1. 以后数据库的用户名和密码一般都会保存到properties（配置文件）
2. 类文件路径:就是使用了classpath的路径来找资源文件
3. eclipse编译的时候是以项目源代码文件夹为当前位置的，注意相对位置

# 文件路径

### 绝对路径

以根目录或某盘符开头的路径（或者说完整的路径）

例如：

* c:/a.txt （Windows操作系统中）
* c:/xxx/a.txt （Windows操作系统中）
* /var/xx/aa.txt （Linux操作系统中）

绝对路径的问题: 比如C:\abc\a.properties文件路径，该路径在windows上执行没有 问题，但是如果把该项目移动到linux上面执行 ，该路径就会出现问题了，因为在linux上面没有c盘的，只有根目录\。

### 相对路径

相对于当前路径的一个路径。例如当前文件夹为c:/abc时：相对路径a.txt表示c:/abc/a.txt，相对路径xx/a.txt = c:/abc/xx/a.txt

* . 表示当前文件夹
* .. 表示上级文件夹

相对路径存在的问题:相对路径是相对于目前执行class文件的时候，控制台所在的路径，这样子也会导致出现问题。

### Java程序中的相对路径

**在Java程序中使用File时写相对路径，是指相对于执行java命令时当前所在的文件夹。**

**测试代码：**

**public** **class** PathTest {

**public** **static** **void** main(String[] args) **throws** Exception {

System.*out*.println(**new** File("a.txt").getAbsolutePath());

}

}

在命令行中使用cd命令切换到不同的路径下试试，可以看到以上所说的效果。

**在Eclipse中，当前路径是工程的根目录。**

### classpath路径

#### classpath路径说明

在Java程序中，一般情况下使用绝对路径还是相对路径都不太合适，因为Java程序的jar包所放的位置不确定，执行java程序时当前的路径也不确定，所以不合适。一般在Java程序中我们会把资源放到classpath中，然后使用classpath路径查找资源。

Classpath路径：就是使用classpath目前的路径。

#### 获取classpath中的资源（InputStream）

**public** **static** **void** main(String[] args) **throws** Exception {

Class clazz = **new** ClassPathTest().getClass();

// 开头的'/'表示classpath的根目录，这个是表示从classpath的根目录中开始查找资源

InputStream in = clazz.getResourceAsStream("/cn/itcast/my.properties");

// 如果开头没有'/'，表示从当前这个class所在的包中开始查找

InputStream in2 = clazz.getResourceAsStream("my.properties");

}