

Java Programming 과제 #6

1. 단어 검색 및 검색된 결과 파일 저장 프로그램 (SimpleGrep) (20점)

제출파일: hw06.zip

커멘트 창에서 프로그램 이름과 검색 옵션, 검색할 단어 및 디렉토리 경로를 입력하면 해당 검색어를 포함하는 파일을 찾고, 해당 파일의 라인번호와 검색 결과를 화면에 출력한다. 또한 검색 결과는 Vector<String>에 저장하고, 저장된 Vector의 내용은 검색이 완료되면 Result.txt 파일에 저장하는 프로그램을 작성하시오.

- 커멘트 창에서 실행이 되지 않으면, 0점 처리함
- 파일을 open하고 close()를 호출하지 않는 경우, 함수별 각 -2점 감점

■ Command 창에서 실행하고 main() 메소드에 명령행 인자로 전달함

- 강의자료 03장 p.47~p.48 참고

```
D:\java_workspace>java SimpleGrep [옵션] [검색어] [디렉토리명]
```

- 위의 명령 조건을 지키지 않는 경우, 에러 메시지 출력후 프로그램 종료 (2점)

```
Usage: SimpleGrep [option] [searchword] [dir]
```

■ 파일 목록 읽어오기 및 정렬 (3점)

- 실행시 지정한 디렉토리명에 존재하는 파일 및 디렉토리 목록 가져옴
- File[] File.listFiles()호출: 해당 디렉토리의 모든 파일을 File[]에 저장
- Arrays.sort(Object[] a): 파일 객체를 오름차순으로 정렬

■ 단어 검색 및 화면 출력 (12점, 각 옵션당 4점)

- option: "-c" (파일이름 및 검색된 카운트만 출력)
 - ✓ 검색 단어를 포함하고 있는 파일을 발견하면, 해당 파일에 몇개의 검색어가 포함되어 있는지 출력함 (모든 검색된 개수를 출력함: grep 명령어와 다름)
 - ✓ 검색 단어를 포함하지 않는 경우에도, 파일이름:0를 출력하고 파일 저장
- option: "-n" (파일이름, 라인번호, 검색된 소스코드 라인 출력)
 - ✓ 검색 단어를 발견하면, 검색 단어를 포함하고 있는 파일이름 및 라인 번호를 화면에 출력하고 Vector에 저장함. (String.indexOf("검색어") 사용)
 - ✓ 검색 단어를 하나도 발견하지 못한 경우 "검색어 not found!"를 화면에 출력하고 종료 (이 경우에는 Result.txt 파일로 저장하지 않음)
- option: "-i"
 - ✓ 대소문자 구분없이 입력된 검색어를 파일에서 찾음
 - ✓ 출력결과: -n 옵션과 동일(파일이름, 라인번호, 검색된 소스코드 라인 출력)
 - ✓ 검색 단어를 하나도 발견하지 못한 경우 "검색어 not found!"를 화면에 출력하고

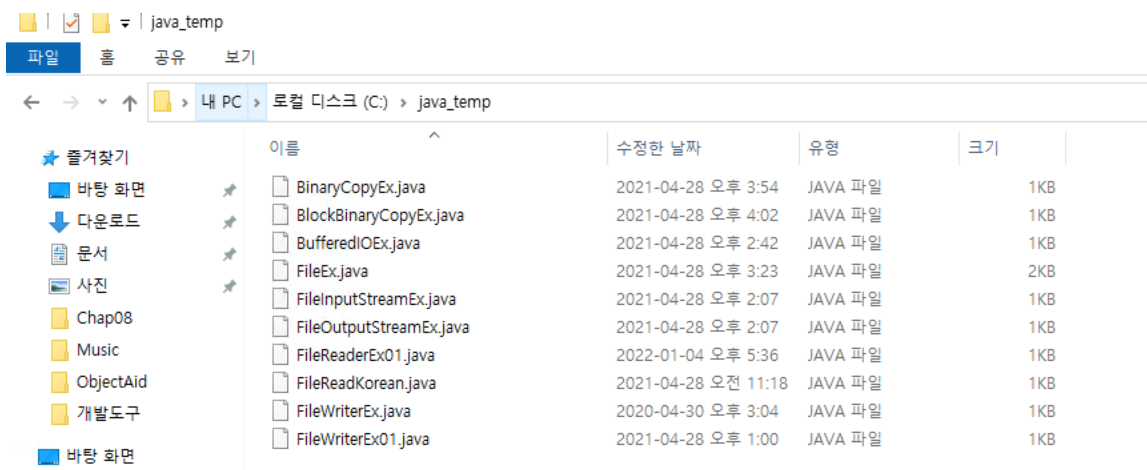
종료 (이 경우에는 Result.txt 파일로 저장하지 않음)

■ 파일 저장(Result.txt) (3점)

- 화면 출력이 끝나면, 검색된 파일이 있는 경우 검색 내용과 동일하게 Result.txt파일을 생성하고 해당 검색 내용을 Text 파일로 저장함.
- "XXX not found"가 아닌 경우에 화면 출력 결과를 파일로 저장함 (-c, -n, -i 옵션 동일)
- Result.txt 파일의 위치는 소스가 있는 폴더에 저장하고 파일은 매번 새롭게 저장해야 됨. (기존 파일을 지우고 새롭게 생성해서 파일로 저장함)
- 이클립스로 테스트 하는 경우, Project폴더 아래에 Result.txt 파일 생성됨

■ 실행 환경

- 제공된 압축 파일(**java_temp.zip**)을 c:\W\java_temp 폴더밑에 복사하고 다음 과제를 진행하면 됩니다. 아래 그림 참조



- **컴파일 및 실행 방법:** 한글 깨짐 방지 (Windows에서 사용, Mac에서는 자동으로 UTF-8로 인식)

// 컴파일

```
E:\java_workspace\Chap08\src>javac SimpeGrep.java -encoding UTF-8
```

// 실행

```
E:\java_workspace\Chap08\src>java -Dfile.encoding=UTF-8 SimpleGrep -n int c:\\java_temp
```

-D<name>=<value>: 시스템 속성 설정

- package를 사용하는 경우, 컴파일은 소스가 있는 폴더에서 컴파일하고 실행은 상위 폴더로 이동해서 "java 패키지이름.클래스이름"으로 실행해야 됨

실행 결과 #1 (오류 검사)

- 디렉토리 이름이나 검색어 및 검색 옵션을 입력하지 않은 경우
- 아래 사용법을 출력하고 프로그램을 종료함

```
E:\java_workspace\homework\src>java -Dfile.encoding=UTF-8 SimpleGrep c:\\java_temp // 검색 옵션, 검색 단어가 없는 경우
Usage: SimpleGrep [option] [searchword] [dir]
```

```
E:\java_workspace\Chap08\src>java -Dfile.encoding=UTF-8 SimpleGrep // 검색 옵션, 검색 단어 및 디렉토리 이름이 없는 경우
Usage: SimpleGrep [option] [searchword] [dir]
```

실행 결과 #2 (-c 옵션)

- c:\Wjava_temp 폴더에 있는 모든 파일들을 검색하고 화면 출력 및 파일 저장

```
E:\java_workspace\homework\src>java -Dfile.encoding=UTF-8 SimpleGrep -c Input c:\\java_temp
-----
Working dir: c:\java_temp
Search Result: option: -c, keyword: Input
-----
BinaryCopyEx.java:2
BlockBinaryCopyEx.java:2
BufferedIOEx.java:0
FileEx.java:0
FileInputStreamEx.java:3
```

```
FileOutputStreamEx.java:0
FileReaderEx01.java:0
FileReadKorean.java:5
FileWriterEx.java:0
FileWriterEx01.java:0
Result.txt saved.
```

실행 결과 #3 (-c 옵션)

- 한 라인에 Stream 단어는 여러 개 존재함 (모두 검색해야 됨)

```
E:\java_workspace\homework\src>java -Dfile.encoding=UTF-8 SimpleGrep -c Stream c:\\java_temp
-----
Working dir: c:\java_temp
Search Result: option: -c, keyword: Stream
-----
BinaryCopyEx.java:4
BlockBinaryCopyEx.java:4
BufferedIOEx.java:2
FileEx.java:0
FileInputStreamEx.java:3
FileOutputStreamEx.java:3
FileReaderEx01.java:0
FileReadKorean.java:5
FileWriterEx.java:0
FileWriterEx01.java:0
Result.txt saved.
```

실행 결과 #4 (-c 옵션):

- 검색 결과가 없는 경우, 모두 0으로 출력하고 파일 저장

```
E:\java_workspace\homework\src>java -Dfile.encoding=UTF-8 SimpleGrep -c HHH c:\\java_temp
-----
Working dir: c:\java_temp
Search Result: option: -c, keyword: HHH
```

```

-----
BinaryCopyEx.java:0
BlockBinaryCopyEx.java:0
BufferedIOEx.java:0
FileEx.java:0
FileInputStreamEx.java:0
FileOutputStreamEx.java:0
FileReaderEx01.java:0
FileReadKorean.java:0
FileWriterEx.java:0
FileWriterEx01.java:0
Result.txt saved.

```

실행 결과 #5 (-n 옵션)

- 검색 결과가 없는 경우, "검색어 not found!" 출력하고 파일 저장 없음

```

E:\java_workspace\homework\src>java -Dfile.encoding=UTF-8 SimpleGrep -n HHH c:\\java_temp
-----
Working dir: c:\java_temp
Search Result: option: -n, keyword: HHH
-----
HHH not found!

```

실행 결과 #6 (-n 옵션)

- 검색 결과가 있는 경우: Result.txt파일로 저장함

```

E:\java_workspace\homework\src>java -Dfile.encoding=UTF-8 SimpleGrep -n int c:\\java_temp
-----
Working dir: c:\java_temp
Search Result: option: -n, keyword: int
-----
BinaryCopyEx.java:9:          int c;
BinaryCopyEx.java:19:          System.out.println(src.getPath() + "를 " +
BinaryCopyEx.java:22:          System.out.println("파일 복사 오류 ");
BinaryCopyEx.java:23:          e.printStackTrace();
BlockBinaryCopyEx.java:15:          int n = fi.read(buf); // 버퍼 크기만큼 읽기. n은 실제 읽은 바이트

```

```

BlockBinaryCopyEx.java:22:                System.out.println( src.getPath() + "를 " + dest.getPath() + "로
복사하였습니다.");
BlockBinaryCopyEx.java:24:                System.out.println("파일 복사 오류");
BufferedIOEx.java:8:                int c;
BufferedIOEx.java:21:                e.printStackTrace();
FileEx.java:5:                System.out.println("-----" + dir.getPath() + "의 서브 리스트입니다. -----");
FileEx.java:7:                for(int i=0; i<subFiles.length; i++) {
FileEx.java:10:                    System.out.print(f.getName());
FileEx.java:11:                    System.out.print("\t 파일 크기: " + f.length());
FileEx.java:12:                    System.out.printf("\t 수정한 시간: %tb %td %ta %tT\n", t, t, t, t);
FileEx.java:18:                System.out.println(f1.getPath() + ", " + f1.getParent() + ", " +
FileEx.java:26:                System.out.println(f1.getPath() + "은 " + res + "입니다.");
FileInputStreamEx.java:11:                int n=0, c;
FileInputStreamEx.java:17:                System.out.println("test.out 에서 읽은 배열을 출력합니다. ");
FileInputStreamEx.java:18:                for (int i=0; i<b.length; i++)
FileInputStreamEx.java:19:                    System.out.print(b[i] + " ");
FileInputStreamEx.java:20:                System.out.println();
FileInputStreamEx.java:23:                e.printStackTrace();
FileOutputStreamEx.java:9:                for (int i=0; i<b.length; i++)
FileOutputStreamEx.java:14:                System.out.println("test.out 을 저장할 수 없습니다. "
FileOutputStreamEx.java:18:                System.out.println("test.out 을 저장하였습니다. ");
FileReaderEx01.java:11:                int c;
FileReaderEx01.java:13:                System.out.print((char)c);
FileReaderEx01.java:19:                e.printStackTrace();
FileReadKorean.java:11:                int c;
FileReadKorean.java:13:                System.out.println("Encoding charset: " + in.getEncoding());
FileReadKorean.java:16:                System.out.print((char)c);
FileReadKorean.java:21:                e.printStackTrace();
FileWriterEx.java:19:                System.out.println("File Close");
FileWriterEx.java:22:                System.out.println("입출력 오류");
FileWriterEx.java:23:                e.printStackTrace();
FileWriterEx01.java:17:                System.out.println("File close");
FileWriterEx01.java:20:                System.out.println("입출력 오류 ");
FileWriterEx01.java:21:                e.printStackTrace();
Result.txt saved.

```

실행 결과 #7 (-i 옵션)

- 대소문자 구분 없이 검색 및 화면 출력 (출력 형식은 -n 옵션과 동일)

```
E:\java_workspace\homework\src>java -Dfile.encoding=UTF-8 SimpleGrep -i out c:\\java_temp
-----
Working dir: c:\java_temp
Search Result: option: -i, keyword: out
-----
BinaryCopyEx.java:12:      FileOutputStream fo = new FileOutputStream(dest);
BinaryCopyEx.java:19:      System.out.println(src.getPath() + "를 " +
BinaryCopyEx.java:22:      System.out.println("파일 복사 오류 ");
BlockBinaryCopyEx.java:12:      FileOutputStream fo = new FileOutputStream(dest);
BlockBinaryCopyEx.java:22:      System.out.println( src.getPath() + "를 " + dest.getPath() + "로
복사하였습니다.");
BlockBinaryCopyEx.java:24:      System.out.println("파일 복사 오류");
BufferedIOEx.java:11:      BufferedOutputStream out = new BufferedOutputStream(System.out, 5);
BufferedIOEx.java:14:      out.write(c);
BufferedIOEx.java:17:      out.flush();    // 버퍼에 남아 있던 문자 모두 출력
BufferedIOEx.java:18:      out.close();
FileEx.java:5:      System.out.println("-----" + dir.getPath() + "의 서브 리스트입니다. -----");
FileEx.java:10:      System.out.print(f.getName());
FileEx.java:11:      System.out.print("\t 파일 크기: " + f.length());
FileEx.java:12:      System.out.printf("\t 수정한 시간: %tb %td %ta %tT\n", t, t, t, t);
FileEx.java:18:      System.out.println(f1.getPath() + ", " + f1.getParent() + ", " +
FileEx.java:26:      System.out.println(f1.getPath() + "은 " + res + "입니다.");
FileInputStreamEx.java:10:      FileInputStream fin = new FileInputStream("test.out");
FileInputStreamEx.java:17:      System.out.println("test.out 에서 읽은 배열을 출력합니다. ");
FileInputStreamEx.java:19:      System.out.print(b[i] + " ");
FileInputStreamEx.java:20:      System.out.println();
FileOutputStreamEx.java:3:public class FileOutputStreamEx {
FileOutputStreamEx.java:8:      FileOutputStream fout = new FileOutputStream("test.out");
FileOutputStreamEx.java:10:      fout.write(b[i]);
FileOutputStreamEx.java:12:      fout.close();
FileOutputStreamEx.java:14:      System.out.println("test.out 을 저장할 수 없습니다. "
FileOutputStreamEx.java:18:      System.out.println("test.out 을 저장하였습니다. ");
FileReaderEx01.java:13:      System.out.print((char)c);
```

```

FileReadKorean.java:13:        System.out.println("Encoding charset: " + in.getEncoding());
FileReadKorean.java:16:        System.out.print((char)c);
FileWriterEx.java:8:        FileWriter fout = null;
FileWriterEx.java:11:        fout = new FileWriter("test.txt");
FileWriterEx.java:16:        fout.write(line, 0, line.length());
FileWriterEx.java:17:        fout.write("\n", 0, 1);
FileWriterEx.java:19:        System.out.println("File Close");
FileWriterEx.java:20:        fout.close();
FileWriterEx.java:22:        System.out.println("입출력 오류");
FileWriterEx01.java:7:        FileWriter fout = null;
FileWriterEx01.java:9:        fout = new FileWriter("test1.txt");
FileWriterEx01.java:14:        fout.write(line, 0, line.length());
FileWriterEx01.java:15:        fout.write("\r\n", 0, 2);
FileWriterEx01.java:17:        System.out.println("File close");
FileWriterEx01.java:18:        fout.close();
FileWriterEx01.java:20:        System.out.println("입출력 오류 ");
Result.txt saved.

```

실행 결과 #8 (-i 옵션)

```

E:\java_workspace\homework\src>java -Dfile.encoding=UTF-8 SimpleGrep -i READ c:\\java_temp
-----
Working dir: c:\java_temp
Search Result: option: -i, keyword: READ
-----
BinaryCopyEx.java:13:        while((c=fi.read()) != -1) {
BlockBinaryCopyEx.java:15:                int n = fi.read(buf); // 버퍼 크기만큼 읽기. n은 실제 읽은 바이트
BufferedIOEx.java:7:        FileReader fin = null;
BufferedIOEx.java:10:        fin = new FileReader("test2.txt");
BufferedIOEx.java:13:        while((c = fin.read()) != -1) {
FileInputStreamEx.java:12:                while ((c = fin.read()) != -1) {
FileReaderEx01.java:3:public class FileReaderEx01 {
FileReaderEx01.java:9:        FileReader fin = new FileReader("test.txt");
FileReaderEx01.java:12:        while ((c = fin.read()) != -1) {
FileReadKorean.java:2:public class FileReadKorean {

```



```

FileReadKorean.java:5:      InputStreamReader in = null;
FileReadKorean.java:9:      in = new InputStreamReader(fin, "MS949");      // 문자 스트림 클래스
FileReadKorean.java:10:     //in = new InputStreamReader(fin, "UTF8");      // 문자 스트림 클래스
FileReadKorean.java:15:     while((c=in.read())!= -1) {
Result.txt saved.

```

Result.txt 파일 내용

- 실행 결과 내용을 파일로 저장. (기존 결과 내용은 지우고, 항상 새로운 결과만 저장함)

```

FileReadKorean.java:5:      InputStreamReader in = null;
FileReadKorean.java:9:      in = new InputStreamReader(fin, "MS949");      // 문자 스트림 클래스
FileReadKorean.java:10:     //in = new InputStreamReader(fin, "UTF8");      // 문자 스트림 클래스
FileReadKorean.java:15:     while((c=in.read())!= -1) {
Result.txt saved.

```

참고 자료

■ Windows 10의 find 명령어 사용 방법

- > **find /N "검색단어" 파일이름** -> 검색단어가 있는 문장을 찾고 해당 라인번호 출력
- > **find "검색단어" 파일이름** -> 검색단어가 있는 문장만 출력

```
C:\Temp>find /N "new" BlockBinaryCopyEx.java

----- BLOCKBINARYCOPYEX.JAVA
[5]          File src = new File("/Users/changsu/temp/img1.jpg");
[6]          File dest = new File("/Users/changsu/temp/desert.jpg");
[8]          FileInputStream fi = new FileInputStream(src);
[9]          FileOutputStream fo = new FileOutputStream(dest);
[10]         byte [] buf = new byte [1024*10]; // 10KB

C:\Temp>find "new" BlockBinaryCopyEx.java

----- BLOCKBINARYCOPYEX.JAVA
          File src = new File("/Users/changsu/temp/img1.jpg");
          File dest = new File("/Users/changsu/temp/desert.jpg");
              FileInputStream fi = new FileInputStream(src);
              FileOutputStream fo = new FileOutputStream(dest);
              byte [] buf = new byte [1024*10]; // 10KB
```

■ grep 실행 화면 (Linux, Mac)

- 참고 사이트: <https://recipes4dev.tistory.com/157>

\$ grep -n "검색단어" 파일이름 또는 * (모든 파일): 검색된 파일만 화면에 출력

- 파일이름:라인번호:검색된 문장

```

java_temp — changsu@Jungui-iMac — ..emp/java_temp — -zsh — 115x26
[changsu@ ~/temp/java_temp$ grep -n Input *
BinaryCopyEx.java:11:         FileInputStream fi = new FileInputStream(src);
BlockBinaryCopyEx.java:11:         FileInputStream fi = new FileInputStream(src);
FileInputStreamEx.java:3:public class FileInputStreamEx {
FileInputStreamEx.java:10:         FileInputStream fin = new FileInputStream("test.out");
FileReadKorean.java:5:         InputStreamReader in = null;
FileReadKorean.java:6:         FileInputStream fin = null;
FileReadKorean.java:8:         fin = new FileInputStream("/Users/changsu/temp/hangul1.txt");
FileReadKorean.java:9:         in = new InputStreamReader(fin, "MS949");           // 문자 스트림 클래스
FileReadKorean.java:10:        //in = new InputStreamReader(fin, "UTF8");           // 문자 스트림 클래스
changsu@ ~/temp/java_temp$

```

\$ grep -c "검색단어" 파일이름 또는 * (모든 파일)

- 검색단어를 찾지 못한 경우 (파일이름:0 출력)
- 출력된 검색된 숫자는 실제 검색된 개수가 아니라 라인 단위 숫자임

```

java_temp — changsu@Jungui-iMac — ..emp/java_temp — -zsh — 115x26
[changsu@ ~/temp/java_temp$ grep -c Input *
BinaryCopyEx.java:1
BlockBinaryCopyEx.java:1
BufferedIOEx.java:0
FileEx.java:0
FileInputStreamEx.java:2
FileOutputStreamEx.java:0
FileReadKorean.java:5
FileReaderEx01.java:0
FileWriterEx.java:0
FileWriterEx01.java:0
changsu@ ~/temp/java_temp$

```

\$ grep -i out *

- 대소문자 구분없이 out(Out)를 검색: ignore case

```
java_temp — changsu@Jungui-iMac — .emp/java_temp — -zsh — 135x46

[changsu@ ~/temp/java_temp$ grep -i out *
BinaryCopyEx.java:      FileOutputStream fo = new FileOutputStream(dest);
BinaryCopyEx.java:      System.out.println(src.getPath() + "를 " +
BinaryCopyEx.java:      System.out.println("파일 복사 오류 ");
BlockBinaryCopyEx.java:      FileOutputStream fo = new FileOutputStream(dest);
BlockBinaryCopyEx.java:      System.out.println( src.getPath() + "를 " + dest.getPath() + "로 복사하였습니다.");
BlockBinaryCopyEx.java:      System.out.println("파일 복사 오류");
BufferedIOEx.java:      BufferedOutputStream out = new BufferedOutputStream(System.out, 5);
BufferedIOEx.java:      out.write(c);
BufferedIOEx.java:      out.flush();    // 버퍼에 남아 있던 문자 모두 출력
BufferedIOEx.java:      out.close();
FileEx.java:      System.out.println("-----" + dir.getPath() + "의 서브 리스트입니다. -----");
FileEx.java:      System.out.print(f.getName());
FileEx.java:      System.out.print("\t파일 크기: " + f.length());
FileEx.java:      System.out.printf("\t수정 한 시간 : %tb %td %ta %tT\n", t, t, t, t);
FileEx.java:      System.out.println(f1.getPath() + ", " + f1.getParent() + ", " +
FileEx.java:      System.out.println(f1.getPath() + "은 " + res + "입니다.");
FileInputStreamEx.java:      FileInputStream fin = new FileInputStream("test.out");
FileInputStreamEx.java:      System.out.println("test.out에서 읽은 배열을 출력합니다. ");
FileInputStreamEx.java:      System.out.print(b[i] + " ");
FileInputStreamEx.java:      System.out.println();
FileOutputStreamEx.java:public class FileOutputStreamEx {
FileOutputStreamEx.java:      FileOutputStream fout = new FileOutputStream("test.out");
FileOutputStreamEx.java:      fout.write(b[i]);
FileOutputStreamEx.java:      fout.close();
FileOutputStreamEx.java:      System.out.println("test.out을 저장할 수 없습니다. "
FileOutputStreamEx.java:      System.out.println("test.out을 저장 하였습니다. ");
FileReadKorean.java:      System.out.println("Encoding charset: " + in.getEncoding());
FileReadKorean.java:      System.out.print((char)c);
FileReaderEx01.java:      System.out.print((char)c);
FileWriterEx.java:      FileWriter fout = null;
FileWriterEx.java:      fout = new FileWriter("test.txt");
FileWriterEx.java:      fout.write(line, 0, line.length());
FileWriterEx.java:      fout.write("\n", 0, 1);
FileWriterEx.java:      System.out.println("File Close");
FileWriterEx.java:      fout.close();
FileWriterEx.java:      System.out.println("입출력 오류 ");
FileWriterEx01.java:      FileWriter fout = null;
FileWriterEx01.java:      fout = new FileWriter("test1.txt");
FileWriterEx01.java:      fout.write(line, 0, line.length());
FileWriterEx01.java:      fout.write("\r\n", 0, 2);
FileWriterEx01.java:      System.out.println("File close");
FileWriterEx01.java:      fout.close();
FileWriterEx01.java:      System.out.println("입출력 오류 ");
changsu@ ~/temp/java_temp$
```