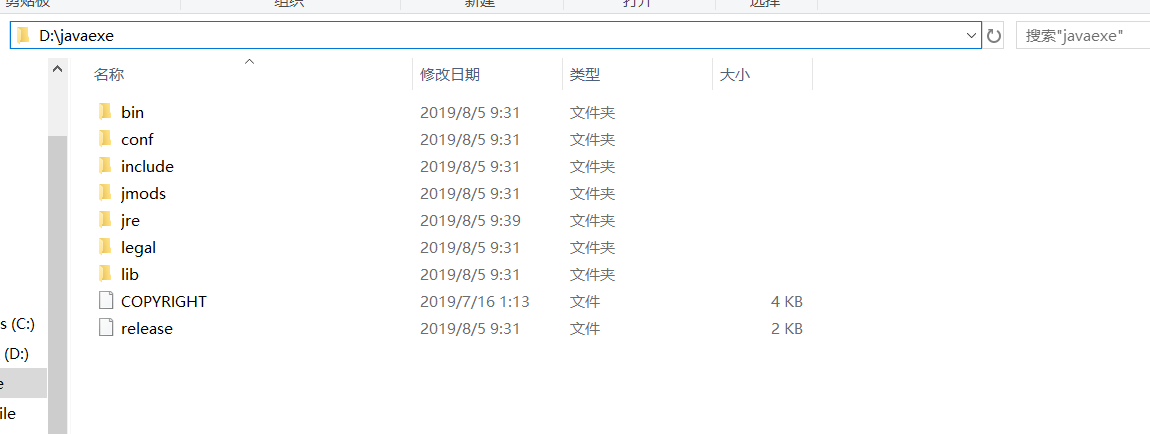
# jdk与jre安装

下载路径<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

新的版本没有jre，

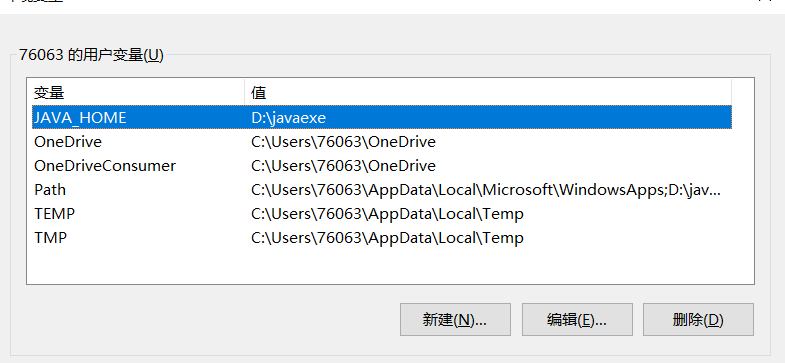
打开有bin文件的目录



按住shift键，鼠标右键打开powershell窗口输入bin\jlink.exe --module-path jmods --add-modules java.desktop --output jre生成jre文件。

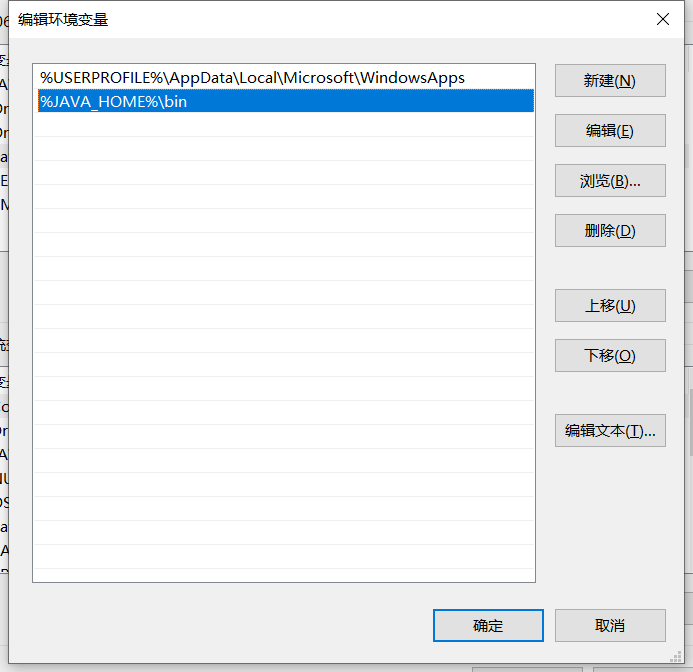
## 配置环境变量

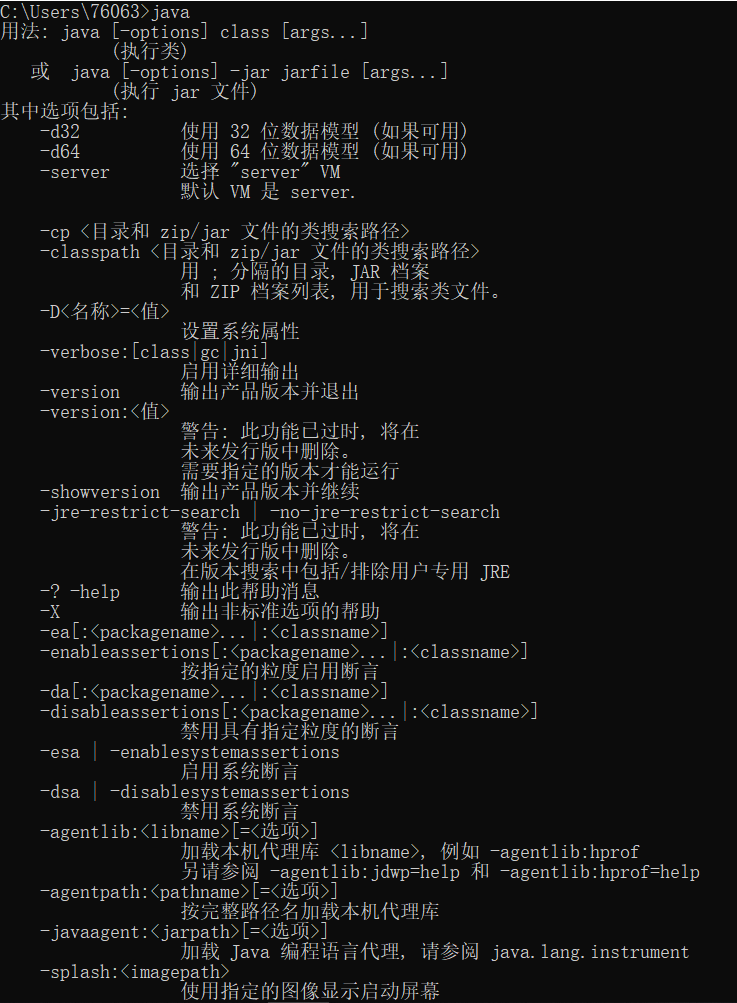
桌面右键我的计算机/属性/高级设置/环境变量



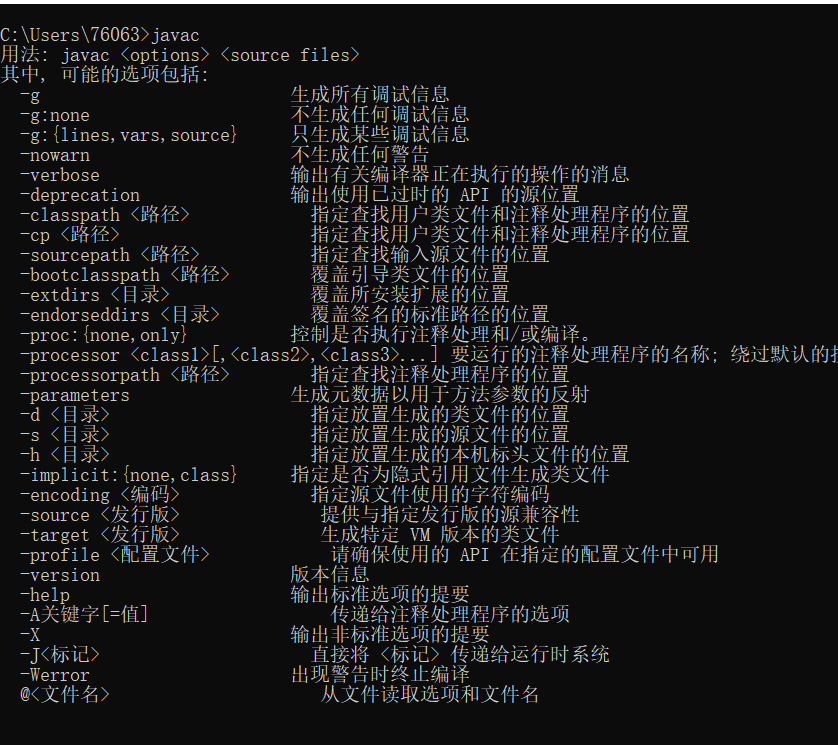
增加变量名JAVA\_HOME，变量值就是你的安装路径。

然后双击Path，增加

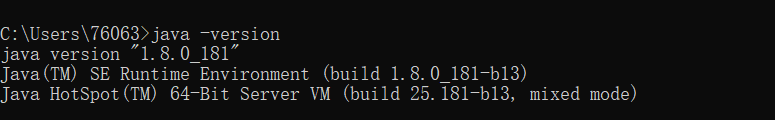


win+

javac



Java -version（安装版本）



echo %JAVA\_HOME%（javahome路径）



# eclipse安装

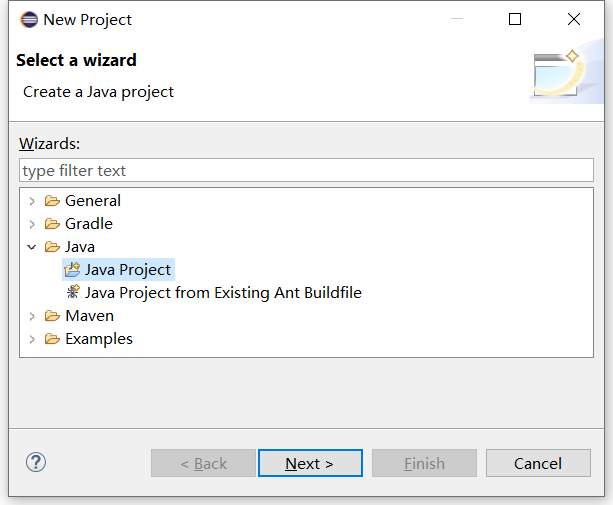
下载地址eclipse.org



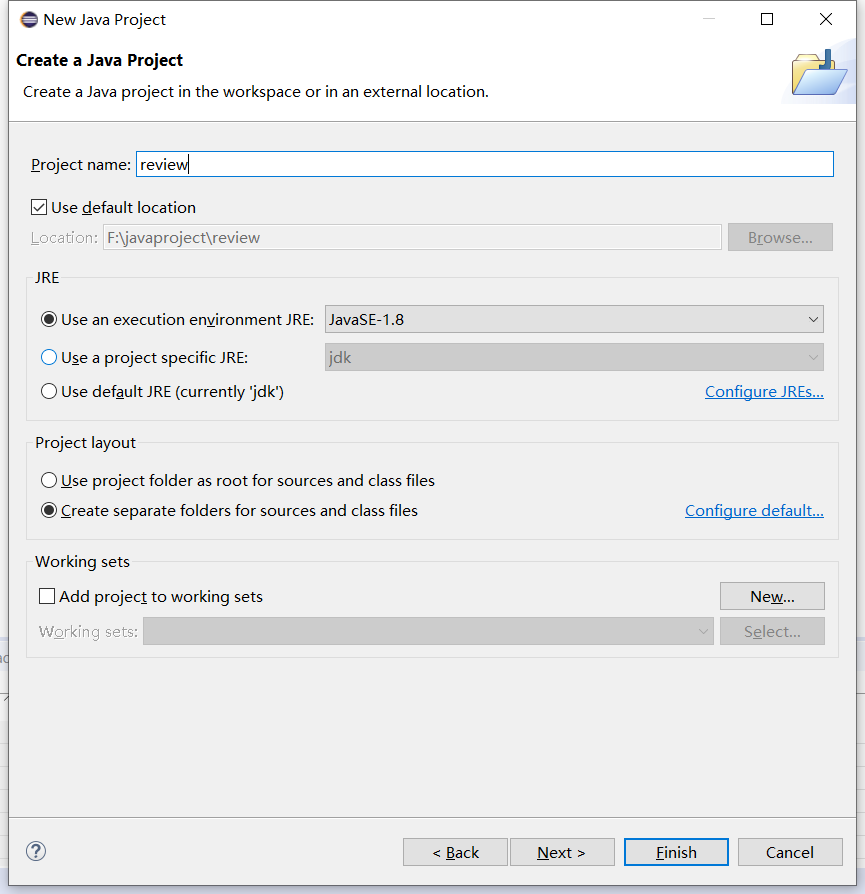
# 新建一个java项目

ctrl+n

# 命名



项目命名：全部小写



新建包

在SRC（SRC是专门放java源代码的文件夹，就是你在IDE里编写的各个java类的文件都在里面）中新建package包

包的命名规范：包名全部使用小写。包名通常由若干个标识符组成，标识符之间用点（.）隔开，其第一个标识符往往表示域名。

在这个包中新建一个类

类的命名规范：首字母大写

java由关键字和标识符。

# 数据类型

基本数据类型：整型，浮点型，字符型，布尔型

引用数据类型：数组、类 （接口？？）

基本数据类型

|  |  |  |  |
| --- | --- | --- | --- |
| **四类** | **八种** | **字节数** | **数据表示范围** |
| **整型** | **byte** | 1 | -128～127 |
| **short** | 2 | -32768～32767 |
| **int** | 4 | -2147483648～2147483648 |
| **long** | 8 | -263～263-1 |
| **浮点型** | **float** | 4 | -3.403E38～3.403E38 |
| **double** | 8 | -1.798E308～1.798E308 |
| **字符型** | **char** | 2 | 表示一个字符，如('a'，'A'，'0'，'家') |
| **布尔型** | **boolean** | 1 | 只有两个值true与false |

# scanner键盘录入

导包：import java.util.Scanner;

数据类型 变量名 = new 数据类型();

每种引用数据类型都有其功能，我们可以调用该类型实例的功能。

变量名.方法名();

sc.nextInt(); 用来接收控制台录入的数字

sc.next(); 用来接收控制台录入的字符串

Scanner 变量名 = new Scanner( )

举例

Scanner sc = **new** Scanner(System.*in*);

**int** n = sc.nextInt();

# 随机数random

导包：import java.util.Random;

Random 变量名 = new Random;

public int nextInt(int maxValue) 产生[0,maxValue)范围的随机整数，包含0，不包含maxValue；

public double nextDouble() 产生[0,1)范围的随机小数，包含0.0，不包含1.0。

举例

Random r = new Random();

int i = r.nextInt(100);

**double** d = r.nextDouble();

# 数组

数组定义

数据类型[ ] 数组名 = new 数据类型[①元素个数或②数组长度];

数组越界异常//划重点

ArrayIndexOutOfBoundsException

空指针异常

NullPointerException。

# 方法

## 方法的定义

无参无返回值

public void 方法名(){

}

无参有返回值；

public 数据类型 方法名 (){  
 return 1;

}

有参有返回值：

public 数据类型 方法名（数据类型 参数名，数据类型 参数名，数据类型 参数名）{

return 1；

}

有参无返回值：

public void 方法名( 数据类型 参数名){

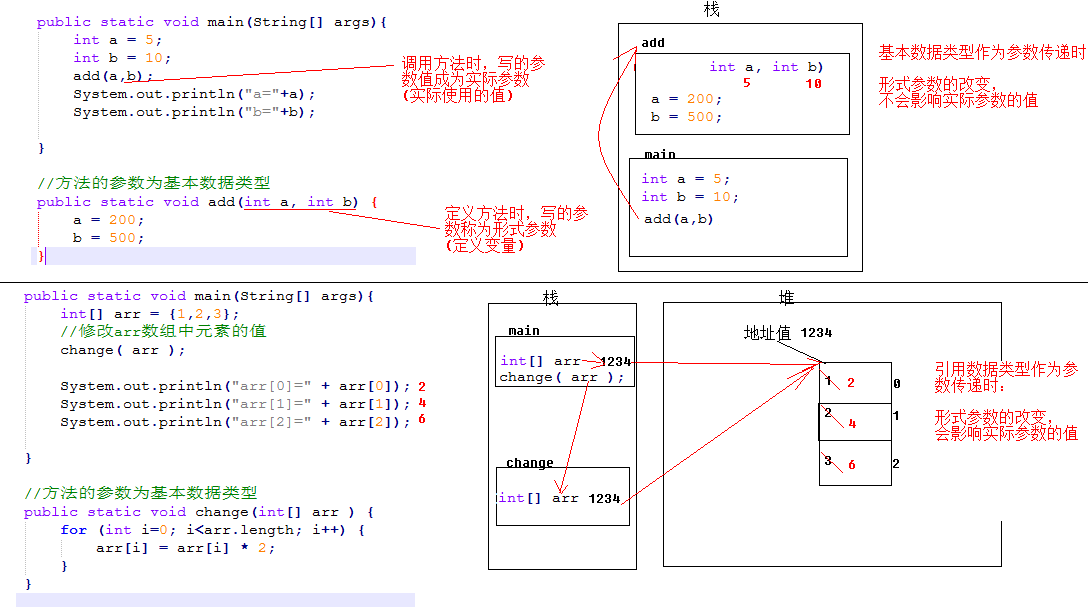
}

方法中不能再定义方法，但是可以调用方法

## 方法的重载

多个方法的方法名相同，但是参数列表不同。参数个数不同，参数类型不同，参数顺序不同

## 参数传递



* 当调用方法时，如果传入的数值为基本数据类型（包含String类型），形式参数的改变对实际参数不影响
* 当调用方法时，如果传入的数值为引用数据类型（String类型除外），形式参数的改变对实际参数有影响

# 类

创建java文件，与类名相同

public class 类名{

数据类型 属性名称1；

数据类型 属性名称2；

…

}