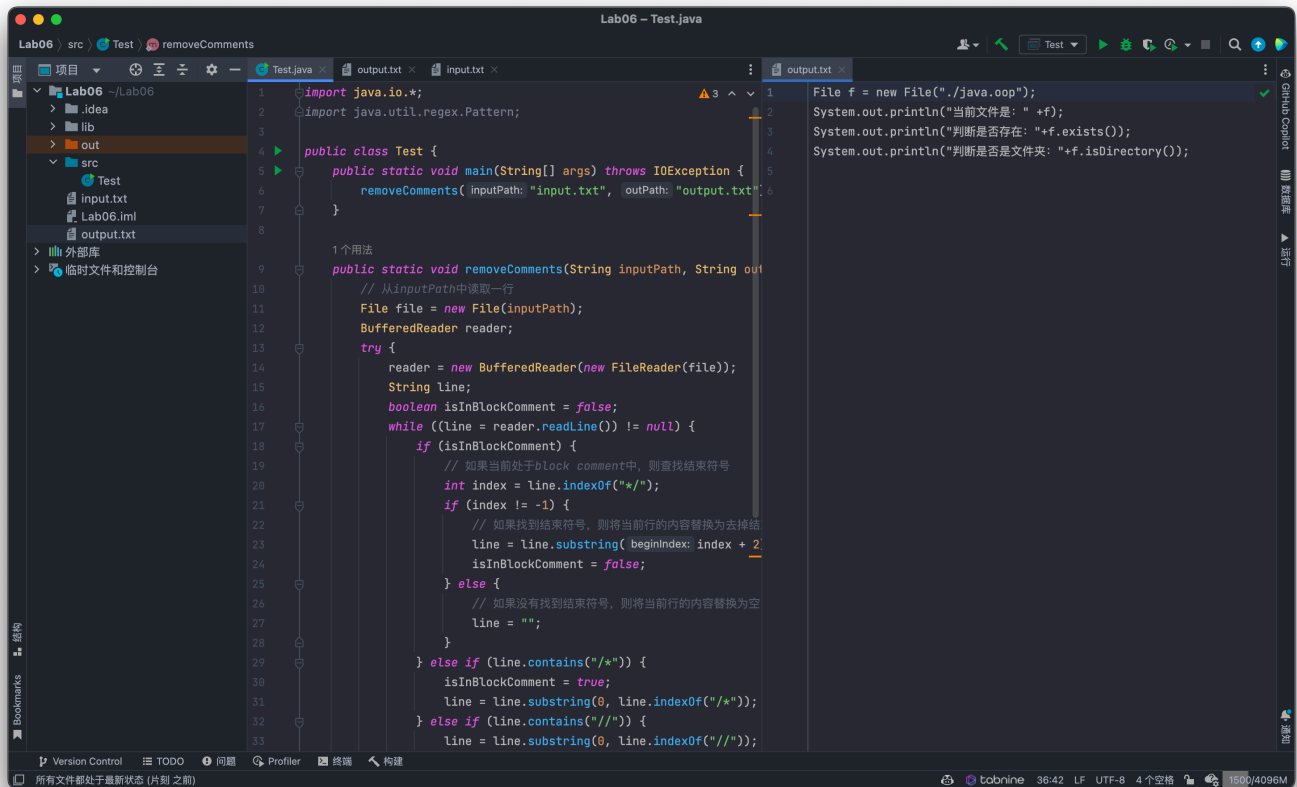
2. 解释 `Part I` 和 `Part II` 的输出为什么不同

m 的值是 read 函数的返回值，也就是读取的长度。

因为文件一共只有 abcd 四个字符，之前读了三个，所以 Part II 的读入只读进来了 1 个字符，故 m 的输出是 1。

tom 之前的内容是 "abc"，读入一个 "d" 后自然变成了 "dbc"。

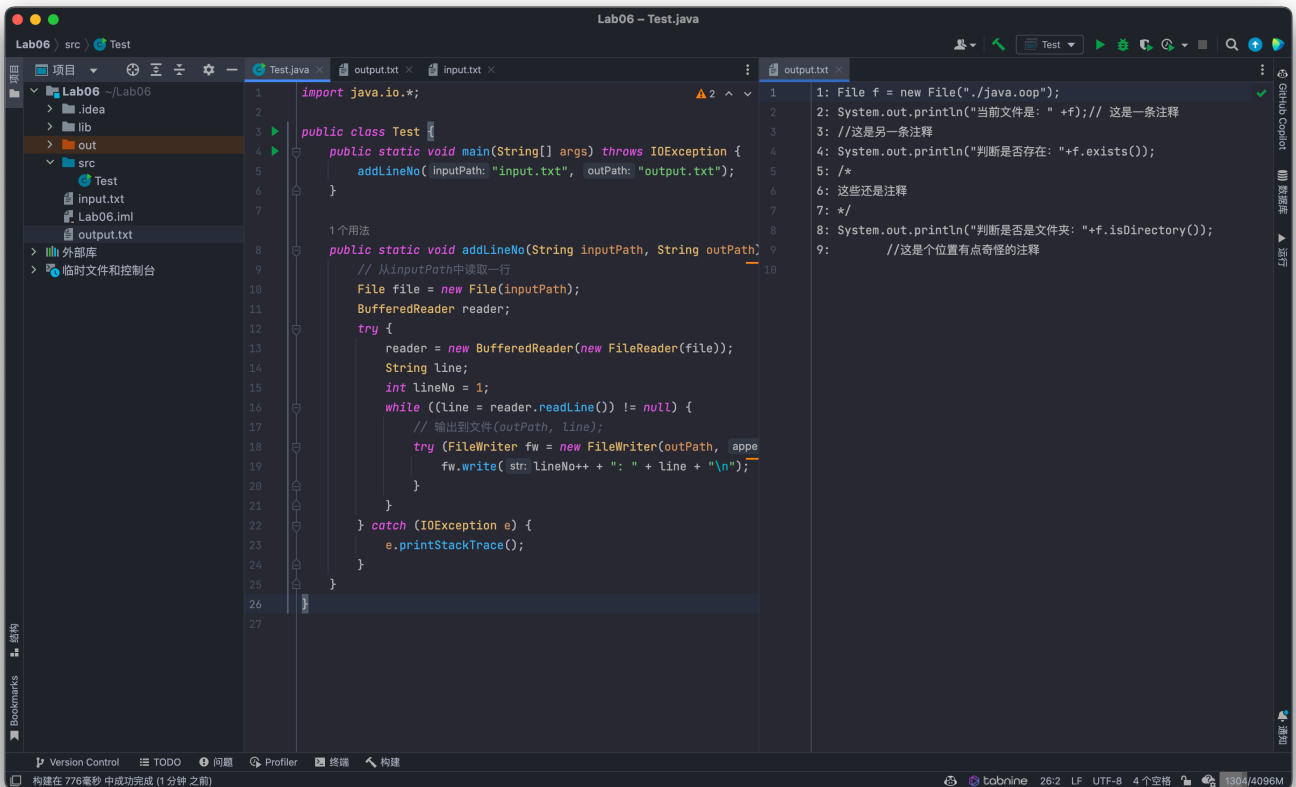
Question3 设计一个方法，用于移除文件中的注释 编程



```

1  import java.io.*;
2  import java.util.regex.Pattern;
3
4  public class Test {
5      public static void main(String[] args) throws IOException {
6          removeComments( inputPath: "input.txt", outputPath: "output.txt");
7      }
8  }
9
10 1个用法
11 public static void removeComments(String inputPath, String out
12 // 从inputPath中读取一行
13 File file = new File(inputPath);
14 BufferedReader reader;
15 try {
16     reader = new BufferedReader(new FileReader(file));
17     String line;
18     boolean isInBlockComment = false;
19     while ((line = reader.readLine()) != null) {
20         if (isInBlockComment) {
21             // 如果当前处于block comment中，则查找结束符号
22             int index = line.indexOf("*/");
23             if (index != -1) {
24                 // 如果找到结束符号，则将当前行的内容替换为去掉结
25                 line = line.substring( beginIndex: index + 2);
26                 isInBlockComment = false;
27             } else {
28                 // 如果没有找到结束符号，则将当前行的内容替换为空
29                 line = "";
30             }
31         } else if (line.contains("/*")) {
32             isInBlockComment = true;
33             line = line.substring(0, line.indexOf("/*"));
34         } else if (line.contains("//")) {
35             line = line.substring(0, line.indexOf("//"));
36         }
37     }
38 }
39
40 File f = new File("./java.oop");
41 System.out.println("当前文件是: " + f);
42 System.out.println("判断是否存在: "+f.exists());
43 System.out.println("判断是否是文件夹: "+f.isDirectory());
    
```

Question4 设计一个方法，使用 Java 的输入、输出流将一个文本文件的内容按行读出，每读出一行就顺序添加行号，并写入到另一个文件中 编程



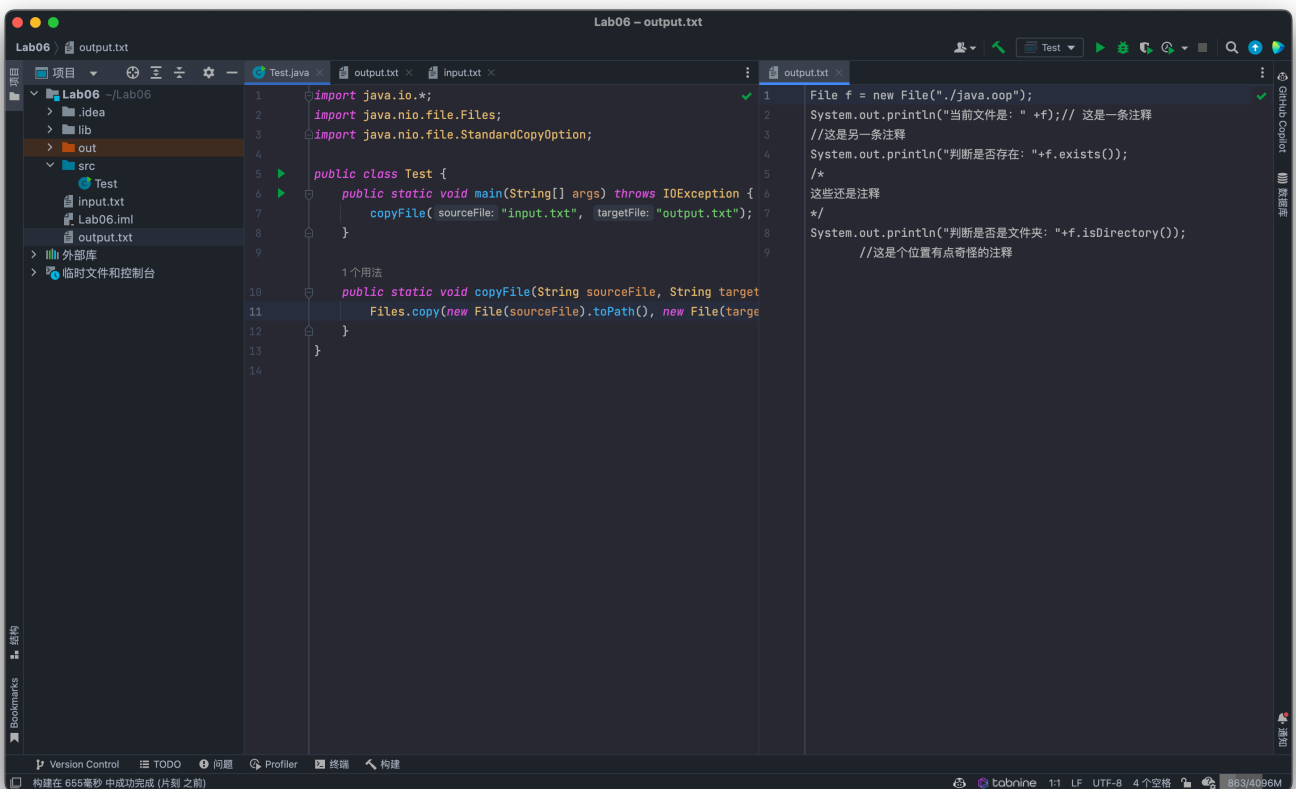
```

1 import java.io.*;
2
3 public class Test {
4     public static void main(String[] args) throws IOException {
5         addLineNo("input.txt", "output.txt");
6     }
7
8     // 1个用法
9     public static void addLineNo(String inputPath, String outputPath) {
10        // 从inputPath中读取一行
11        File file = new File(inputPath);
12        BufferedReader reader;
13        try {
14            reader = new BufferedReader(new FileReader(file));
15            String line;
16            int lineNo = 1;
17            while ((line = reader.readLine()) != null) {
18                // 输出到文件(outputPath, line);
19                try (FileWriter fw = new FileWriter(outputPath, append)) {
20                    fw.write(lineNo++ + ": " + line + "\n");
21                }
22            }
23        } catch (IOException e) {
24            e.printStackTrace();
25        }
26    }
27 }
  
```

```

1: File f = new File("./java.oop");
2: System.out.println("当前文件是: " + f); // 这是一条注释
3: //这是另一条注释
4: System.out.println("判断是否存在: "+f.exists());
5: /*
6: 这些还是注释
7: */
8: System.out.println("判断是否是文件夹: "+f.isDirectory());
9: //这是个位置有点奇怪的注释
  
```

Question5 复制文件是常见的IO操作。设计如下方法，实现复制源文件 `sourceFile` 到目标文件 `targetFile` 编程



```

1 import java.io.*;
2 import java.nio.file.Files;
3 import java.nio.file.StandardCopyOption;
4
5 public class Test {
6     public static void main(String[] args) throws IOException {
7         copyFile("input.txt", "output.txt");
8     }
9
10    // 1个用法
11    public static void copyFile(String sourceFile, String targetFile) {
12        Files.copy(new File(sourceFile).toPath(), new File(targetFile).toPath(), StandardCopyOption.REPLACE_EXISTING);
13    }
14 }
  
```

```

1: File f = new File("./java.oop");
2: System.out.println("当前文件是: " + f); // 这是一条注释
3: //这是另一条注释
4: System.out.println("判断是否存在: "+f.exists());
5: /*
6: 这些还是注释
7: */
8: System.out.println("判断是否是文件夹: "+f.isDirectory());
9: //这是个位置有点奇怪的注释
  
```

Question6 复制一个文件夹下面所有文件和子文件夹内容到另一文件夹

编程

```

1  import java.io.*;
2  import java.nio.file.Files;
3  import java.nio.file.StandardCopyOption;
4
5  public class Test {
6
7      public static void main(String[] args) throws IOException {
8          copyDirectory("input", "output");
9      }
10
11      2 个用法
12      public static void copyDirectory(String sourceDir, String targetDir) throws IOException {
13          File source = new File(sourceDir);
14          File target = new File(targetDir);
15
16          if (source.isDirectory()) {
17              if (!target.exists()) {
18                  target.mkdir();
19              }
20              String[] file = source.list();
21              for (String s : file) {
22                  copyDirectory(sourceDir + "/" + s, targetDir + "/" + s);
23              }
24          } else {
25              Files.copy(source.toPath(), target.toPath(), StandardCopyOption.REPLACE_EXISTING);
26          }
27      }
28  }
    
```