



# 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

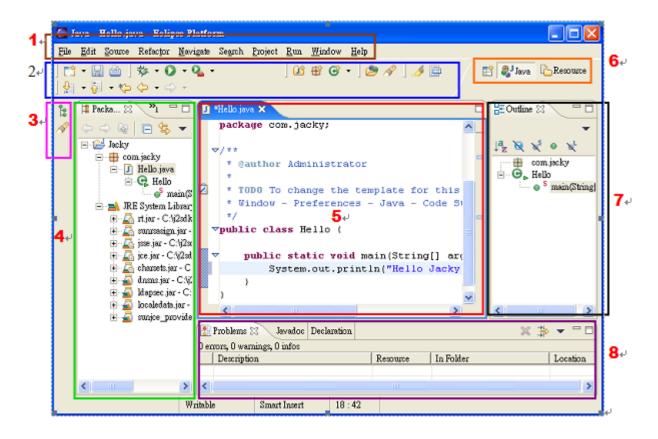
详细可查阅百度百科:eclipse

# 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
  - b. 尚硅谷官网下载地址: jdk7下载
- 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
  - b. 尚硅谷官网下载地址: eclipse 下载
- 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

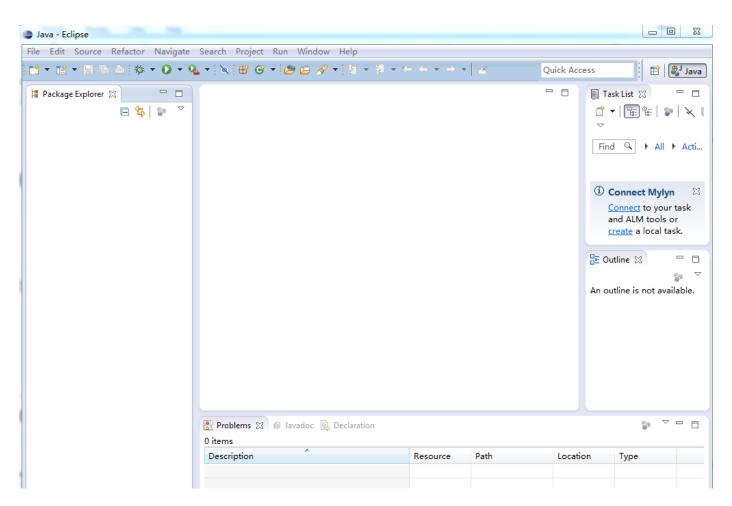


径)

- 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

### 2.2). 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

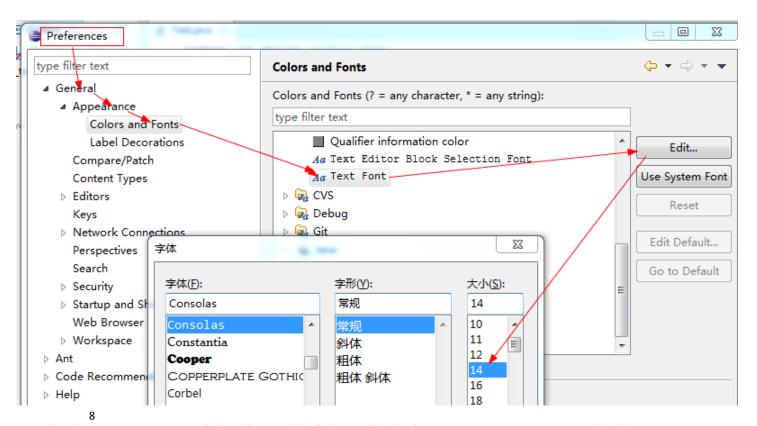
Java EE

**Debug** 

自定义

- 2.5). Project(工程)
- 2.6). Preferences(首选项)

### 设置字体



【更多 Java - Android 资料下载,可访问尚硅谷(中国)官网 <u>www.atguigu.com</u> 下载区】



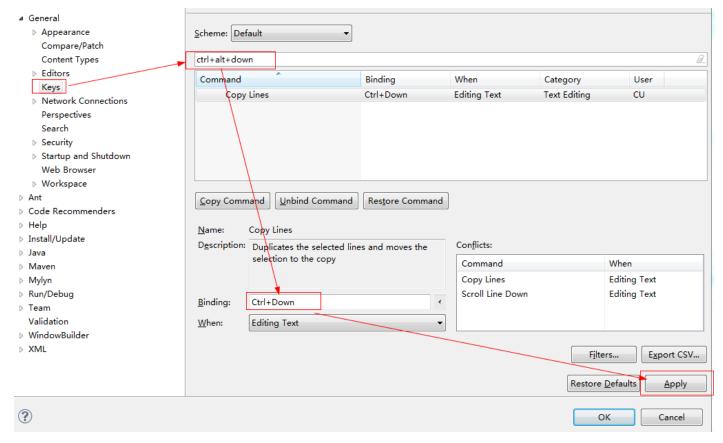
# 设置字符编码

type inter text	vvorkspace		~ · ~ · •
■ General	See <u>'Startup and Shutdown'</u> for workspace startup and shutdown preferences.		
> Appearance			
Compare/Patch	▼ Build automatically		
Content Types			
▶ Editors	Refresh using native hooks or polling		
Keys	▼ Refresh on access		
▶ Network Connections	Save automatically before build		
Perspectives	Always close unrelated projects without prompt		
Search			
	Workspace save interval (in minutes): 5		
Web Browser	Workspace name (shown in window title):		
⊳ Ant	Open referenced projects when a project is opened		
▶ Code Recommenders	○ Alwa <u>v</u> s ○ <u>N</u> ever <u> ® P</u> rompt		
⊳ Help			
▶ Install/Update	Text file encoding	New text <u>fi</u> le line delimiter	
Java	O Default (GBK)	<ul><li>Default (Windows)</li></ul>	
▶ Maven	⊚ Other: UTF-8 ▼	Other: Windows 🔻	
▶ Mylyn			
▶ Run/Debug			
⊳ Team			
Validation		_	
▶ WindowBuilder			
XML			
		Restore <u>D</u> efault	s <u>A</u> pply
?		OK	Cancel

# 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



# 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		