

安装｜配置与使用

对应此节课程的视频，请于此下载：<http://pan.baidu.com/s/1hq4zWhe>

# 课程主要内容

  1. Eclipse的相关介绍

2. JDK的安装和配置

3. Eclipse的安装和启动

4. Eclipse的相关概念和设置(重点)

5.Eclipse的相关操作(重点)

6. Eclipse的快捷键(难点)

7. Eclipse的Debug调试(难点)

# 1. What is Eclipse

1). Eclipse是由IBM开发的一个**集成开发工具**(IDE)

2). 特点:

a. 免费,开源(Java)

b. 可以较容易的实现功能的扩展(插件)

c. 用Java语言写的, 但现在可以针对多种语言开发

详细可查阅百度百科:eclipse

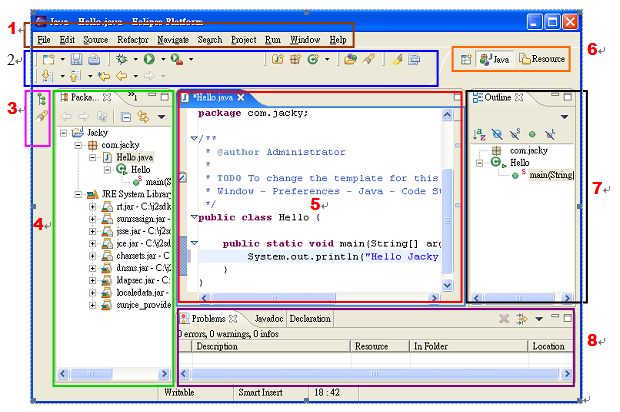
# 2. Why use Eclipse

1. 用文件编辑器工具来开发应用, 效率实在太低了, 而IDE可以极大的提交开发效率

2. 与JBuider,NetBean等同类型的IDE相比, Eclipse的使用更便捷, 功能更强大

3. 现在绝大部分软件公司都是使用Eclipse(MyEclipse)作为开发工具

# 3. How use Eclipse



|  |
| --- |
| 1.菜单(Menu Bar)               2.工具列(Tool Bar)  3.快速视图(Fast View)          4.Package Explorer视图  5.Editor视图                   6.快捷方式工具列(Shortcut Toolbar)  7.Outline视图                  8.Tasks视图和Console视图 |
|  |

## 1). 搭建环境

### 1.1). JDK的安装和配置

1. 下载JDK7的安装文件(32位/64位)

  a. Oracle官网下载地址: [jdk7下载](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)

  b. 尚硅谷官网下载地址: [jdk7下载](http://atguigu.com/download.shtml)

2. 双击jdk-7u45-windows-i586.exe即可开始安装, 安装过程中设置

安装目录(不可在包含中文的路径下)

3. 配置path(why?)

4. 检查安装配置是否成功

### 1.2). Eclipse的安装和启动

1. **下载eclipse的zip包文件(Java版/JavaEE版)**

a. eclipse官网下载地址: [eclipse下载](http://www.eclipse.org/downloads/)

b. 尚硅谷官网下载地址: [eclipse下载](http://atguigu.com/download.shtml)

2. **将下载的zip文件解压到安装目录下即完成安装**(不包含中文的路径)

3. **双击eclipse.exe文件即可启动eclipse了**

## 2). 相关概念及设置

### 2.1). workspace(工作空间)

它对应一个文件夹, 你用当前eclipse开发创建的项目都在此文件夹下

### graphic2.2). workbench(工作台)

### 2.3). View(视图)

常用

Package Explorer

包浏览器视图:

   显示当前workspace中的所有项目(以**包结构**查看项目中编写的所有java类)

Editor

编辑视图:

   编写代码和显示已有代码的View

Outline

大纲视图:

   显示当前Editor中文件的结构

Navigator

导航视图(文件浏览器视图):

 可以查看项目文件夹下所有的文件和文件夹

Console

控制台视图:

 显示打印信息

时用

Type Hierarchy

类型树结构视图:

显示当前类及其以上所有类的继承结构

Search

搜索视图:

   在应用中查询某个字符串, 所有与其匹配的结果都显示在此视图中

JUnit

单元测试视图:

    显示单元测试是否通过

### 2.4). Perspective(视景，视窗组合)

Perspective就是多个视图的组合, 可以进行调整, 并可以定义自己的视图组合

Java

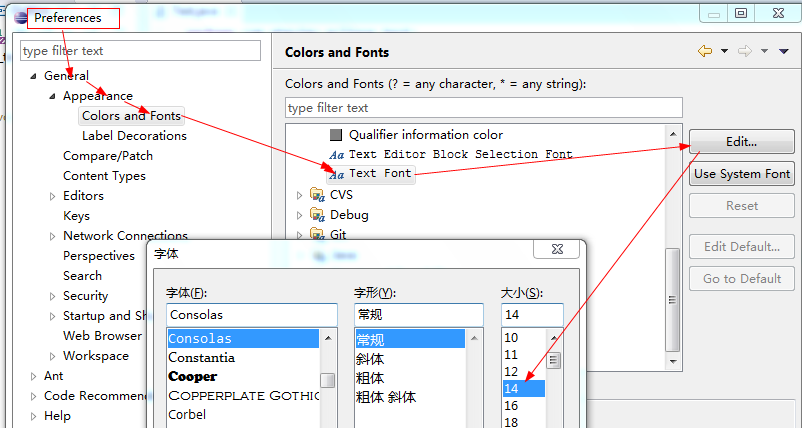
Java EE

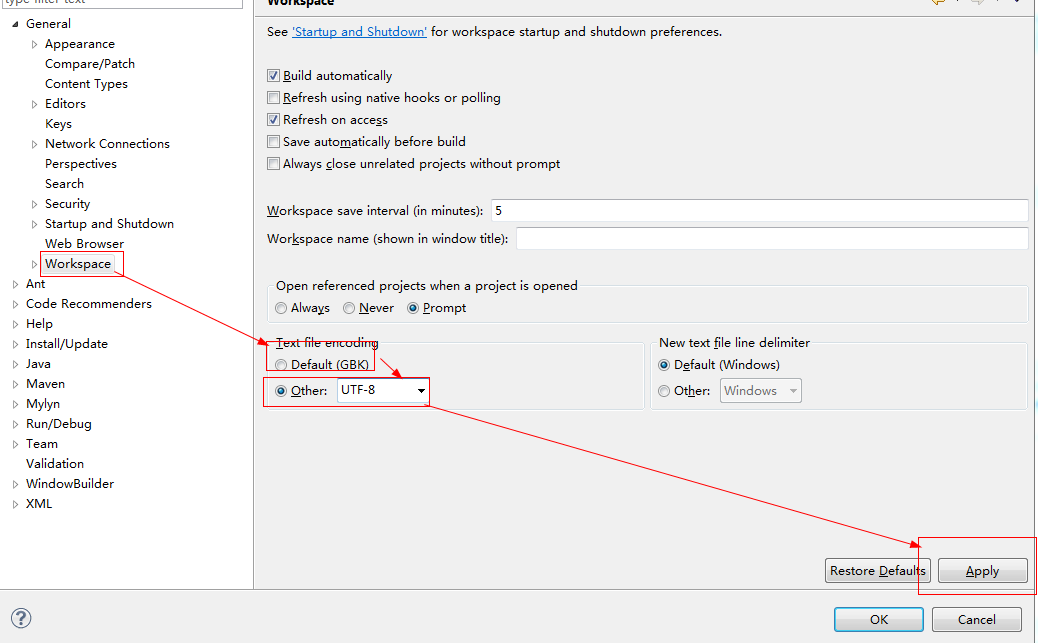
Debug

自定义

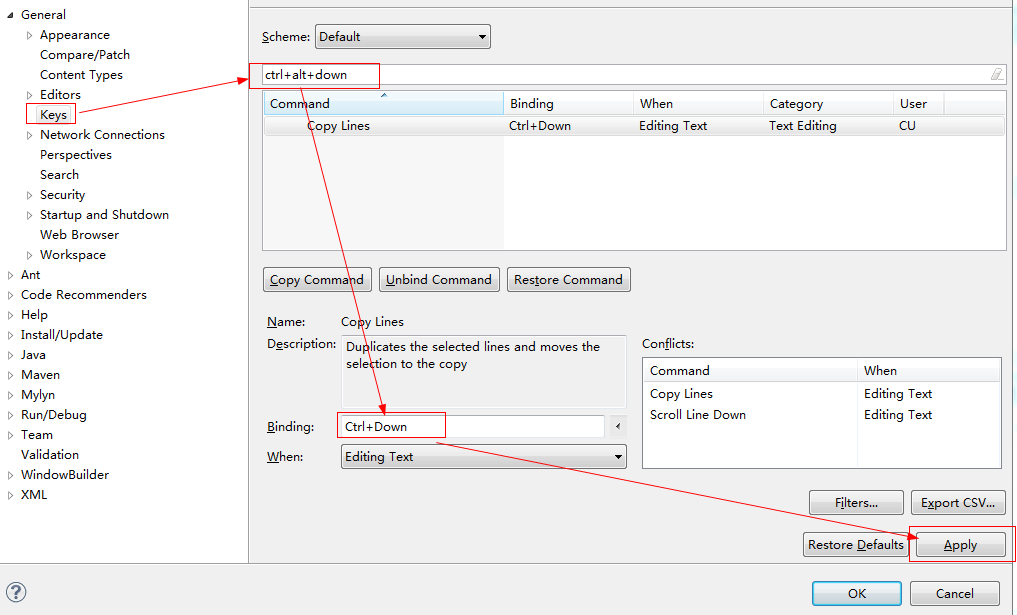
### 2.5). Project(工程)

### 2.6). Preferences(首选项)

设置字体

设置字符编码

设置快捷键



## 3). 相关操作

### 创建一个工程

### 创建一个类

### 运行一个类

### 导入一个外面的工程

## 4). 快捷键

### 4.1). 常用快捷键

#### 常用

alt+/ 代码助手完成一些代码的插入

Ctrl+1 快速修复

Shift+Enter 在当前行的下一行插入空行(这时鼠标可以在当前行的任一位置,不一定是最后)

Ctrl+Shift+Enter 在当前行插入空行(原理同上条)

Ctrl+D: 删除当前行

Ctrl+Alt+↓ 复制当前行到下一行(复制增加) //冲突孜为:ctrl+↓

Ctrl+Alt+↑ 复制当前行到上一行(复制增加) //冲突孜为:ctrl+↑

Alt+↓ 当前行和下面一行交互位置(特别实用,可以省去先剪切,再粘贴了)

Alt+↑ 当前行和上面一行交互位置(同上)

Ctrl+Shift+F 格式化当前代码

Ctrl+F 查找与替换

Ctrl+Shift+X 把当前选中的文本全部变味小写

Ctrl+Shift+Y 把当前选中的文本全部变为小写

Ctrl+/ 注释当前行或选中的行,再按则取消注释

Ctrl+Shift+/ 注释选中的行

Ctrl+Shift+\ 解注释

#### 时用

Ctrl+Q 定位到当前行的最前面(自定义,默认为home键)

Ctrl+W 定位到当前行的最后面(自定义,默认为end键)

Ctrl+Shift+/(小键盘) 收缩当前类的所有方法

Ctrl+/(小键盘) 展开当前类的所有方法

Ctrl+O 显示当前类的outlook快捷视图

Ctrl+K 参照选中的Word快速定位到下一个

Ctrl+Shift+K 参照选中的Word快速定位到上一个

Ctrl+L 定位到指定的行

Ctrl+E 显示所有打开的文件的下拉列表

Ctrl+M 最大化当前的Edit或View (再按则反之)

Ctrl+向左键/向右键 水平快速移动光标

Ctrl+Shift+O 导入所有需要的类

Ctrl+Shift+T 根据类名查找类

Ctrl+Shift+R 根据文件名查找文件(包括Java文件)

Ctrl+H 打开搜索对话框(根据文件内容出现的关键字)

Alt+Shift+S 显示常用封装,重写的菜单

Alt+shift+R 修改变量名或方法名

Alt+Shift+M 抽取方法

Alt+Enter 显示当前选择资源(工程,or 文件 or文件)的属性

Alt+← 前一个编辑的页面

Alt+→ 下一个编辑的页面(当然是针对上面那条来说了)

F3 打开定位到的类

F2 显示当前定位的类型的文档

F11 运行当前测试方法或main方法

### 4.2). 自定义快捷键

参考 : [设置快捷键](#设置快捷键)

## 5). Debug调试

### 5.1). 测试代码

|  |
| --- |
| **public** **class** DebugTest {  **public** **static** **void** main(String[] args) {  **int** i = 5;  **int** j = 3;     System.*out*.println("i=" + i + " j=" + j);  **int** max = *getMax*(i, j);     System.*out*.println("max=" + max);   }  **private** **static** **int** getMax(**int** k, **int** m) {  **int** max = 0;  **if** (k < m) {      max = k;     } **else** {      max = m;     }  **return** max;   }  } |

### 5.2). 开始调试

1 设置断点（可能会有多个）

2 debug运行 debug as application

3 常用操作：

|  |  |
| --- | --- |
| 操作 | 作用 |
| step over跳过(f6) | 执行完当前行的语句，进入下一行 |
| step into跳入(f5) | 进入当前行所调用的方法中 |
| step return跳回(f7) | 执行完当前行所在的方法，进入下一行 |
| drop to frame | 回到当前行所在方法的第一行 |
| resume 恢复 | 执行完当前行所在断点的所有代码，进入下一个断点，如果没有就结束 |
| Terminate 终止 | 停止JVM, 后面的程序不会再执行 |