

第2章 Java 开发环境及开发工具

本章主要介绍 Java 开发环境的搭建，首先介绍 JDK 的下载安装和环境变量的设置，并通过一个简单的示例程序展示 JDK 的简单使用方法，对于 Java 开发工具方面，简单介绍集成开发环境 Eclipse 的基本使用方法，通过本章的学习，读者可以迅速掌握 Java 开发环境的搭建，并对 Eclipse 开发工具的基本用法有所了解。

2.1 下载安装 JDK

JDK 中包含了 Java 开发中必需的工具和 Java 程序的运行环境（即 JRE）。JDK 的安装文件可以从 <http://java.sun.com/javase/downloads/index.jsp> 下载，目前的 JDK 版本为 6.0，JDK 的安装文件有三个不同操作系统的版本，在这里我们使用 Windows 版本的 JDK，下载下来的文件为 jdk-6u2-windows-i586-p.exe。如果需要 6.0 之前版本的 JDK，也可以在 Sun 的官方网站进行下载，具体的链接地址为 <http://java.sun.com/javase/downloads/previous.jsp>，在这个地址中提供了 JDK1.3 以后的各个版本 JDK 安装文件下载服务。

在下面的内容中，对 JDK 的安装过程进行详细的说明。

(1) 双击运行下载下来的安装文件 jdk-6u2-windows-i586-p.exe，就可以进入如图 2.1 所示的 JDK 安装界面。

(2) 图 2.1 的欢迎界面在几秒中之后会自动切换到如图 2.2 所示的安装协议界面。



图 2.1 安装 JDK 的欢迎界面

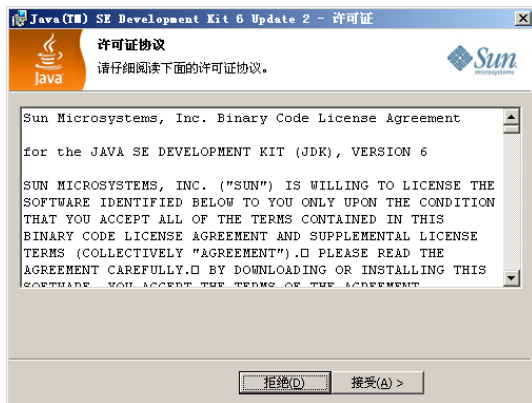


图 2.2 JDK 安装协议选择界面

(3) 在图 2.2 中，提供的是 JDK 的安装许可证协议信息，在这里只有选择接受，如果选择拒绝就会退出 JDK 的安装，在这里直接单击“接受”按钮，就会进入如图 2.3 所示的安装功能选择界面。

(4) 在图 2.3 所示的界面中，可以选择需要安装的功能，在这里接受默认的安装即可，在默认的安装中已经提供了基本的 Java 开发和运行环境。单击“下一步”按钮就可以进入如图 2.4 所示的安装界面。



图 2.3 JDK 安装功能选择界面



图 2.4 JDK 安装进度界面

(5) 在如图 2.4 的 JDK 安装进度结束以后，会自动进入如图 2.5 所示的 JRE 的安装界面，在 JDK 的默认安装中，会自动安装 JRE 的功能模块。

(6) 在如图 2.5 的安装界面中，可以选择 JRE 的功能模块和安装位置，在这里我们选择默认的设置即可，单击“下一步”按钮，可以进入如图 2.6 所示的 JRE 的安装进度界面。

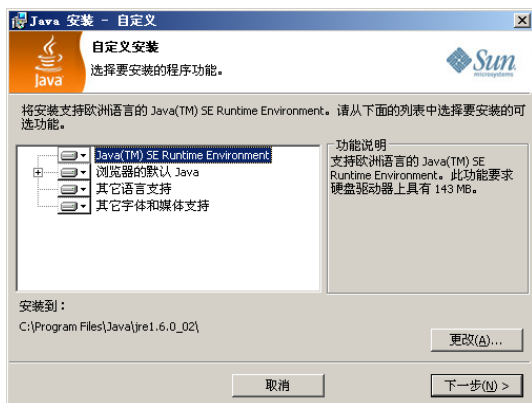


图 2.5 JRE 安装功能选择界面



图 2.6 JRE 安装进度界面

在如图 2.6 中的安装进度结束以后，就会自动进入如图 2.7 所示的结束信息提示界面。



图 2.7 JDK 安装结束界面

在图 2.6 所示的界面中单击“完成”按钮，就可以完成 JDK 的安装，到这一步为止，JDK 和 JRE 的安装工作已经全部结束，但是现在还不能马上使用 JDK 中提供的开发工具，JDK 安装结束之后，必需设置必要的环境变量，然后才能正常使用。在接下来的章节中将介绍 JDK 环境变量的设置方法。

2.2 环境变量设置

在上面的章节中，介绍了 JDK 的安装方法，但是在 JDK 安装结束之后，必需进行环境变量的设置，然后才可以使用 JDK 提供的开发工具。下面对环境变量的设置步骤进行详细的介绍。

(1) 右键单击“我的电脑”，在弹出的菜单中选择“属性”可以打开系统属性窗口，在系统属性选项卡中选择“高级”|“环境变量”，可以得到如图 2.8 所示的环境变量界面。

(2) 在如图 2.8 所示的界面中，可以对系统的环境变量进行设置，图中上半部分是用户变量，下半部分是系统变量，在下部的系统变量中，选择“Path”这个变量，然后单击“编辑”按钮，就可以进入如图 2.9 所示“Path”环境变量的编辑界面。

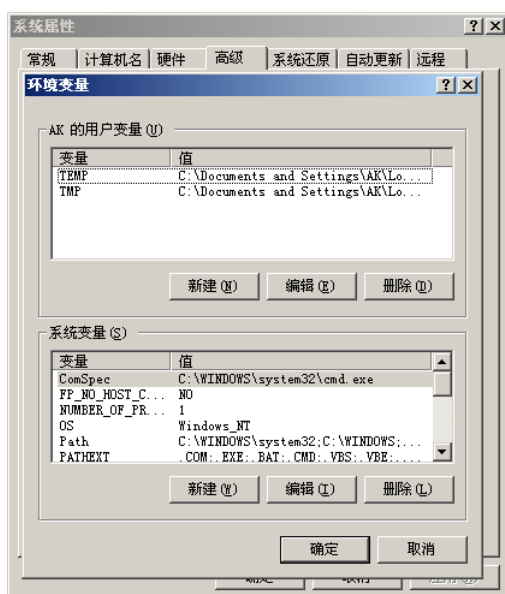


图 2.8 环境变量设置界面

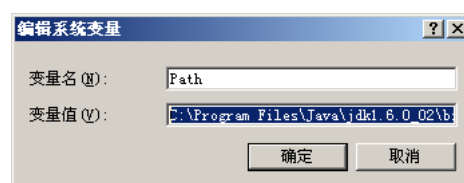


图 2.9 Path 变量编辑界面

(3) 在如图 2.8 所示的界面中，可以修改 Path 变量的值，在这里需要在 Path 变量中添加 JDK 的路径信息，例如在安装 JDK 的时候，选择的安装路径为“C:\Program Files\Java\jdk1.6.0_02”，所以需要在把 Path 变量值的最前面添加下面引号中的内容“C:\Program Files\Java\jdk1.6.0_02\bin;”，需要注意的是，在引号的内容中，最后有一个逗号，这个一定不能缺少。

经过上面的处理，JDK 的环境变量设置已经完成，下面需要测试 JDK 的安装配置是否成功。可以在 DOS 命令行中测试 JDK 是否安装成功。选择“开始”|“运行”，会弹出运行程序的选择界面，输入 cmd 回车，就可以进入 DOS 界面，在这个界面中输入 java - version，如果可以得到如图 2.10 所示的界面，说明 JDK 的安装配置已经成功。



图 2.10 JDK 安装配置测试界面

2.3 小试身手--HelloWorld

在上面两个小节的介绍中，已经成功安装配置 JDK，在 JDK 中提供了编译执行 Java 的基本工具，使用这些工具已经可以进行基本的 Java 程序的编写工作，虽然在使用继承的开发环境进行开发的效率会更高，但是，为了是读者对 JDK 的基本使用方法有基本的了解，在接下来的内容中，将不使用集成开发环境，而是通过 DOS 命令行对简单的 Java 示例程序进行编译和运行。

2.3.1 编辑 Java 源文件

在编辑 Java 源文件的时候，有很多工具可供选择，只要是能够进行简单文本编辑的工具都可以用来编辑 Java 源文件。在这里我们选择使用 Windows 中自带的记事本工具。在记事本中输入下面的代码。

```
//-----文件名: HelloWorld.java-----  
public class HelloWorld{  
    public static void main(String[] args){  
        System.out.println("Hello world!");  
    }  
}
```

上面这段 Java 代码中，仅仅在控制台打印“Hello World!”这个字符串。在记事本中输入这些代码以后，保存为 HelloWorld.java 文件，需要注意的是，Java 的源文件名必需和类名是相同的。否则源代码的编译就不能通过。完成这些操作，一个简单的 Java 文件的编辑工作就完成了。在接下来的内容中将对这个源文件进行编译。

2.3.2 编译 Java 源文件

完成对 Java 源文件的编辑工作以后，就可以对源代码进行编译，在 JDK 中提供了编译 Java 源文件的工具，可以在 DOS 命令行中调用 JDK 中的 javac 命令，这个命令可以对 Java 源文件进行编译。

例如上面的 HelloWorld.java 保存的路径为 C:\HelloWorld.java，那么需要把 DOS 命令中的当前路径切换到 HelloWorld.java 这个 Java 源文件所在的目录，然后调用 javac 对这个源文件进行编译，具体的处理过程如图 2.11 所示。

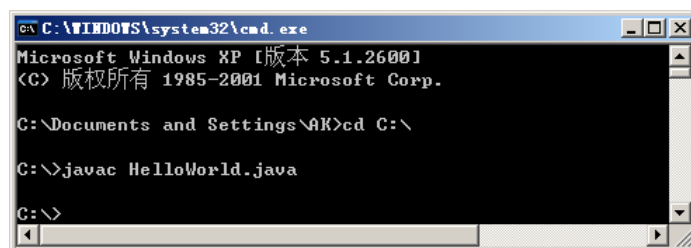


图 2.12 HelloWorld 示例程序编译过程

在图 2.12 所示的界面中，调用 javac 来编译 HelloWorld.java，在调用 javac 的时候需要注意的是，必需提供完整的 Java 源文件名，包括后缀名.java。在 javac 命令成功执行以后，会在 HelloWorld.java 的同一个目录中生成一个名为 HelloWorld.class 的文件，这是由 HelloWorld.java 源文件编译得到的 Java 类文件，在执行 Java 程序的时候，实际执行的就是编译得到的类文件。

2.3.3 执行类文件

在编译工作成功通过以后，可以得到对应的 Java 类文件，在 JDK 中同样提供了执行 Java 类文件的工具，可以在 DOS 命令行中调用 `java` 命令执行 Java 的类文件。在上面的操作中，成功编译了 `HelloWorld.java` 这个 Java 源文件，并在 C 盘的根目录下生成了 `HelloWorld.class` 文件，在 DOS 命令行中需要把当前的路径切换到 Java 类文件的目录，然后调用 JDK 中的 `java` 命令就可以执行这个类文件，具体的处理过程和运行结果如图 2.13 所示。



图 2.13 HelloWorld 示例程序运行过程

在如图 2.13 所示的界面中，调用 JDK 中的 `java` 命令执行 `HelloWorld.class` 这个类文件，在执行 `java` 命令的时候需要注意，只需要提供类文件的名称即可，不需要带 `.class` 的后缀名，JDK 中的 `java` 工具可以根据名称自动搜索所需要的类文件。

在这个简单的示例程序中，仅仅在控制台打印一个字符串，在图 2.13 中 `java` 命令执行完成就可以得到运行的结果。

在以上的内容中，简单介绍一个简单的 Java 程序的编写执行过程，通过这个程序的编译执行，读者可以对 Java 程序的开发有一个初步的认识，对于 JDK 中的 `javac` 工具和 `java` 工具也有基本的了解。其实在 JDK 中还提供了其他一系列的 Java 工具，在这里不再一一举例说明，在后面的章节中，会对涉及到的 JDK 工具的用法进行详细的说明。

2.4 开发工具 Eclipse 简介

在前面章节的内容中，介绍了直接使用 JDK 提供的工具开发一个简单的示例程序，在这个示例程序的开发过程中，没有使用任何集成的开发工具，这只是为了使读者对 JDK 的功能有一个大体的了解，在实际的开发过程中，是不可能脱离集成开发工具的帮助的，使用集成开发工具可以大大提高开发效率，从而保证项目的进度。

在本节的内容中，将简单介绍几种常用的 Java 开发工具，其中，对 Eclipse 开发平台会做比较详细的介绍。

2.4.1 Java 开发工具简介

在开发工具这方面，IBM、Borland、Sun 公司等也都推出了自己的 Java 开发工具，例如 `WSAD`、`JBuilder`、`NetBeans` 等，在开发 Java 程序的过程中，有很多开发工具可供选择，用户可以根据项目性质和用途的不同，选择适合项目需要的开发工具，目前常用的 Java 开发工具基本上可以分为两大类。

一种是简单小巧的开发工具。例如 `TextPad`、`JCreator` 等，这些开发工具的特点就是简单易用，只提

供最基本的编辑、编译、运行 Java 程序的功能，而没有提供太多的附加功能，例如在 TextPad 中，只能提供编辑、编译、运行的功能，无法创建工程，不能引入第三方类库，而且还没有提供 Debug 的功能。这类工具只适合初学者学习使用，或者开发简单的示例程序时使用，不能用来开发大型的工程。

另一种是具有强大功能的集成开发环境，例如 Eclipse、JBuilder 等，这些开发工具不仅仅提供最基本的编辑、编译功能，而且还提供了强大的附加功能，以 Eclipse 为例，在 Eclipse 中，可以创建不同类型的工程，Eclipse 可以根据工程类型的不同进行不同的配置，为用户准备好这种类型工程所需要的基本文件和基本的配置，这样用户就可以专注于项目自身业务逻辑的处理，而不用花费精力来准备配置项目的基本信息，而且在 Eclipse 中还提供了强大的 Debug 功能，用户可以通过这种功能迅速定位程序的错误，从而提高来开发的效率。

在实际的开发过程种，Eclipse、JBuilder 等集成开发工具都是很好的选择，这些集成开发工具提供的功能基本类似，读者可以根据自己的兴趣从中选择。

2.4.2 Eclipse 安装

在 Java 项目的开发过程中，越来越多的开发人员选择使用 Eclipse，在这里就介绍 Eclipse 开发环境的安装和使用。Eclipse 的安装文件可以从 <http://www.eclipse.org/downloads/> 下载，目前 Eclipse 最新的版本为 europa，下载下来的文件为 eclipse-java-europa-win32.zip，Eclipse 的安装非常简单，只需要把下载下来的文件解压缩到某路径下即可，解压后的目录结构如图 2.14 所示。

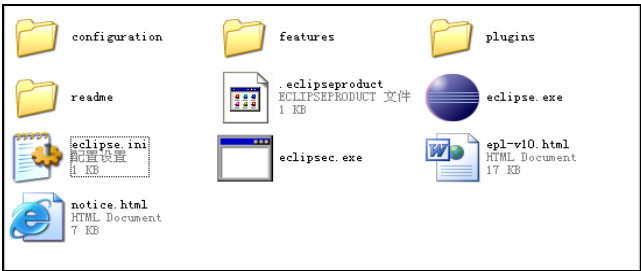


图 2.14 Eclipse 安装目录结构

在如图 2.14 所示的目录结构中，运行 eclipse.exe 就可以启动 Eclipse，Eclipse 也是用 Java 语言编写的，在使用 Eclipse 之前需要提前安装 JDK，Eclipse 可以自动从系统环境中寻找 JRE 路径，也正是这点特性，使 Eclipse 不用安装即可运行。

如果在运行 Eclipse.exe 的时候，出现如图 2.15 所示的界面，就说明 JDK 没有安装，或者是环境变量没有配置成功，这时候 Eclipse 找补到 JDK 的路径，所以就不能运行。



图 2.15 Eclipse 找不到 JDK 路径错误界面

上面这种错误情况需要重新检查 JDK 的安装配置是否成功，成功安装配置 JDK 以后，就可以正常运行 Eclipse，如图 2.16 就是 Eclipse 的开发界面。

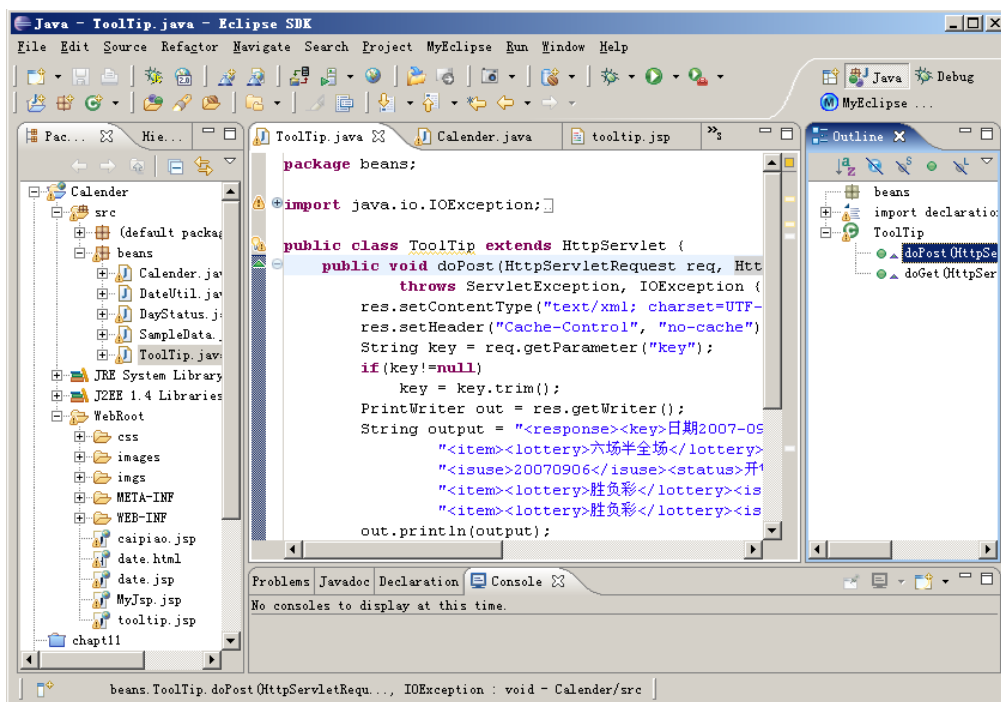


图 2.16 Eclipse 的开发界面

注意：在安装 Oracle 等软件的时候，往往会自动安装这些软件自带的 JDK，而且会修改 Path 系统变量值，这样的 JDK 版本就有可能不能支持 Eclipse 的运行，所以，在 Eclipse 不能运行，报出 JDK 错误的时候，就要考虑是否是这些软件修改了 Path 中的 JDK 路径。

2.4.3 Eclipse 使用简单例程

在 Eclipse 中，编译运行 Java 程序的方法和其他 Java 开发工具稍微有些不同，在本节的内容中，将对 Eclipse 的基本使用方法进行简单的介绍。在下面的内容中，将使用 Eclipse 创建并运行一个简单的工程。

(1) 在使用 Eclipse 进行 Java 开发的时候，首先要做的工作就是建立一个工程。在 Eclipse 的工具栏中选择“File”|“New”|“Project”就可以进入如图 2.17 所示的新建工程界面。

(2) 在如图 2.17 所示的界面中，选择“Java Project”，然后单击“Next”按钮就可以进入如图 2.18 所示的工程详细信息设置界面。

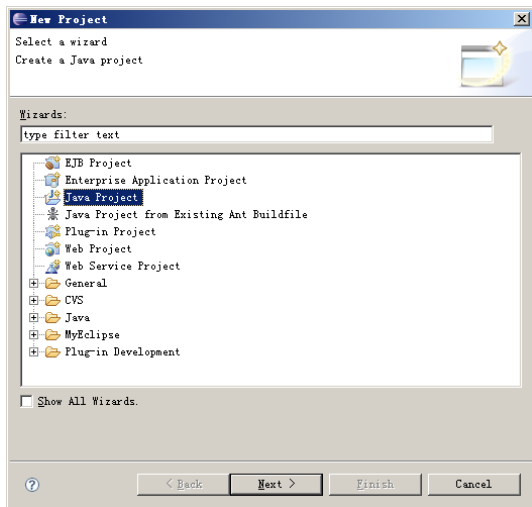


图 2.17 Eclipse 新建工程界面

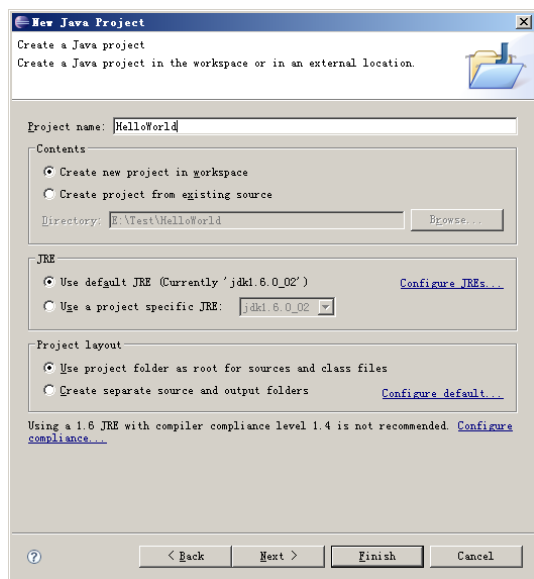


图 2.18 Eclipse 新建工程详细设置界面

因为在如图 2.18 所示的界面中，可以设置新建工程的具体信息，包括工程的名称、工程存放位置、所使用的 JRE 版本等信息。设置完这些信息以后，单击“Finish”按钮就可以结束工程的设置，这时候 Eclipse 会自动进入的工程开发视图。

(3) 现在开始创建 Java 文件，在 Eclipse 的工具栏中选择“File”|“New”|“Class”就可以进入如图 2.19 所示的新建 Java 类界面。

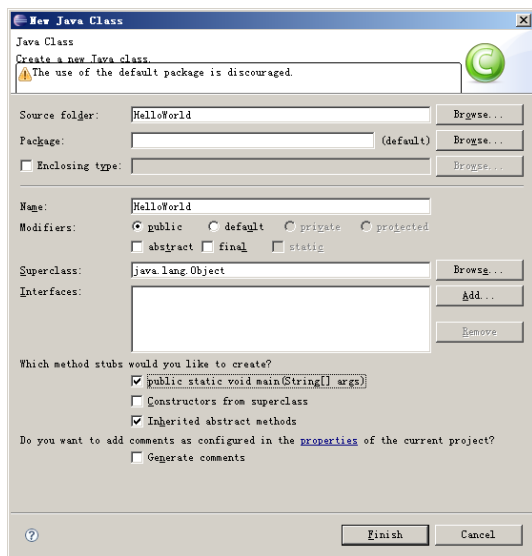


图 2.19 新建 Java 类界面

(4) 在图 2.19 所示的界面中，可以设置所要创建 Java 类的信息，例如类名、包路径、所需创建的方法等信息。在这里创建一个名为 HelloWorld 的 Java 类，选择使用默认的包路径，即把 HelloWorld 这个类放在工程的根目录下。然后在生成方法中选择自动创建 Main 方法。完成这些信息设置以后，单击“Finish”完成新建类的设置，就可以进入如图 2.20 所示的文件编辑界面。

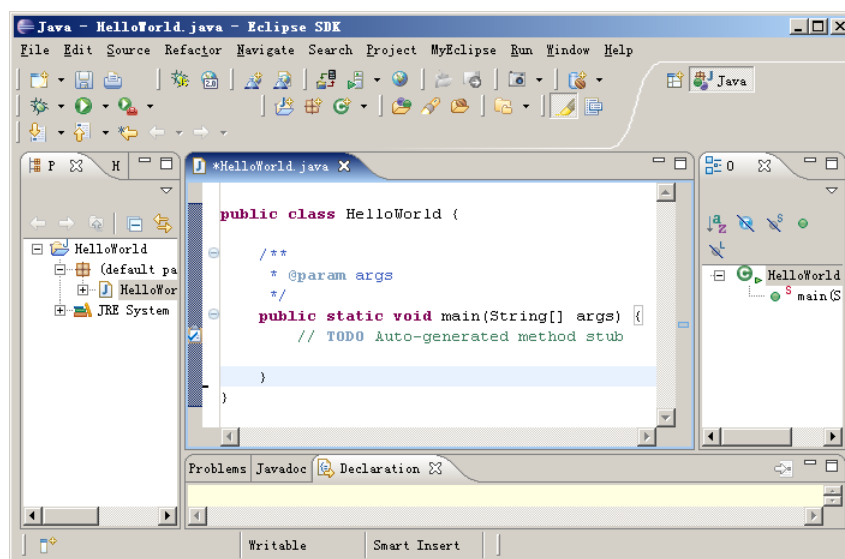


图 2.20 Eclipse 代码编辑界面

在如图 2.20 所示的界面中，已经生成了 HelloWorld 这个 Java 类的基本代码，用户只需要在 Main 方法中加入要运行的内容即可。在这里我们在 Main 方法中加入 `System.out.println("Hello World!");` 这行代码，然后保存，在保存 Java 文件的同时，Eclipse 已经自动对这个 Java 文件进行编译。

(5) 接下来要做的工作是运行 HelloWorld 这个 Java 类。在 Eclipse 开发界面左侧的目录树中右键单击 HelloWorld.java 这个文件，在弹出的菜单中选择“Run As”|“Java Application”，就可以运行 HelloWorld.java 这个文件，在如图 2.21 所示的控制台界面可以得到程序运行的结果。

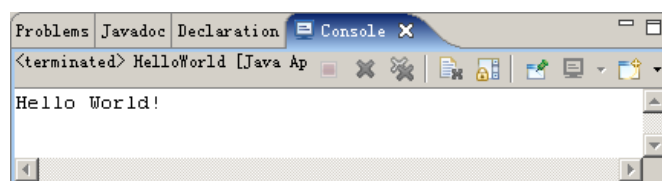


图 2.21 在 Eclipse 中 HelloWorld.java 的运行效果

2.4.4 在 Eclipse 中调试程序

在 Eclipse 中不仅可以方便的编辑执行 Java 程序，而且还提供了功能强大的调试功能，在调试 Java 程序的过程中，可以给程序设置断点，程序在运行到断点以后会暂停执行，通过设置断点，可以跟踪程序中的变量，从而对程序中的错误进行定位。

要调试程序，首先需要在 Java 源文件中添加断点，在如图 2.22 所示界面中左侧的阴影部分双击鼠标左键即可添加断点，图中的小圆点就是断点，断点和源代码中的对应行是一一对应的。如果要取消某个断点，只需要用鼠标右键单击断点，在弹出的菜单中选择“Toggle Breakpoint”就可以取消这个断点。一般情况下，只需要在怀疑程序出错的地方设置断点，用来跟踪错误。

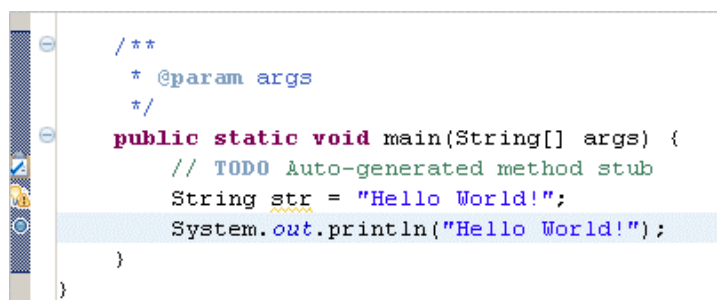


图 2.22 在 HelloWorld.java 中添加断点

在断点设置结束以后，就可以对程序进行调试，在 Eclipse 开发界面左侧的目录树中右键单击 HelloWorld.java 这个文件，在弹出的菜单中选择“Debug As”|“Java Application”，就可以进入如图 2.23 所示的调试界面。在图 2.23 所示的调试界面中，左上角的窗口中可以查看变量的值、断点和表达式，在中间的源代码窗口可以查看程序执行到的位置，在调试的过程中，默认的快捷键有 F8（恢复）、F5（进入方法）、F6（跳过语句）、F7（跳出方法），通过这些快捷键可以方便的控制调试的过程。

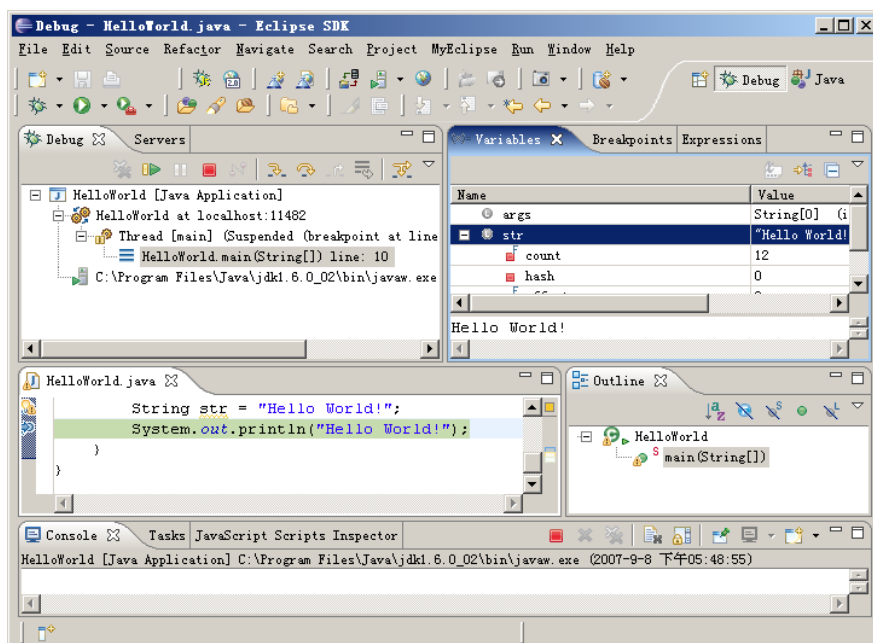


图 2.23 Eclipse 调试界面

2.4.5 Eclipse 常用快捷键

Eclipse 提供了丰富的辅助开发功能，而且很多常用的功能都提供了快捷键，在本节内容中，整理出一些相对比较常用的快捷键。编辑功能的快捷键如表 2.1 所示。

表 2.1 Eclipse 编辑快捷键

快捷键	功能	作用范围
Ctrl+F	查找、替换	全局
Ctrl+Shift+K	查找上一个	文本编辑器
Ctrl+K	查找下一个	文本编辑器
Ctrl+Z	撤销	全局

Ctrl+C	复制	全局
Alt+Shift+↓	恢复上一个选择	全局
Ctrl+X	剪切	全局
Ctrl+I	快速修正	全局
Alt+/	内容辅助	全局
Ctrl+A	全部选中	全局
Delete	删除	全局
Alt+?	上下文信息	全局
F2	显示工具提示描述	Java编辑器
Alt+Shift+↑	选择封装元素	Java编辑器
Alt+Shift+←	选择上一个元素	Java编辑器
Alt+Shift+→	选择下一个元素	Java编辑器
Ctrl+J	增量查找	文本编辑器
Ctrl+Shift+J	增量逆向查找	文本编辑器
Ctrl+V	粘贴	全局
Ctrl+Y	重做	全局

Eclipse 是一个多窗口的编辑器，在操作每个窗口的时候也提供了对应的快捷键，Eclipse 的窗口快捷键如表 2.2 所示。

表 2.2 Eclipse中的窗口快捷键

快捷键	功能	作用范围
F12	激活编辑器	全局
Ctrl+Shift+W	切换编辑器	全局
Ctrl+Shift+F6	上一个编辑器	全局
Ctrl+Shift+F7	上一个视图	全局
Ctrl+Shift+F8	上一个透视图	全局
Ctrl+F6	下一个编辑器	全局
Ctrl+F7	下一个视图	全局
Ctrl+F8	下一个透视图	全局
Ctrl+W	显示标尺上下文菜单	文本编辑器
Ctrl+F10	显示视图菜单	全局
Alt+-	显示系统菜单	全局

在 Eclipse 提供了导航的快捷键，具体设置如表 2.3 所示。

表 2.3 Eclipse中的导航快捷键

快捷键	功能	作用范围
Ctrl+F3	打开结构	Java编辑器
Ctrl+Shift+T	打开类型	全局
F4	打开类型层次结构	全局
F3	打开声明	全局
Shift+F2	打开外部javadoc	全局
Ctrl+Shift+R	打开资源	全局
Alt+←	后退历史记录	全局
Alt+→	前进历史记录	全局
Ctrl+,	上一个	全局

Ctrl+.	下一个	全局
Ctrl+O	显示大纲	Java编辑器
Ctrl+Shift+H	在层次结构中打开类型	全局
Ctrl+Shift+P	转至匹配的括号	全局
Ctrl+Q	转至上一个编辑位置	全局
Ctrl+Shift+ ↑	转至上一个成员	Java编辑器
Ctrl+Shift+ ↓	转至下一个成员	Java编辑器
Ctrl+L	转至行	文本编辑器

在 Eclipse 中提供了强大的搜索功能，这些搜索功能的快捷键如表 2.4 所示。

表 2.4 Eclipse中的搜索快捷键

快捷键	功能	作用范围
Ctrl+Shift+U	出现在文件中	全局
Ctrl+H	打开搜索对话框	全局
Ctrl+G	工作区中的声明	全局
Ctrl+ Shift+G	工作区中的引用	全局

Eclipse 中对文件的操作也提供了一系列的快捷键，具体设置如表 2.5 所示。

表 2.5 Eclipse中的文件快捷键

快捷键	功能	作用范围
Ctrl+S	保存	全局
Ctrl+P	打印	全局
Ctrl+F4	关闭	全局
Ctrl+Shift+S	全部保存	全局
Ctrl+Shift+F4	全部关闭	全局
Alt+Enter	属性	全局
Ctrl+N	新建	全局

Eclipse 中对源代码的操作也提供了一系列的快捷键，具体设置如表 2.6 所示。

表 2.6 Eclipse中的源代码快捷键

快捷键	功能	作用范围
Ctrl+Shift+F	格式化	Java编辑器
Ctrl+\	取消注释	Java编辑器
Ctrl+/	注释	Java编辑器
Ctrl+Shift+M	添加导入	Java编辑器
Ctrl+Shift+O	组织导入	Java编辑器

Eclipse 中对运行程序的操作也提供了一系列的快捷键，具体设置如表 2.7 所示。

表 2.7 Eclipse中的运行快捷键

快捷键	功能	作用范围
F7	单步返回	全局
F6	单步跳过	全局
F5	单步跳入	全局
Ctrl+F5	单步跳入选择	全局
F11	调试上次启动	全局
F8	继续	全局
Shift+F5	使用过滤器单步执行	全局
Ctrl+Shift+B	添加/去除断点	全局

Ctrl+D	显示	全局
Ctrl+F11	运行上次启动	全局
Ctrl+R	运行至行	全局
Ctrl+U	执行	全局

2.4.6 Java Web 开发工具 MyEclipse 简介

在开发 Java 桌面应用程序的时候,使用 Eclipse 是非常方便的,但是在进行 Web 开发的时候,Eclipse 的一些功能就不能够满足用户的需求了,例如在使用 Struts 或者是 Hibernate 这些开源框架的时候,在 Eclipse 中就没有很好的辅助工具,在这种情况下,自由自己安装 Eclipse 的对应插件,然而还有很多 Web 开发中的需求是 Eclipse 插件所不能满足的。

针对上面这种情况,在开发 Web 应用的时候,我们选择使用 MyEclipse 集成开发工具,MyEclipse 是依赖于 Eclipse 的一个开发工具,对 Eclipse 的功能进行了扩展,主要是给 Eclipse 增加了一系列的 Web 开发工具,从而是 Web 开发的效率大大提高。正是因为 MyEclipse 强大的 Web 开发功能,在目前的 Java Web 软件开发中,很大部分开发人员会选择是使用 Eclipse+MyEclipse 进行开发。

MyEclipse 的安装源文件可以从 MyEclipse 的官方网站 <http://www.myeclipseide.com> 下载,具体的下载地址为 <http://www.myeclipseide.com/module-htm/pages-display-pid-4.html>,在这里可以下载 MyEclipse 各种版本的安装文件,MyEclipse 的版本为 6.0,可以和 Eclipse3.3 配合使用,其他版本的 Eclipse 不能使用这个版本的 MyEclipse,在这里可以下载 All-in-One 的版本,在这个版本中,已经集成了 Eclipse,不需要另外安装 Eclipse,直接安装 MyEclipse 即可使用。MyEclipse 的安装过程比较简单,按照安装向导的指示就可以进行安装,安装以后无需设置就可以使用。

在 MyEclipse 安装成功以后,就可以创建 Web 工程,Web 工程的所有的基础配置信息都由 MyEclipse 来完成,MyEclipse 提供了一系列的 Web 开发工具,大大简化了 Java Web 开发的难度,提高了开发的效率,这样是 MyEclipse 受到开发人员喜欢的最大原因。

2.5 使用版本控制工具 CVS

在团队开发中,需要团队各个成员之间进行分工配合,这就给源代码的版本控制带来很大困难,不可能手工来完成源代码版本的迭代,在这里选择使用版本控制工具,通过版本控制工具完成对源代码的控制,各个团队成员只需要把修改过的版本提交给版本控制工具,有版本控制工具来把每个成员提交的版本整合成一个最新的版本。

在本节内容中,将介绍如何在 Eclipse 中使用版本控制工具,通过本节内容的学习,读者可以了解团队合作开发中版本控制的基本知识。

2.5.1 CVSNT 的安装与配置

在源代码版本控制方面,有很多工具可供选择,在这里选择使用 CVS 来对版本进行控制,其中 CVS 需要客户端和服务端配合使用,在使用 CVS 的时候,首先需要建立一个 CVS 服务器,然后团队中的每个成员此可以把自己的版本通过客户端提交给 CVS 服务器,从而由 CVS 服务器完成版本的整合更新任务。

在这里选择使用 CVSNT 作为 CVS 的服务器,CVSNT 的安装文件可以从 <http://www.cvsnt.org/archive/>

下载，在这里有各种版本的 CVSNT 安装文件可供下载，用户可以选择下载自己需要的版本，在本书中使用 cvsnt-2.5.03.2382，所以选择下载 cvsnt-2.5.03.2382.msi 这个文件即可。

CVSNT 的安装过程比较简单，在安装向导的指引下就可以顺行安装。安装完成以后，需要重启计算机，然后才能正常使用 CVSNT 的功能。

完成 CVSNT 安装以后，就可以配置 CVSNT 服务器，下面简单介绍 CVSNT 服务器的配置。

(1) 选择“开始”|“CVSNT”|“cvsnt control panel”，就可以进入如图 2.24 所示的 CVSNT 控制面板。在这个控制面板中，可以管理 CVSNT 的服务、添见 CVS 资源、设置 CVSNT 服务器的参数。

(2) 在如图 2.24 所示的界面中，选择“repository config”选项卡，鼠标单击“Add”按钮，就可以进入如图 2.25 所示添加资源的界面。

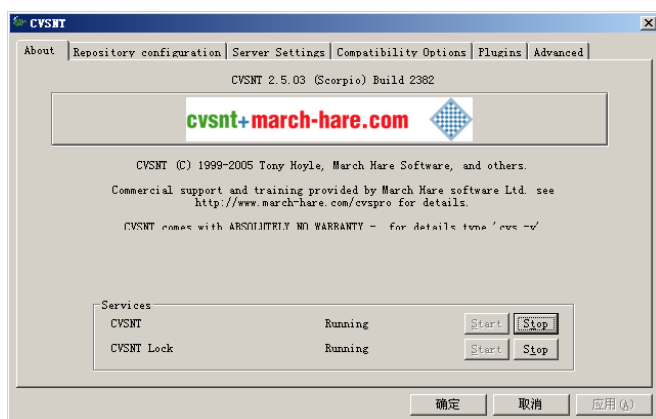


图 2.24 CVSNT 控制面板界面

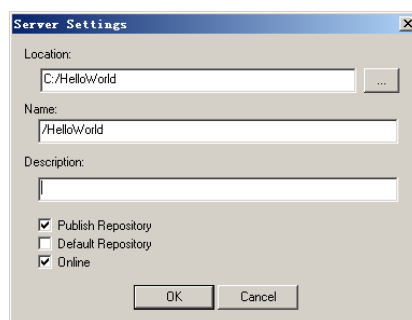


图 2.25 添加 CVS 资源界面

(3) 在如图 2.25 所示的界面中，可以把资源添加到 CVS 服务器中，其中，Location 是要添加到 CVS 服务器中的资源路径，Name 是 CVS 客户端访问这个资源需要的名称。Description 是这个 CVS 资源的描述信息。输入这些信息以后，单击“OK”按钮就完成了资源的添加。

(4) 资源添加完成以后，在如图 2.24 所示的界面中，重新启动 CVSNT 的两个服务，使刚刚添加的 CVS 资源设置生效，重启这两个服务以后，就可以通过 CVS 客户端访问这个 CVS 资源。

注意：当使用 CVS 客户端访问 CVSNT 服务器的时候，需要用户名和密码，这时候使用 CVSNT 服务器所在计算机的系统用户名和密码即可。

2.5.2 使用 Eclipse 集成的 CVS 客户端

当 CVS 服务器安装配置结束以后，就可以通过 CVS 客户端访问 CVS 服务器中的资源。其中 CVS 客户端有很多中选择，例如 WinCVS、TortoiseCVS 等，使用这些工具都可以非常方便的访问 CVS 资源，在 Eclipse 中同样也内置了 CVS 客户端的功能。在下面的内容中就简单介绍如何使用 Eclipse 内置的 CVS 客户端来访问 CVS 资源。

(1) 在 Eclipse 的菜单栏中选择“Window”|“Show View”|“Other”就可以打开如图 2.26 所示的视图选择界面。

(2) 在如图 2.26 所示的视图选择界面中，选择“CVS Repositories”，然后单击“OK”按钮，就可以打开如图 2.27 所示的 CVS 视图。

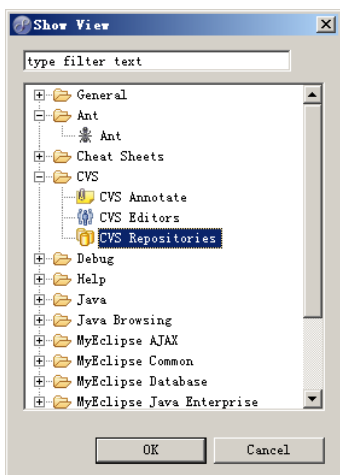


图 2.26 Eclipse 中的视图选择界面

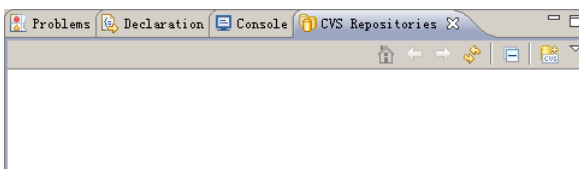


图 2.27 Eclipse 中的 CVS 视图

(3) 在如图 2.27 所示的界面中,单击鼠标右键,在弹出的菜单中选择“New”“Repository Location...”就可以进入如图 2.28 所示的连接 CVS 服务器的信息配置界面。

(4) 在如图 2.28 所示的界面中,Host 是 CVS 服务器所在的 IP 地址,在这里 CVSNT 的服务器就安装在本机,所以输入 localhost 即可,Repository Path 即 CVS 资源的访问名称,也就是在 CVSNT 中设置的资源名称,用户名和密码使用 CVSNT 服务器所在计算机的用户名和密码即可。其他设置使用默认值即可。设置完这些信息以后,单击“Finish”完成设置,就可以把这个 CVS 地址添加在如图 2.27 所示的 CVS 视图中。

(5) 在 CVS 视图中右键单击上面添加成功的 CVS 服务器路径,在弹出的菜单中选择“Refresh Branches”即可连接到 CVS 服务器,进入如图 2.29 所示的 CVS 资源选择界面。

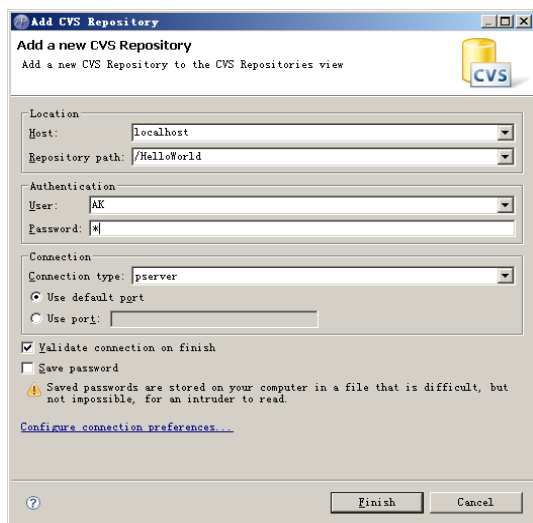


图 2.28 添加 CVS 服务器信息界面

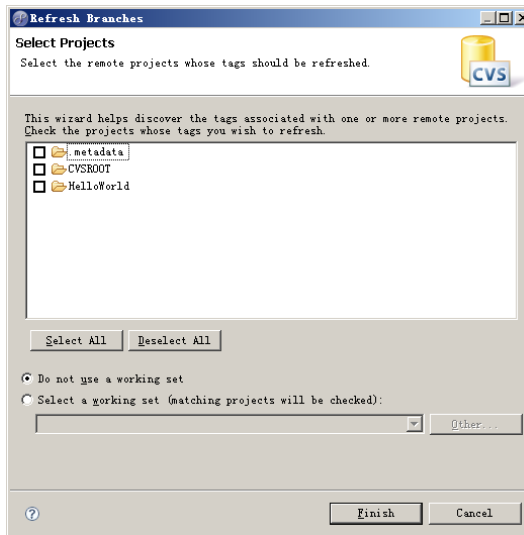


图 2.29 CVS 资源选择界面

(6) 在如图 2.29 所示的界面中,选择需要的资源,然后单击“Finish”按钮,就可以把 CVS 服务器上的这个资源导入到 Eclipse 中。

上面这些操作完成了把 CVS 服务器中的资源导入到 Eclipse 中的工作,然后既可以通过 Eclipse 来进行各种 CVS 的操作,例如更新到最新版本、提及自己的版本、恢复到历史版本等操作。这些工作都可以在 Eclipse 中完成。

2.6 小结

在本章内容中，对 Java 开发环境的搭建进行了大体的介绍，其中重点讲述了 JDK 的安装设置和 Eclipse 的基本使用方法，而且还提供了大量 Eclipse 中的快捷键，在本章最后的内容中，介绍了团队写作中源代码的版本控制问题，介绍了如何架设 CVS 服务器，如何使用 Eclipse 中内置的 CVS 客户端访问 CVS 服务器，通过本章内容的学习，读者可以对基本了解 Java 开发环境的基本知识，并且学会自己搭建设置这样的环境，为后面章节中的开发打下坚实的基础。这些技能都是在实际开发过程中必备的基础技能。读者需要熟练掌握。