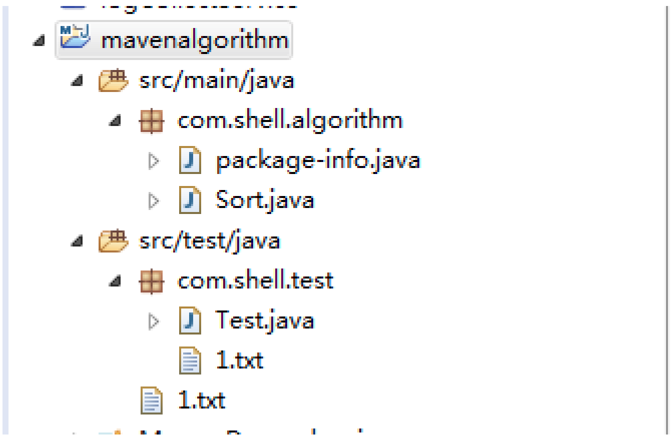
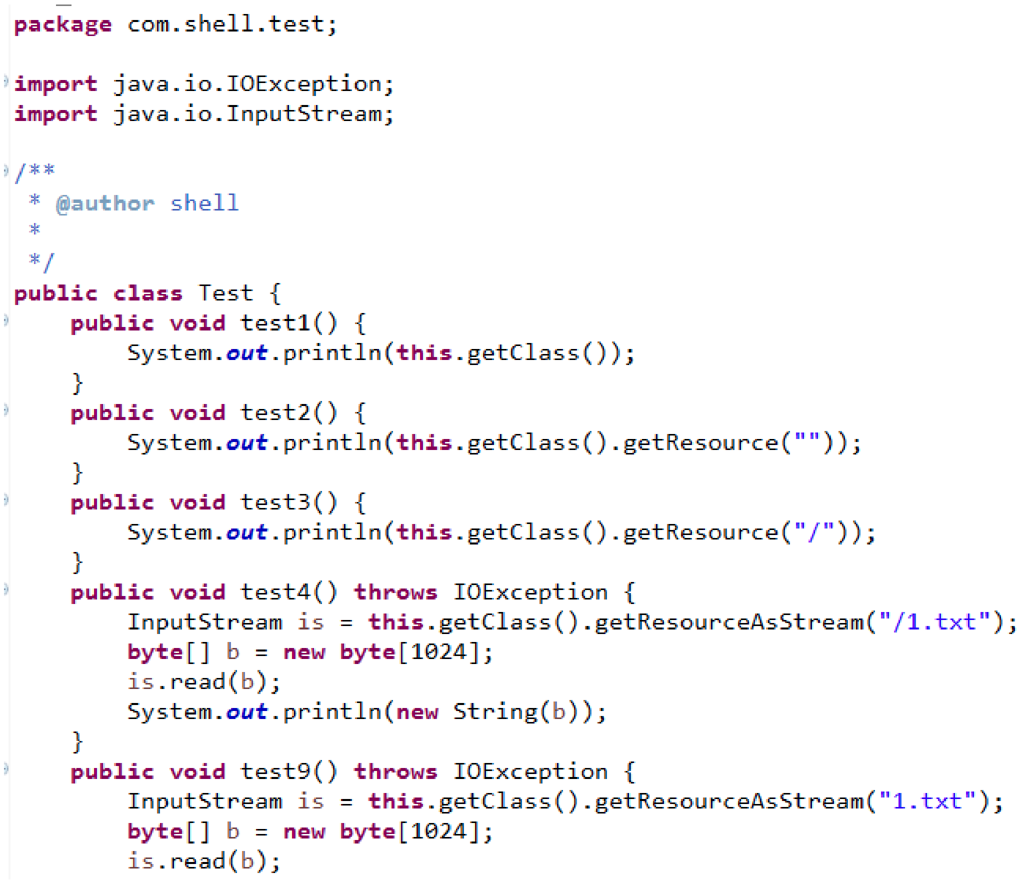
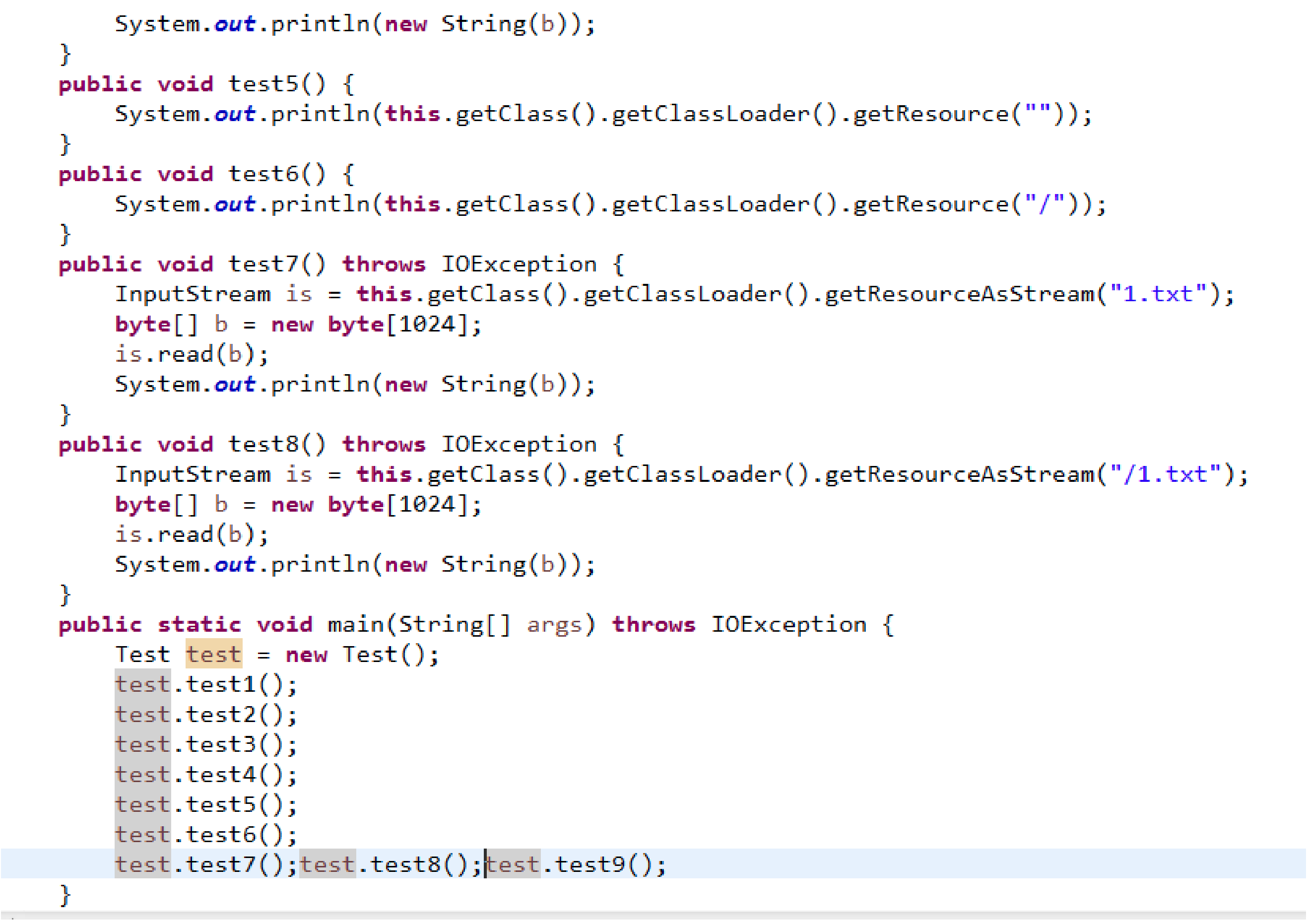
getClass().getResource()等获取文件的方法总结

首先要明白的一点是：所有方法的路径都是针对的类路径，也就是说是编译后的.class文件所在路径来说的。项目结构如下：



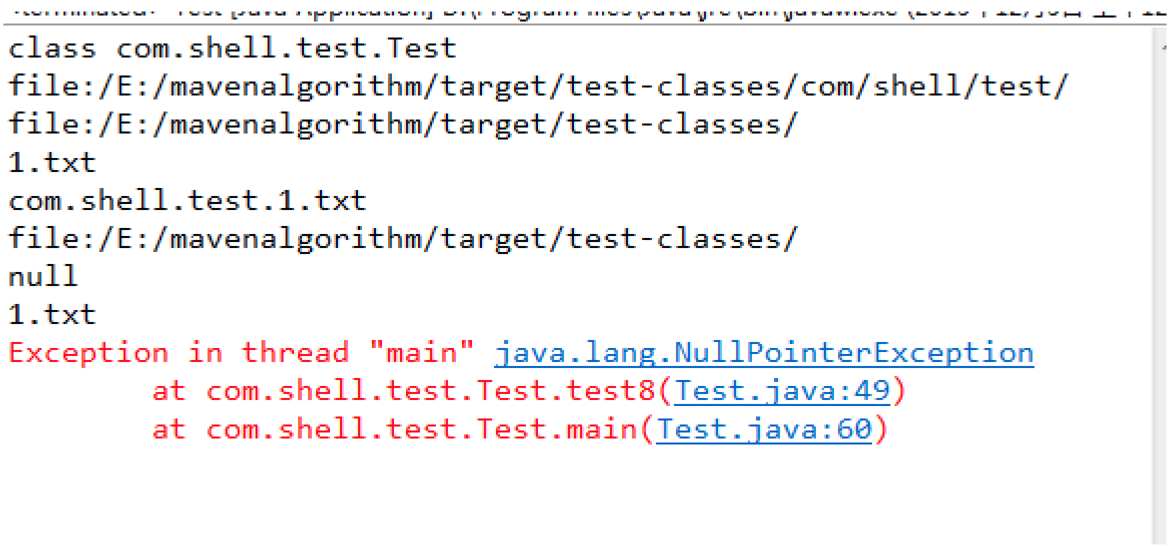


```java
package com.shell.test;

import java.io.IOException;
import java.io.InputStream;

/**
 * @author shell
 *
 */
public class Test {
    public void test1() {
        System.out.println(this.getClass());
    }
    public void test2() {
        System.out.println(this.getClass().getResource(""));
    }
    public void test3() {
        System.out.println(this.getClass().getResource("/"));
    }
    public void test4() throws IOException {
        InputStream is = this.getClass().getResourceAsStream("/1.txt");
        byte[] b = new byte[1024];
        is.read(b);
        System.out.println(new String(b));
    }
    public void test9() throws IOException {
        InputStream is = this.getClass().getResourceAsStream("1.txt");
        byte[] b = new byte[1024];
        is.read(b);
```



```java
        System.out.println(new String(b));
    }
    public void test5() {
        System.out.println(this.getClass().getClassLoader().getResource(""));
    }
    public void test6() {
        System.out.println(this.getClass().getClassLoader().getResource("/"));
    }
    public void test7() throws IOException {
        InputStream is = this.getClass().getClassLoader().getResourceAsStream("1.txt");
        byte[] b = new byte[1024];
        is.read(b);
        System.out.println(new String(b));
    }
    public void test8() throws IOException {
        InputStream is = this.getClass().getClassLoader().getResourceAsStream("/1.txt");
        byte[] b = new byte[1024];
        is.read(b);
        System.out.println(new String(b));
    }
    public static void main(String[] args) throws IOException {
        Test test = new Test();
        test.test1();
        test.test2();
        test.test3();
        test.test4();
        test.test5();
        test.test6();
        test.test7();test.test8();test.test9();
    }
```

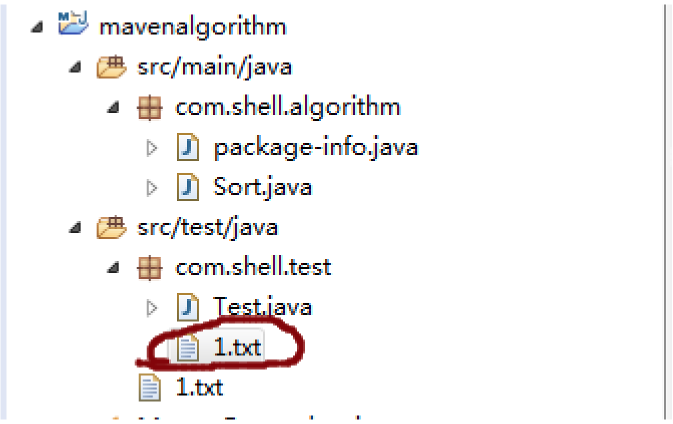
结果：



```
class com.shell.test.Test
file:/E:/mavenalgorithm/target/test-classes/com/shell/test/
file:/E:/mavenalgorithm/target/test-classes/
1.txt
com.shell.test.1.txt
file:/E:/mavenalgorithm/target/test-classes/
null
1.txt
Exception in thread "main" java.lang.NullPointerException
	at com.shell.test.Test.test8(Test.java:49)
	at com.shell.test.Test.main(Test.java:60)
```

1. this.getClass().getResource("")：返回的是包含类全名的路径，一直是com.shell.test包的路径，也就是：E:/mavenalgorithm/target/test-classes/com/shell/test/
2. this.getClass().getResource("/")：返回的是项目配置的classpath路径，也就是：E:/mavenalgorithm/target/test-classes

3. this.getClass().getResourceAsStream("1.txt")：获取的是当前类所在目录下的文件1.txt, 也就是：E:/mavenalgorithm/target/test-classes/com/shell/test/1.txt



```
◢ mavenalgorithm
    ◢ src/main/java
        ◢ com.shell.algorithm
            ▷ package-info.java
            ▷ Sort.java
    ◢ src/test/java
        ◢ com.shell.test
            ▷ Test.java
              1.txt
          1.txt
```

4. this.getClass().getResourceAsStream("/1.txt")：获取的是项目存放class文件的根目录下的1.txt, 也就是：E:/mavenalgorithm/target/test-classes/1.txt

5. this.getClass.getClassLoader().getResource()跟this.getClass().getResource("/")的用处一；同理：this.getClass().getClassLoader().getResourceAsStream("")跟this.getClass().getResourceAsStream("/")是一样的

6. this.getClass.getClassLoader().getResource("/")会报错

   同理：this.getClass().getClassLoader().getResourceAsStream("/")会报错

   也就是说this.getClass().getClassLoader()执行之后已经是定位到了项目配置的classpath路径了

   this.getClass().getResource("/")和this.getClass().getResourceAsStream("/")一样的是定位到了项目的classpath下