java基础知识 第2天

**【学习目标】理解、了解、应用、记忆**

通过今天的学习，参训学员能够：

1. **【应用】完成Eclipse工具的下载、安装、配置并能使用eclipse编写代码**
2. 【应用】独立下载并安装eclipse
3. 【记忆】独立配置eclipse的工作空间
4. 【应用】应用eclipse的辅助键和快捷键编写代码
5. 【应用】能够完成导入项目操作和删除项目操作
6. **【理解】能够灵活的使用java中的各种运算符**
7. 【应用】独立编写代码使用(+,-,\*,/,%,++,--等)算数运算符完成数学运算
8. 【应用】阐述取余和除法的区别
9. 【理解】阐述字符和字符串参与加法运算操作需要注意的事项
10. 【理解】阐述++和--的作用，以及++，--在前或在后的区别
11. 【理解】阐述赋值运算符和扩展运算符有哪些
12. 【理解】阐述关系运算符有哪些，关系运算符得到的结果是什么
13. 【理解】阐述逻辑运算符的基本使用及&与&&的区别
14. 【应用】阐述三元运算符使用格式及使用场景
15. **【应用】使用Scanner完成键盘录入功能**
16. 【应用】阐述键盘录入的基本步骤
17. 【应用】独立编写代码，使用键盘录入的方式完成：求两个数的和、判断两个数是否相等、获取三个数中的最大值

# Eclipse开发工具

## Eclipse概述和安装

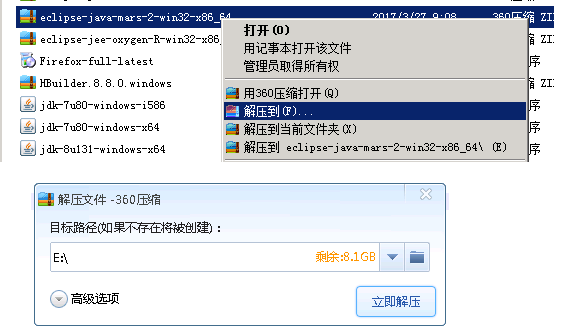
Eclipse是一个IDE(集成开发环境)IDE(Integrated Development Environment)

集成了代码编写功能，分析功能，编译功能，调试功能等一体化的开发软件。

Eclipse的特点描述：免费、纯Java语言编写、免安装、扩展性强

下载和安装、下载 http://eclipse.org/

安装：绿色版、解压就可以使用(Eclipse)

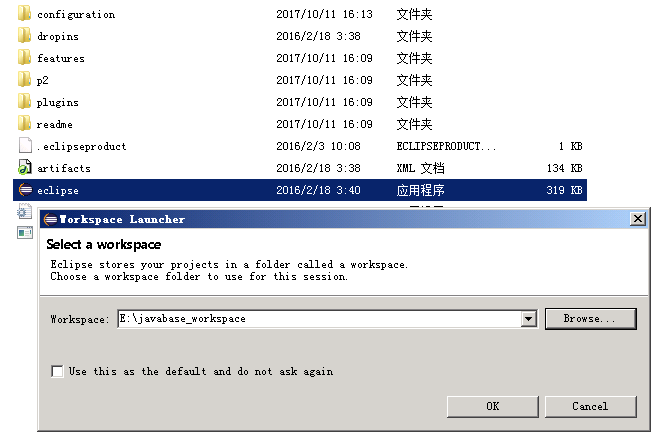


## Eclipse的基本使用

### Eclipse基本操作

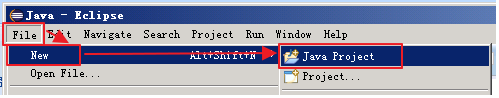
选择工作空间

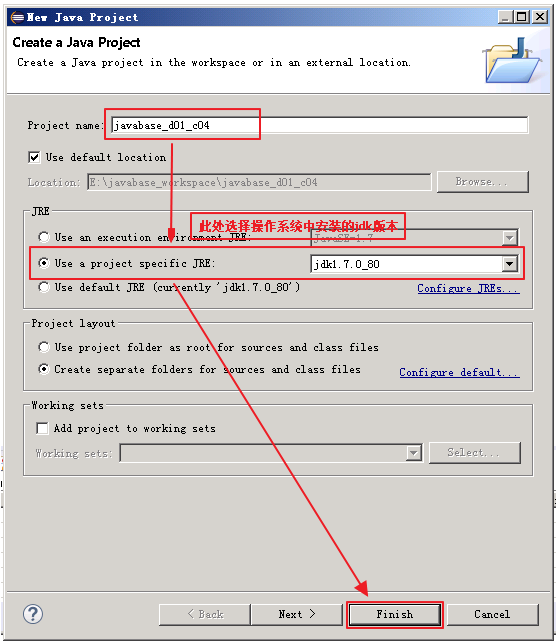
工作空间 其实就是我们写的源代码所在的目录



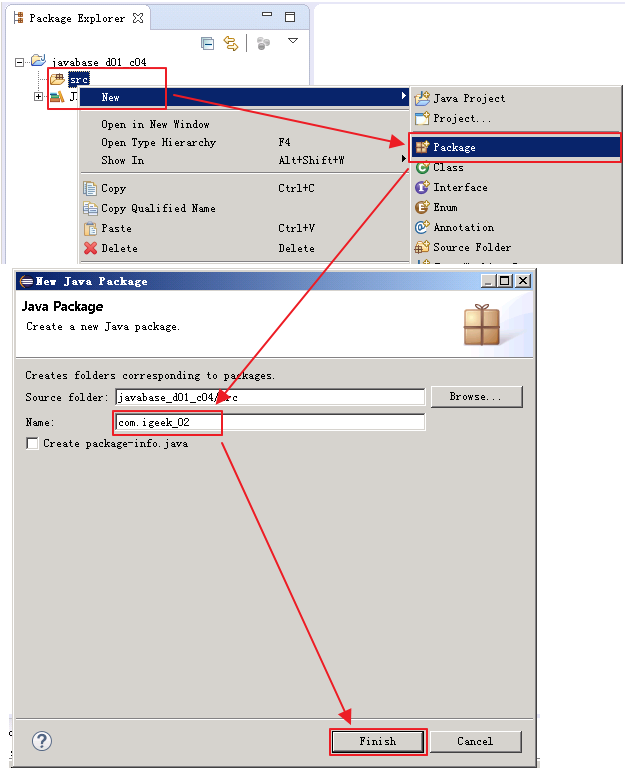
用Eclipse来完成一个HelloWorld案例

A:创建Java项目：点击File或者在最左侧空白处，选择Java项目，在界面中写一个项目名称javabase\_d01\_c04，然后Finish即可。



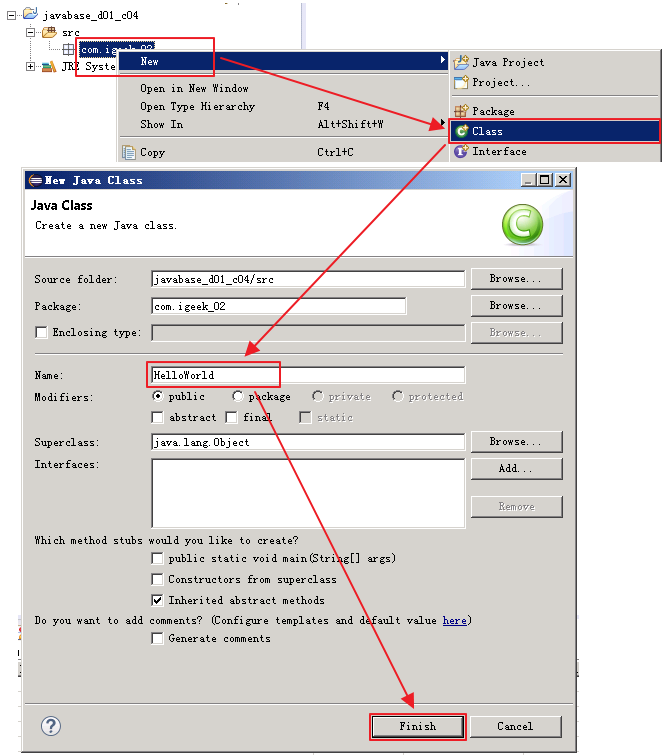


B:创建包：展开项目，在源包src下建立一个包com.igeek\_02



C:创建类：在com. igeek\_02包下建立一个类HelloWorld

在界面中写一个类名：HelloWorld，然后finish即可。



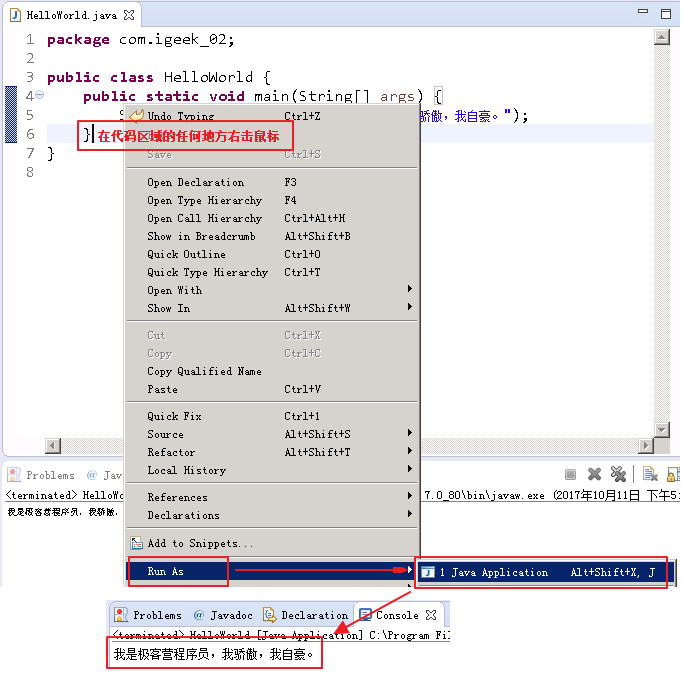
D:编写代码：在HelloWorld类写main方法，在main方法中写

一条输出语句：我是极客营程序员，我骄傲，我自豪。

E:编译：自动编译，在保存的那一刻帮你做好了

F:运行 选择要运行的文件或者在要运行的文件内容中

右键 -- Run as - Java Application即可，在控制台可以看到运行结果



### 代码案例十

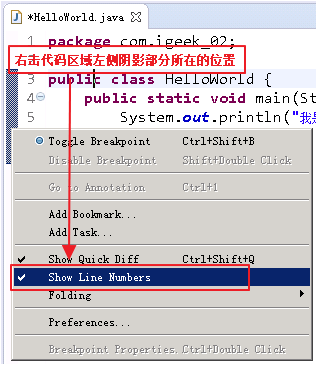
|  |
| --- |
| **package** com.igeek\_02;  /\*\*  \* **@ClassName**: HelloWorld  \* **@Description**: 在控制台打印相关信息  \* **@date** 2017年10月11日 下午6:49:38  \* Company www.igeekhome.com  \*  \*/  **public** **class** HelloWorld {  **public** **static** **void** main(String[] args) {  System.***out***.println("我是极客营程序员，我骄傲，我自豪。");  }  } |

## Eclipse工作空间的基本配置

A:行号的显示和隐藏

显示：在代码区域的最左边的空白区域，右键 -- Show Line Numbers即可。

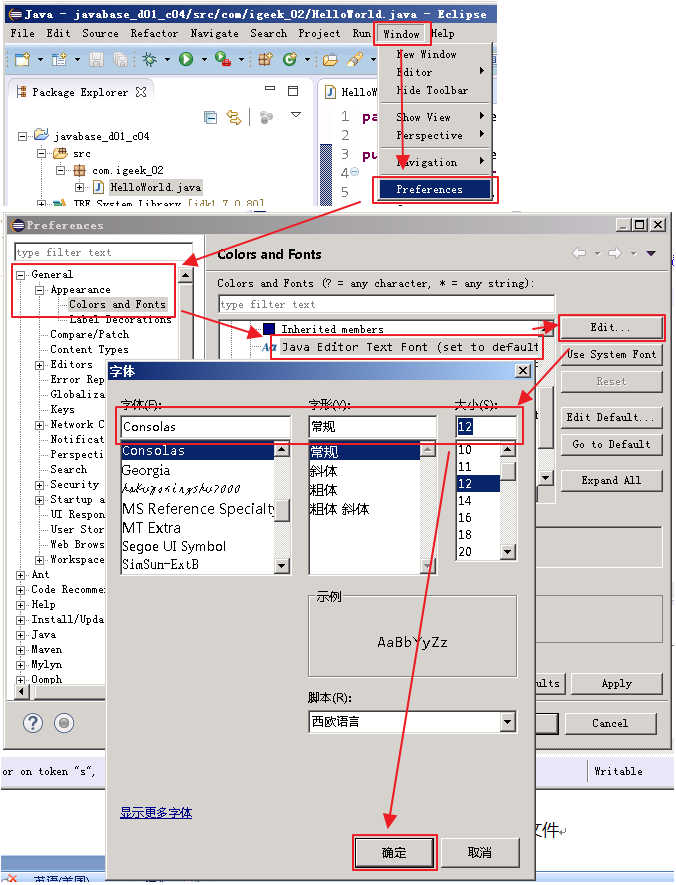
隐藏：把上面的动作再做一次。



B:字体大小及颜色

a:Java代码区域的字体大小和颜色：

window -- Preferences -- General -- Appearance -- Colors And Fonts -- Java -- Java Edit Text Font



b:控制台（类似a步骤过程）

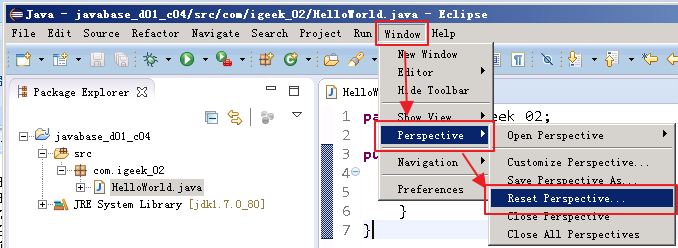
window -- Preferences -- General -- Appearance -- Colors And Fonts -- Debug -- Console font

c:其他文件（类似a步骤过程）

window -- Preferences -- General -- Appearance -- Colors And Fonts -- Basic -- Text Font

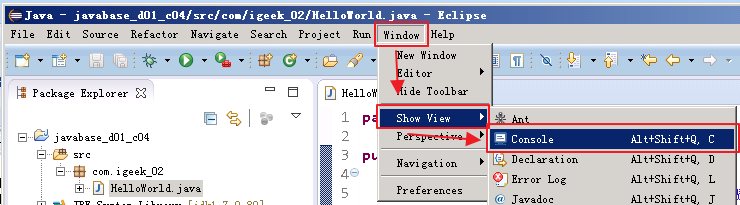
C:窗体给弄乱了，怎么办?

window -- Perspective -- Reset Perspective



D:控制台找不到了，怎么办?

Window--Show View—Console



## Eclipse中辅助键和快捷键的使用

### 常用辅助键和快捷键概述

内容辅助键 alt+/：

main 然后alt+/

sysout 然后alt+/

快捷键：

注释：

单行 选中内容，ctrl+/, 再来一次取消

多行 选中内容，ctrl+shift+/, ctrl+shift+\

格式化 ctrl+shift+f

### 代码案例十一

|  |
| --- |
| **package** com.igeek\_04;  /\*\*  \* **@ClassName**: HelloWorld  \* **@Description**: 演示快捷键的使用  \* **@date** 2017年10月11日 下午6:49:38  \* Company www.igeekhome.com  \*  \* 内容辅助键：alt+/  \* A:main方法 main,然后alt+/,回车  \* B:输出语句 sysout,然后然后alt+/,回车  \*  \* 快捷键：  \* A:注释  \* 单行 选中内容，ctrl+/,再来一次就是取消  \* 多行 选择内容，ctrl+shift+/,ctrl+shift+\  \* B:格式化  \* ctrl+shift+f  \*/  **public** **class** HelloWorld {  **public** **static** **void** main(String[] args) {  System.***out***.println("HelloWorld1");  System.***out***.println("HelloWorld2");  System.***out***.println("HelloWorld3");  System.***out***.println("HelloWorld4");  System.***out***.println("HelloWorld5");  }  } |

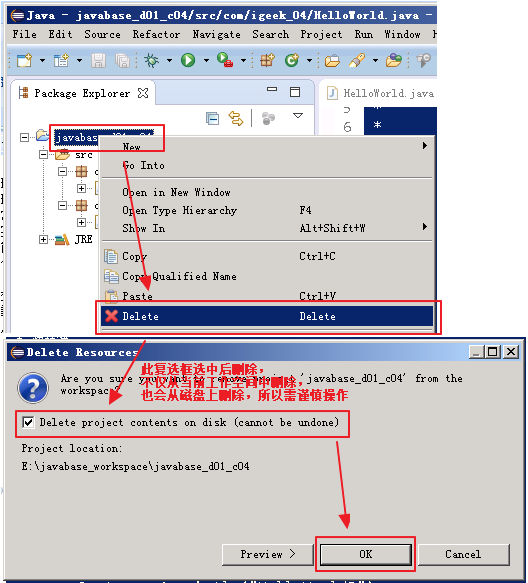
## Eclipse中项目的删除和导入

A:删除项目

选中项目 – 右键 – 删除

从项目区域中删除

从硬盘上删除



B:导入项目

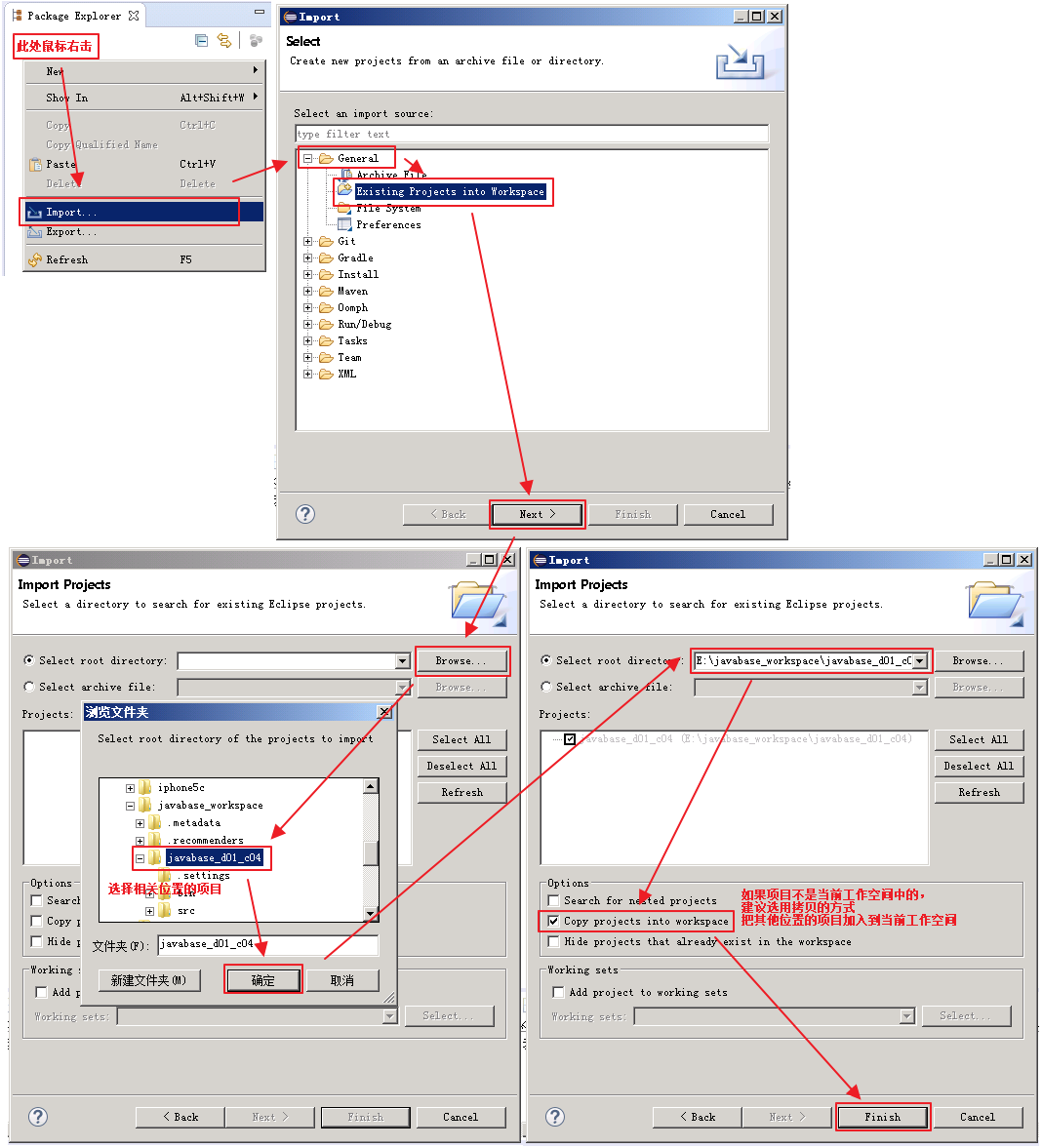
在项目区域右键找到import

找到General，展开，并找到

Existing Projects into Workspace

点击next,然后选择你要导入的项目

注意：这里选择的是项目名称



# 运算符

## 算数运算符

* 运算符

对常量和变量进行操作的符号称为运算符

* 表达式

用运算符把常量或者变量连接起来符号java语法的式子就可以称为表达式不同运算符连接的式子体现的是不同类型的表达式

定义两个int类型的变量a,b，做加法(a + b)

* 常用运算符

算术运算符

赋值运算符

关系运算符

逻辑运算符

三元运算符

### 算数运算符的基本用法

### 算数运算符概述

A:什么是运算符

就是对常量和变量进行操作的符号

B:算数运算符有哪些

\* +,-,\*,/,%,++,--

### 代码案例十二

|  |
| --- |
| **package** com.igeek\_01;  /\*\*  \* **@ClassName**: HelloWorld  \* **@Description**: 算术运算符的使用  \* **@date** 2017年10月11日 下午6:49:38  \* Company www.igeekhome.com  \*  \* 运算符：对常量和变量进行操作的符号  \* 表达式：用运算符连接起来的符合java语法的式子。不同类型的运算符连接起来的式子是不同的表达式。  \* 举例：定义两个int类型的变量a,b,  \* a + b  \*  \* 运算符分类：  \* 算术运算符，赋值运算符，关系运算符，逻辑运算符，三元运算符。  \*  \* 算术运算符：  \* +，-，\*，/的基本使用  \*/  **public** **class** OperatorDemo {  **public** **static** **void** main(String[] args) {  // 定义两个变量  **int** a = 3;  **int** b = 4;  System.***out***.println(a + b);  System.***out***.println(a - b);  System.***out***.println(a \* b);  System.***out***.println(a / b);  // 整数相除只能得到整数，要想得到小数，就必须有浮点数参与运算  System.***out***.println(3 / 4.0);  System.***out***.println(3.0 / 4);  }  } |

### 算数运算符取余和除法的区别

%：取余运算符。得到的是两个相除数据的余数。

/：除法运算符。得到是两个相除数据的商。

使用场景： %，判断两个数据是否整除。

### 代码案例十三

|  |
| --- |
| **package** com.igeek\_01;  /\*\*  \* **@ClassName**: HelloWorld  \* **@Description**: 算数运算符取余和除法的区别  \* **@date** 2017年10月11日 下午6:49:38  \* Company www.igeekhome.com  \*  \*/  **public** **class** OperatorDemo2 {  **public** **static** **void** main(String[] args) {  **int** a = 5;  **int** b = 3;  System.***out***.println(a / b);  System.***out***.println(a % b);  }  } |

### 字符和字符串参与加法操作

字符参与运算

其实是拿该字符对应的数值来操作

‘a’ 97

‘A’ 65

‘0’ 48

字符串参与运算

这里其实做的不是加法运算，而是字符串拼接。

字符串和其他类型的数据做拼接，结果是字符串类型的。

### 代码案例十四

|  |
| --- |
| **package** com.igeek\_01;  /\*\*  \* **@ClassName**: OperatorDemo3  \* **@Description**: 字符和字符串参与加法操作的区别  \* **@date** 2017年10月11日 下午7:10:27  \* Company www.igeekhome.com  \*  \* 整数的加法。  \* 字符参与加法操作。拿字符在计算机中底层存储对应的数据值来参与运算的。  \* '0' 48  \* 'a' 97  \* 'A' 65  \* 字符串参与加法操作。  \* 这里的+其实不是加法，而是字符串连接符。  \*/  **public** **class** OperatorDemo3 {  **public** **static** **void** main(String[] args) {  // 整数加法  **int** a = 10;  **int** b = 20;  System.***out***.println(a + b);  System.***out***.println("------------------");  // 字符参与加法操作  **char** c = '0';  **char** c2 = 'a';  System.***out***.println(a + c);  System.***out***.println(a + c2);  System.***out***.println("------------------");  // 字符串参与加法操作  System.***out***.println("hello" + a);  System.***out***.println("hello" + a + b); // "hello"+10,然后再和b进行拼接  System.***out***.println(a + b + "hello");  }  } |

### 算数运算符++和--的用法

++,--运算符：对变量做加1或者减1的操作。

++或者--既可以放在变量的后面，也可以放在变量的前面。

单独使用的时候，++或者--无论是放在变量的前面还是后面，结果是一样的。

参与操作的时候：

如果++或者--在变量的后面，先拿变量参与操作，后变量做++或者--

如果++或者--在变量的前面，先变量做++或者--，后拿变量参与操作

### 代码案例十五

|  |
| --- |
| **package** com.igeek\_01;  /\*\*  \* **@ClassName**: OperatorDemo4  \* **@Description**: 算数运算符++和--的用法  \* **@date** 2017年10月11日 下午7:13:51  \* Company www.igeekhome.com  \*  \*/  **public** **class** OperatorDemo4 {  **public** **static** **void** main(String[] args) {  **int** a = 10;  System.***out***.println("a:" + a);  // 单独使用  // a++;  // ++a;  // System.out.println("a:" + a);  // 参与操作使用  // int b = a++;  **int** b = ++a;  System.***out***.println("a:" + a);  System.***out***.println("b:" + b);  }  } |

## 赋值运算符

### 赋值运算符分类

基本的赋值运算符：=

扩展的赋值运算符：+=,-=,\*=,/=,%=

+=: a+=20;相当于a = (a的数据类型)(a + 20);

### 代码案例十六

|  |
| --- |
| **package** com.igeek\_02;  /\*\*  \* **@ClassName**: OperatorDemo  \* **@Description**: 赋值运算符的使用  \* **@date** 2017年10月11日 下午7:16:38  \* Company www.igeekhome.com  \*  \* 赋值运算符：  \* A:基本 =  \* B:扩展 +=,-=,\*=,...  \*  \* +=:  \* a+=20;  \* 相当于  \* a = (a的数据类型)(a + 20);  \*/  **public** **class** OperatorDemo {  **public** **static** **void** main(String[] args) {  // 把10赋值给int类型的变量a  **int** a = 10;  // += 把左边和右边的数据进行运算，最后赋值给左边。左边的只能是变量  a += 10;// 相当于a = a + 10  System.***out***.println("a:" + a);  System.***out***.println("----------------------");  **short** s = 10;  // s += 20; // 相当于 s = s + 20;  s = (**short**) (s + 20);  System.***out***.println("s:" + s);  }  } |

## 关系运算符

### 基本使用及注意事项

关系运算符包含以下内容:

==,!=,>,>=,<,<=

关系运算符的结果都是boolean型，也就是要么是true，要么是false。

注意事项:

关系运算符“==”不能误写成“=”。

### 代码案例十七

|  |
| --- |
| **package** com.igeek\_03;  /\*\*  \* **@ClassName**: OperatorDemo  \* **@Description**: 关系运算符  \* **@date** 2017年10月11日 下午7:19:36  \* Company www.igeekhome.com  \*  \* 关系运算符：  \* ==,!=,>,>=,<,<=  \* 关系运算符的结果是boolean类型。  \*  \* 注意：  \* 千万不要把==写成=  \*/  **public** **class** OperatorDemo {  **public** **static** **void** main(String[] args) {  **int** a = 10;  **int** b = 20;  **int** c = 10;  System.***out***.println(a == b);  System.***out***.println(a == c);  System.***out***.println("-----------------");  System.***out***.println(a != b);  System.***out***.println(a != c);  System.***out***.println("-----------------");  System.***out***.println(a > b);  System.***out***.println(a > c);  System.***out***.println("-----------------");  System.***out***.println(a >= b);  System.***out***.println(a >= c);  System.***out***.println("-----------------");  **int** x = 3;  **int** y = 4;  // System.out.println(x == y);  // System.out.println(x = y);// 把y赋值给x，把x的值输出  **boolean** bb = (x == y);  // 报错  // boolean cc = (x = y);  **int** cc = (x = y);  }  } |

## 逻辑运算符

### 逻辑运算符概述

A:逻辑运算符有哪些

&,|,^,!

&&,||

B:案例演示

逻辑运算符的基本用法

C:注意事项：

a:逻辑运算符一般用于连接boolean类型的表达式或者值。

b:表达式：就是用运算符把常量或者变量连接起来的符合java语法的式子。

算术表达式：a + b

比较表达式：a == b(条件表达式)

D:结论：

&逻辑与:有false则false。

|逻辑或:有true则true。

^逻辑异或:相同为false，不同为true。

!逻辑非:非false则true，非true则false。

### 代码案例十八

|  |
| --- |
| **package** com.igeek\_04;  /\*\*  \* **@ClassName**: OperatorDemo  \* **@Description**: 逻辑运算符：用于连接关系表达式  \* **@date** 2017年10月11日 下午7:23:27  \* Company www.igeekhome.com  \*  \* 逻辑运算符：用于连接关系表达式。  \* &,|,^,!  \* &&,||  \*  \* 与：& 有false则false  \* 或：| 有true则true  \* 异或：^ 相同则false,不同则true。(男女朋友)  \* 非：! true则false,false则true  \*/  **public** **class** OperatorDemo {  **public** **static** **void** main(String[] args) {  **int** a = 10;  **int** b = 20;  **int** c = 30;  System.***out***.println((a > b) & (a > c));// false & false  System.***out***.println((a < b) & (a > c)); // true & false  System.***out***.println((a > b) & (a < c)); // false & true  System.***out***.println((a < b) & (a < c)); // true & true  System.***out***.println("---------------");  System.***out***.println((a > b) | (a > c));// false | false  System.***out***.println((a < b) | (a > c)); // true | false  System.***out***.println((a > b) | (a < c)); // false | true  System.***out***.println((a < b) | (a < c)); // true | true  System.***out***.println("---------------");  System.***out***.println((a > b) ^ (a > c));// false ^ false  System.***out***.println((a < b) ^ (a > c)); // true ^ false  System.***out***.println((a > b) ^ (a < c)); // false ^ true  System.***out***.println((a < b) ^ (a < c)); // true ^ true  System.***out***.println("---------------");  System.***out***.println((a > b)); // false  System.***out***.println(!(a > b)); // !false  System.***out***.println(!!(a > b)); // !!false  }  } |

### 逻辑运算符&&与&的区别

A:&&和&的区别?

a:最终结果一样。

b:&&具有短路效果。左边是false，右边不执行。

&是无论左边是false还是true,右边都会执行

B:||和|的区别?

a:最终结果一样

b:||具有短路效果.左边是true,右边不执行

|是无论左边是false还是true,右边都会执行

### 代码案例十九

|  |
| --- |
| **package** com.igeek\_04;  /\*\*  \* **@ClassName**: OperatorDemo2  \* **@Description**: 逻辑运算符&&与&的区别  \* **@date** 2017年10月11日 下午7:25:23  \* Company www.igeekhome.com  \*  \* &&和&的结果一样  \* ||和|的结果一样  \*  \* &&和&的区别：  \* &&如果左边是false，右边不执行。  \* &无论左边是true还是false，右边都会执行。  \*/  **public** **class** OperatorDemo2 {  **public** **static** **void** main(String[] args) {  **int** a = 10;  **int** b = 20;  **int** c = 30;  System.***out***.println((a > b) && (a > c));// false && false  System.***out***.println((a < b) && (a > c)); // true && false  System.***out***.println((a > b) && (a < c)); // false && true  System.***out***.println((a < b) && (a < c)); // true && true  System.***out***.println("---------------");  System.***out***.println((a > b) || (a > c));// false || false  System.***out***.println((a < b) || (a > c)); // true || false  System.***out***.println((a > b) || (a < c)); // false || true  System.***out***.println((a < b) || (a < c)); // true || true  System.***out***.println("---------------");  **int** x = 3;  **int** y = 4;  // System.out.println((x++ > 4) & (y++ > 5)); // false & false  System.***out***.println((x++ > 4) && (y++ > 5)); // false && false  System.***out***.println("x:" + x);  System.***out***.println("y:" + y);  }  } |

## 三元运算符

### 三元运算符概述

A:格式

(关系表达式)?表达式1：表达式2；

如果条件为true，运算后的结果是表达式1；

如果条件为false，运算后的结果是表达式2；

B:示例：

获取两个数中大数。

int x=3,y=4,z;

z = (x>y)?x:y;//z变量存储的就是两个数的大数

### 代码案例二十

|  |
| --- |
| **package** com.igeek\_05;  /\*\*  \* **@ClassName**: OperatorDemo  \* **@Description**: 三元运算符  \* **@date** 2017年10月11日 下午7:28:17  \* Company www.igeekhome.com  \*  \* 三元运算符：  \*  \* 格式：  \* (关系表达式)?表达式1:表达式2;  \* 执行流程:  \* A:计算关系表达式的值，看是true还是false  \* B:如果是true,表达式1就是运算结果  \* 如果是false,表达式2就是运算结果  \*/  **public** **class** OperatorDemo {  **public** **static** **void** main(String[] args) {  **int** a = 10;  **int** b = 20;  **int** c = (a > b) ? a : b;  System.***out***.println("c:" + c);  }  } |

### 代码案例二十一(比较两个数是否相同)

|  |
| --- |
| **package** com.igeek\_05;  /\*\*  \* **@ClassName**: OperatorTest  \* **@Description**: 三元运算符的练习  \* **@date** 2017年10月11日 下午7:30:11  \* Company www.igeekhome.com  \*  \* 三元运算符的练习  \* 比较两个整数是否相同  \*/  **public** **class** OperatorTest {  **public** **static** **void** main(String[] args) {  // 定义两个int类型的变量  **int** a = 10;  **int** b = 20;  **boolean** flag = (a == b) ? **true** : **false**;  // boolean flag = (a == b);  System.***out***.println(flag);  }  } |

### 代码案例二十二(获取三个数中的最大值)

|  |
| --- |
| **package** com.igeek\_05;  /\*\*  \* **@ClassName**: OperatorTest2  \* **@Description**: 获取三个整数中的最大值  \* **@date** 2017年10月11日 下午7:31:48  \* Company www.igeekhome.com  \*  \* 获取三个整数中的最大值  \*/  **public** **class** OperatorTest2 {  **public** **static** **void** main(String[] args) {  // 定义三个int类型的变量  **int** a = 10;  **int** b = 30;  **int** c = 20;  // 先比较两个整数的大值  **int** temp = ((a > b) ? a : b);  **int** max = ((temp > c) ? temp : c);  System.***out***.println("max:" + max);  }  } |

## 重点和小结

1、使用(+,-,\*,/,%,++,--等)算数运算符

2、取余和除法的区别

3、字符和字符串参与加法运算操作需要注意的事项

4、++和--的作用，以及++，--在前或在后的区别

5、赋值运算符和扩展运算符

6、关系运算符，关系运算符的结果

7、逻辑运算符的基本使用及&与&&的区别

8、三元运算符

# 键盘录入

## 键盘录入的基本步骤

### 键盘录入数据概述

我们目前在写程序的时候，数据值都是固定的，但是实际开发中，数据值肯定是变化的，所以，把数据改进为键盘录入，提高程序的灵活性。

键盘录入数据的步骤:

A:导包(位置放到class定义的上面)

import java.util.Scanner;

B:创建对象

Scanner sc = new Scanner(System.in);

C:接收数据

int x = sc.nextInt();

### 代码案例二十三

|  |
| --- |
| **package** com.igeek\_01;  **import** java.util.Scanner;  /\*\*  \* **@ClassName**: ScannerDemo  \* **@Description**: 实现键盘录入数据  \* **@date** 2017年10月11日 下午7:34:38  \* Company www.igeekhome.com  \*  \* 为了提高程序的灵活性，我们就把数据改进为键盘录入。  \* 如何实现键盘录入呢?目前我们只能使用JDK提供的类Scanner。  \* 这个使用的步骤，目前大家记住就可以了。  \*  \* 使用步骤：  \* A:导包  \* import java.util.Scanner;  \* 类中的顺序：package > import > class  \* B:创建对象  \* Scanner sc = new Scanner(System.in);  \* C:接收数据  \* int i = sc.nextInt();  \*/  **public** **class** ScannerDemo {  **public** **static** **void** main(String[] args) {  //创建键盘录入数据的对象  Scanner sc = **new** Scanner(System.***in***);    //接收数据  System.***out***.println("请录入一个整数：");  **int** i = sc.nextInt();    //输出数据  System.***out***.println("i:"+i);  }  } |

## 键盘录入的练习

### 键盘录入两个数据并求和

键盘录入两个数据，并对这两个数据求和，输出其结果

键盘录入：

A:导包

B:创建对象

C:接收数据

### 代码案例二十四

|  |
| --- |
| **package** com.igeek\_02;  **import** java.util.Scanner;  /\*\*  \* **@ClassName**: ScannerTest  \* **@Description**: 键盘录入两个数据并求和  \* **@date** 2017年10月11日 下午7:37:48  \* Company www.igeekhome.com  \*  \*/  **public** **class** ScannerTest {  **public** **static** **void** main(String[] args) {  // 创建对象  Scanner sc = **new** Scanner(System.***in***);  // 接收数据  System.***out***.println("请输入第一个数据：");  **int** a = sc.nextInt();  System.***out***.println("请输入第二个数据：");  **int** b = sc.nextInt();  // 对数据进行求和  **int** sum = a + b;  System.***out***.println("sum:" + sum);  }  } |

### 键盘录入两个数据比较是否相等

键盘录入两个数据，比较这两个数据是否相等

### 代码案例二十五

|  |
| --- |
| **package** com.igeek\_02;  **import** java.util.Scanner;  /\*\*  \* **@ClassName**: ScannerTest2  \* **@Description**: 键盘录入两个数据，比较这两个数据是否相等  \* **@date** 2017年10月11日 下午7:39:40  \* Company www.igeekhome.com  \*  \*/  **public** **class** ScannerTest2 {  **public** **static** **void** main(String[] args) {  // 创建对象  Scanner sc = **new** Scanner(System.***in***);  // 接收数据  System.***out***.println("请输入第一个数据：");  **int** a = sc.nextInt();  System.***out***.println("请输入第二个数据：");  **int** b = sc.nextInt();  // 比较两个数据是否相等  // boolean flag = ((a == b) ? true : false);  **boolean** flag = (a == b);  System.***out***.println("flag:" + flag);  }  } |

### 键盘录入三个数据获取最大值

键盘录入三个数据，获取这三个数据中的最大值

### 代码案例二十六

|  |
| --- |
| **package** com.igeek\_02;  **import** java.util.Scanner;  /\*\*  \* **@ClassName**: ScannerTest3  \* **@Description**: 键盘录入三个数据，获取这三个数据中的最大值  \* **@date** 2017年10月11日 下午7:41:20  \* Company www.igeekhome.com  \*  \*/  **public** **class** ScannerTest3 {  **public** **static** **void** main(String[] args) {  // 创建对象  Scanner sc = **new** Scanner(System.***in***);  // 接收数据  System.***out***.println("请输入第一个数据：");  **int** a = sc.nextInt();  System.***out***.println("请输入第二个数据：");  **int** b = sc.nextInt();  System.***out***.println("请输入第三个数据：");  **int** c = sc.nextInt();  // 如何获取三个数据的最大值  **int** temp = (a > b ? a : b);  **int** max = (temp > c ? temp : c);  System.***out***.println("max:" + max);  }  } |

重点和总结

1、Eclipse的安装、配置及使用

2、运算符：+,-,\*,/,%,++,--,&&,&,||,|

3、键盘录入