

文件说明：

-src 文件夹：代码工程文件

-htt 插装代码文件夹

-CompareIn.java 比较两个版本结果文件的输入输出文件

-CopyFile.java 比较结果不同的文件写入 DifferentResultFiles 文件

-InsertFun.java 插装函数

-InsertJava.java 对 java 文件进行插装即插入插装函数

-readDict.java 遍历 run 文件夹下的所有文件.java 进行插装

-Myjavac.java 遍历 run 文件夹下的所有文件.java 进行编译

-Myjava.java 遍历 run 文件夹下的所有文件.class 进行运行

-SpiderDemo1-3.java 对相应的网址爬取源代码

-run 存放要插装并编译运行的所有实验源文件（包含 InsertFun.java）

-coyfile.txt,用于插装时临时存储的源文件.

-源代码：爬取的测试用的源代码文件（包含 InsertFun.java）

-ResultFiles 文件夹：存放两个版本产生的所有结果文件

-DifferentResultFiles 文件夹：存放有差异的结果文件

-DifferentInput 文件夹：结果文件的输入文件不同

- DifferentReturn 文件夹：结果文件的返回值文件不同

-htm 文件夹：存放爬取的代码

-code1-3 存放 SpiderDemo1-3.java 对相应的网址爬取源代码

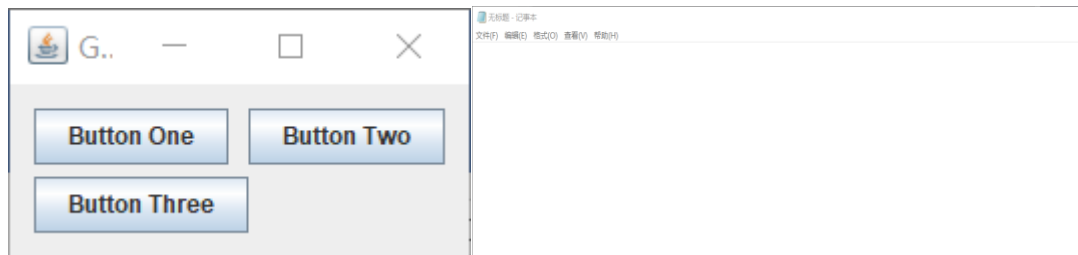
-URL1-3.txt :存放爬取的网址.

注：重新运行 Myjava.java 时，要把 ResultFiles 文件夹中生成的对应的版本删除，避免相同名字的结果文件写入两次.还有 DifferentResultFiles 文件

注：若 Myjava.java 很快结束，就要检查 javac 是否执行全部.java 文件，若没有全部生成.class 文件，就重新运行 javac。若已经全部生成.class 文件就重新运行 Myjava

注：Myjava.java 执行过程中生成子程序如打开记事本等要及时关掉，否则会阻塞。

如下要关掉



使用说明：

1. 将源代码文件夹中所有源文件放入 sc\run 中.
2. 运行 sc\htt\ readDict.java 进行插装
3. 删除 sc\run 的错误的.java 文件，更换相应的 jdk 版本，运行 sc\htt\ Myjavac.java 进行批量编译
4. 运行 sc\htt\ Myjava.java 进行批量运行，若阻塞就将对应的文件删去，还要把此刻生成的所有结果文件全部删去.重新运行.
5. 运行 sc\htt\ CompareIn.java 显示有差异的文件，自动将差异文件存入 DifferentResultFiles 文件夹.

代码说明：

**ReadDict** 类调用 **InsertJava** 中的函数，对 **run** 目录下的所有文件进行插装。

**InsertJava** 是依靠正则表达式去匹配 **API**，在 **API** 的接下几行插入 **InsertFun** 中的插装函数。

**InsertFun** 中的插装函数通过参数 **.getClass（）.getClassLoader（）==null** 判断调用这个 **API** 的类是否是 **JAVA** 的类从而判断这个 **API** 是否是 **jdk** 的 **API**。

**Myjava** 和 **Myjavac** 是通过调用 **cmd** 命令去批处理。

**CompareIn** 自动比较 **ResultFiles** 文件夹中两个版本输入输出的区别，从而将有区别的文件自动复制到 **DifferentResultFiles** 文件夹中。