


# 第 2 章

---

## 熟悉 Eclipse 开发工具

(  视频讲解：22 分钟 )

学习 Java 语言程序设计必须选择一个功能强大、使用简单、能够辅助程序设计的 IDE 集成开发工具，而 Eclipse 是目前最流行的 Java 语言开发工具，它以强大的代码辅助功能，帮助程序开发人员自动完成语法、补全文字、代码修正、API 提示等编码工作，节省了大量的程序开发时间。本章将简要介绍 Eclipse 开发工具，使读者能够初步了解 Eclipse 并使用它完成程序设计工作。

通过阅读本章，您可以：

- » 初步了解 Eclipse
- » 掌握 Eclipse 的下载和安装
- » 掌握 Eclipse 的汉化
- » 了解 Eclipse 界面
- » 掌握 Eclipse 的使用
- » 掌握调试 Eclipse 程序



## 2.1 熟悉 Eclipse

### 视频讲解：光盘\TM\lx\2\熟悉 Eclipse.exe

虽然使用“记事本”工具和 JDK 编译工具已经可以编写 Java 程序，但是在项目开发过程中必须使用大型的 IDE 集成开发工具来编写 Java 程序，这样可以避免编码错误、方便管理项目结构，而且使用 IDE 集成开发工具的代码辅助功能可以快速地录入程序代码。本节将介绍 Eclipse 开发工具，包括它的安装、配置与启动、菜单栏、工具栏和各种视图的作用等。

### 2.1.1 Eclipse 简介

Eclipse 是由 IBM 公司投资 4000 万美元开发的 IDE 集成开发工具。它是目前最流行的 Java 集成开发工具之一，基于 Java 语言编写，并且是开放源代码的、可扩展的（Integrated Development Environment, IDE）开发工具。另外，IBM 公司捐出 Eclipse 源代码，组建了 Eclipse 联盟，由该联盟负责这种工具的后续开发。Eclipse 为编程人员提供了一流的 Java 程序开发环境。它的平台体系结构是在插件概念的基础上构建的，插件是 Eclipse 平台最具特色的特征之一，也是其区别于其他开发工具的特征之一。学习完本章之后，读者将对 Eclipse 有一个初步的了解，为后面的深入学习做铺垫。

### 2.1.2 Eclipse 的下载与安装

(1) 打开浏览器，进入 Eclipse 官方主页，地址是 <http://www.eclipse.org>，如图 2.1 所示。



图 2.1 Eclipse 官方主页

(2) 单击 Download Eclipse 超链接, 在弹出的界面中根据读者的操作系统不同, 在 Eclipse IDE for Java Developers 标签右侧选择适当的版本, 如图 2.2 所示。



图 2.2 Eclipse 版本选择页面

(3) 弹出如图 2.3 所示的界面, 单击[China] Beijing Institute of Technology (http)超链接进行下载。



图 2.3 Eclipse 下载页面

(4) 将下载完成的压缩包进行解压缩, 就完成了 Eclipse 的安装。

### 2.1.3 Eclipse 汉化

为了方便不熟悉英语的用户使用 Eclipse, 下面讲解如何对 Eclipse 进行汉化。

(1) 打开浏览器，进入 Eclipse Babel 官方主页，地址是 <http://www.eclipse.org/babel/>，如图 2.4 所示。



图 2.4 Eclipse Babel 官方主页

(2) 单击页面左侧的 Downloads 超链接，弹出如图 2.5 所示的界面，复制 Babel Language Pack Update Site for Indigo 下方的 <http://download.eclipse.org/technology/babel/update-site/R0.9.0/indigo> 超链接。



图 2.5 Eclipse Babel 下载页面

(3) 启动 Eclipse，选择 Help 菜单，如图 2.6 所示。

(4) 选择 Install New Software 命令，弹出如图 2.7 所示的对话框。

(5) 单击 Add 按钮，显示 Add Repository（增加仓库）对话框，如图 2.8 所示。在 Name 文本框中输入“Eclipse Babel”，在 Location 文本框中输入刚刚复制的网址，单击 OK 按钮联网查找可用软件包。



图 2.6 Eclipse 的 Help 菜单



图 2.7 “安装新插件”对话框

(6) 弹出如图 2.9 所示的界面，选择 Babel Language Packs for eclipse 下的 Babel Language Pack for eclipse in Chinese(Simplified)，单击 Next 按钮。

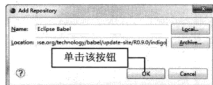


图 2.8 “增加仓库”对话框



图 2.9 “安装新插件”对话框

(7) 弹出如图 2.10 所示的界面，其中显示了插件的详细信息，包括插件名称、版本号 and ID。单

击 Next 按钮继续。

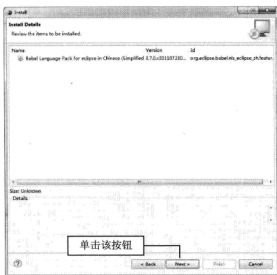


图 2.10 选择安装插件的详细信息

(8) 弹出如图 2.11 所示的安装协议界面，选中 I accept the terms of the license agreement 单选按钮同意协议，然后单击 Next 按钮开始安装。

(9) 弹出如图 2.12 所示的窗口，显示了插件的下载进度。



图 2.11 选择安装插件的详细信息

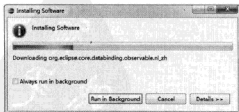


图 2.12 插件的下载进度

(10) 在下载完成后会询问是否安装该插件，如图 2.13 所示。

(11) 单击 OK 按钮确认安装，在安装完成后会要求重新启动 Eclipse，如图 2.14 所示。单击 Restart Now 按钮完成安装。

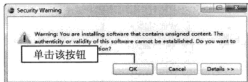


图 2.13 询问是否安装该插件

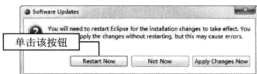


图 2.14 询问是否重启 Eclipse

## 2.1.4 Eclipse 工作台

现在已经配置好 Eclipse 的多国语言包，可以启动 Eclipse 了。在 Eclipse 的安装文件夹中运行 eclipse.exe 文件，Eclipse 在启动时会弹出“工作空间启动程序”对话框，该对话框用于设置 Eclipse 的工作空间（工作空间用于保存 Eclipse 建立的程序项目和相关设置）。本书的开发环境统一设置工作空间为 Eclipse 安装位置的 workspace 文件夹。在“工作空间启动程序”对话框的“工作空间”下拉列表框中输入“\workspace”，如图 2.15 所示，单击“确定”按钮，将启动 Eclipse。



### 注意

在每次启动 Eclipse 时，Eclipse 都会出现设置工作空间的对话框，通过选中“将此值用作默认值并且不再询问”复选框可以设置默认工作空间，这样 Eclipse 启动时就不会再询问工作空间的设置了。

Eclipse 首次启动时会显示欢迎界面，其中包含了 Eclipse 的概述、新增内容、样本、教程以及工作台的相关按钮，如图 2.16 所示。

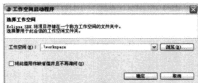


图 2.15 设置工作空间

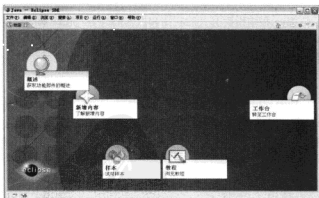


图 2.16 Eclipse 的欢迎界面

在 Eclipse 的欢迎界面中，单击“工作台”按钮或者关闭欢迎视图界面，将显示 Eclipse 的工作台。

它是程序人员开发程序的主要场所。Eclipse 还可以将各种插件无缝地集成到工作台中，也可以在工作台中开发各种插件。Eclipse 工作台主要包括标题栏、菜单栏、工具栏、编辑器、透视图和相关的视图等，如图 2.17 所示。下面将介绍 Eclipse 的菜单栏与工具栏，以及什么是透视图、视图，并介绍常用视图。

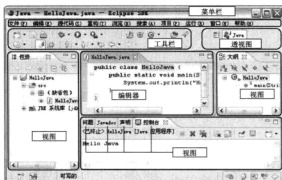


图 2.17 缩放后的 Eclipse 工作台

## 2.1.5 透视图与视图

透视图和视图是 Eclipse 中的概念，下面将分别介绍什么是透视图、什么是视图，以及它们在 Eclipse 中的作用。

### 1. 透视图

透视图是 Eclipse 工作台提供的附加组织层，也就是说，透视图起到一个组织的作用，它实现多个视图的布局和可用操作的集合，并为这个集合定义一个名称。例如，Eclipse 提供的 **Java** 透视图就组织了与 Java 程序设计有关的视图和操作的集合，而 **调试** 透视图负责组织与程序调试有关的视图和操作集。在 Eclipse 的 Java 开发环境中提供了几种常用的透视图，如 Java 透视图、资源透视图、调试透视图、小组同步透视图等。不同的透视图之间可以进行切换，但是同一时刻只能使用一个透视图。

### 2. 视图

视图多用于浏览信息的层次结构、显示活动编辑器的属性，如“控制台”视图用于显示程序运行时的输出信息和异常错误，而“包资源管理器”视图可以浏览项目的文件组织结构。视图可以单独出现，也可以与其他视图以选项卡样式叠加在一起，它们可以有自己独立的菜单和工具栏，并且可以通过拖曳随意改变布局位置。



**技巧**

在视图标题上单击鼠标右键，在弹出的快捷菜单中选择“已拆离”命令可以将视图从 Eclipse 工作台界面中分离，以单独窗口存在，同样的操作也可以将视图再次合并到工作台界面中。本书大部分视图都是从 Eclipse 中分离后才截取的界面效果。



## 2.1.6 菜单栏

Eclipse 的菜单栏包含了 Eclipse 的基本命令, 在使用不同的编辑器时, 还会动态地添加有关该编辑器的菜单。基本的菜单栏中除了常用的“文件”、“编辑”、“窗口”和“帮助”等菜单以外, 还提供了一些功能菜单, 如“源代码”和“重构”等, 如图 2.18 所示。

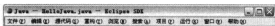


图 2.18 Eclipse 的菜单栏

每个菜单中都包含不同的命令, 这些命令用于完成最终的操作, 如文件的打开与保存、代码格式化、程序的运行与分步调试等。每个菜单所包含的命令如图 2.19 所示。

菜单中的命令虽多, 但并不是所有命令都经常使用。本节将介绍几个最常用的菜单及其命令, 至于其他不常用的菜单, 读者可以在日后程序开发过程中慢慢掌握。



图 2.19 Eclipse 菜单命令

### 1. “文件”菜单

“文件”菜单中包含“新建”、“保存”、“关闭”、“打印”、“切换工作空间”、“属性”等命令。菜单中包含的内容虽多，但也有常用和不常用的。在此将“文件”菜单中的常用命令都集中列在表 2.1 中，这样阅读和查询更方便。

表 2.1 “文件”菜单中的常用命令

图 标	命 令	说 明	快 捷 键
	新建	创建新项目、元素或资源	Shift+Alt+N
	打开文件	打开已经存在的文件	
	关闭	关闭当前编辑器	Ctrl+W
	全部关闭	关闭所有编辑器	Shift+Ctrl+W
	保存	保存当前编辑器的内容	Ctrl+S
	刷新	刷新所选元素的内容	F5
	切换工作空间	切换工作空间到其他位置。这将导致 Eclipse 重启	
	导入	打开“导入向导”对话框	
	导出	打开“导出向导”对话框	
	属性	打开所选元素的属性对话框	Alt+Enter

### 2. “编辑”菜单

“编辑”菜单用于辅助程序代码设计工作，除常用的“剪切”、“复制”、“粘贴”命令之外，还提供了“快速修正”、“将选择范围扩展到”和“内容辅助”等高级命令。常用命令如表 2.2 所示。

表 2.2 “编辑”菜单中的常用命令

图 标	命 令	说 明	快 捷 键
	将选择范围扩展到	将选择编辑内容的范围扩大到外层元素、下一个元素、上一个元素或者恢复上一次选择的元素	
	查找/替换	搜索编辑器中的内容片段，并根据需要替换为新的内容	Ctrl+F
	查找下一个	搜索当前所选内容下一次出现的地方	Ctrl+K
	查找上一个	搜索当前所选内容上一次出现的地方	Shift+Ctrl+K
	添加书签	当前光标所在行添加书签	
	添加任务	当前光标所在行添加任务	
	灵活插入方式	切换插入方式。当禁用灵活插入方式时，将禁用自动缩进、添加右方括号等辅助功能	Shift+Ctrl+Insert
	内容辅助	在当前光标位置打开内容辅助对话框	
	文字补全	补全当前编辑器中正在输入的文字	Ctrl+Alt+/
	快速修正	如果光标位于问题代码附近，则打开一个解决方案对话框	Ctrl+I

### 3. “源代码”菜单

“源代码”菜单中包含的命令都是和代码编写相关的一些命令，主要用于辅助编程。常用命令如表 2.3 所示。

表 2.3 “源代码”菜单中的常用命令

图 标	命 令	说 明	快 捷 键
	切换注释	注释或取消注释当前选择的所有行	Ctrl+/ /
	添加块注释	在当前选择的多行代码周围添加块注释	Shift+Ctrl+/ /
	除去块注释	从当前选择的多行代码中除去块注释	Shift+Ctrl+\
	更正缩进	更正当前选择的代码行的缩进	Ctrl+I
	格式	使用代码格式化程序来格式化当前 Java 代码	Shift+Ctrl+F
	组织导入	导入当前类所使用的类包	Shift+Ctrl+O
	覆盖/实现方法	使用向导覆盖父类或实现接口中的方法	
	生成 Getter 和 Setter	使用向导创建成员变量的 getXXX()/setXXX()方法	
	生成 hashCode()和 equals()	打开“生成 hashCode()和 equals()”对话框	
	使用字段生成构造函数	添加构造函数，这些构造函数初始化当前选择的类型的字段，可用于类型、字段或类型中的文本选择	
	从超类中生成构造函数	对于当前所选择的类型，按照超类中的定义来添加构造函数	
	包围方式	使用代码模板包围所选语句	Shift+Alt+Z
	外部化字符串	打开“将字符串外部化”向导，此向导允许通过使用语句访问属性文件来替换代码中的所有字符串	

#### 4. “重构”菜单

“重构”菜单是 Eclipse 最关键的菜单，主要包括对项目重构的相关命令，是读者最应该掌握的。常用命令如表 2.4 所示。

表 2.4 “重构”菜单中的常用命令

图 标	命 令	说 明	快 捷 键
	重命名	重命名所选择的 Java 元素	Shift+Alt+R
	移动	移动所选择的 Java 元素	Shift+Alt+V
	抽取方法	创建一个包含当前所选择的语句或表达式的新方法，并相关地引用	Shift+Alt+M
	抽取局部变量	创建为当前所选择的表达式指定的新变量，并将选择替换为对新变量的引用	Shift+Alt+L
	抽取常量	从所选表达式创建静态终态字段并替换字段引用，并且可以选择重写同一表达式的其他出现位置	
	内联	直接插入局部变量、方法或常量	Shift+Alt+I
	将匿名类转换为嵌套类	将匿名内部类转换为成员类	
	将成员类型转换为顶级	为所选成员类型创建新的 Java 编译单元，并根据需要更新所有引用	
	将局部变量转换为字段	如果该变量是在创建时初始化的，则此操作将把初始化移至新字段的声明或类的构造函数	
	抽取超类	从一组同类型中抽取公共超类	
	抽取接口	根据当前类的方法创建接口，并使该类实现这个接口	
	包括字段	将对变量的所有引用替换为 getXXX()/setXXX()方法	
	历史记录	浏览工作空间重构历史记录，并提供用于从重构历史记录中删除重构的选项	

## 5. “项目”菜单

“项目”菜单主要用于管理 Eclipse 中的项目，可以实现项目的打开与关闭、自动构建等操作。常用命令如表 2.5 所示。








表 2.5 “项目”菜单中的常用命令

图 标	命 令	说 明	快 捷 键
	打开项目	打开选择的已关闭的项目	
	关闭项目	关闭当前所选择的项目	
	全部构建	在工作空间中构建所有项目	Ctrl+B
	构建项目	构建当前所选择的项目	
	构建工作集	构建当前工作集中所有项目	
	清理	选择要清理的项目	
	自动构建	选择此项，在保存所有已修改的文件时都将自动重建它们	
	生成 Javadoc	对当前选择的项目打开“生成 Javadoc”向导	
	属性	查看选择项目的属性	

## 6. “运行”菜单

“运行”菜单中列出了与程序运行和调试相关的各种操作，同时还能保存运行和调试的记录。常用命令如表 2.6 所示。

表 2.6 “运行”菜单中的常用命令

图 标	命 令	说 明	快 捷 键
	运行上次启动	快速重复最近的程序启动	Ctrl+F11
	调试上次启动	快速重复最近的程序调试	F11
	运行历史记录	显示以运行方式启动的启动配置的最近历史记录子菜单	
	运行方式	显示当前程序所支持的运行方式	
	运行	打开“启动配置”对话框来管理运行方式启动配置	
	调试历史记录	显示最近程序调试的历史记录	
	调试方式	显示当前程序所支持的调试方式	
	调试	打开“调试配置”对话框来管理调试配置	
	切换运行断点	添加或除去 Java 编辑器中当前行的断点	Shift+Ctrl+B
	切换观察点	添加或除去当前 Java 变量的观察点	
	跳过所有断点	跳过已经设定的所有断点	
	除去所有断点	从“断点”视图中去除所有断点	
	添加 Java 异常断点	添加异常断点	

## 7. “窗口”菜单

“窗口”菜单主要用于显示、隐藏或处理 Eclipse 中的各种视图、透视图和操作。常用命令如表 2.7 所示。

表 2.7 “窗口”菜单中的常用命令

图 标	命 令	说 明	快 捷 键
	打开透视图	在 Eclipse 中打开新的透视图	
	显示视图	在当前透视图显示指定的视图	
	将透视图另存为	将当前透视图布局保存为一个新的透视图	
	复位透视图	将当前透视图的布局复位到上次保存的状态	
	关闭透视图	关闭当前透视图	
	关闭所有透视图	关闭 Eclipse 中所有打开的透视图	
	工作集	有关工作集的管理操作	
	首选项	打开 Eclipse 的首选项对话框, 对 Eclipse 进行系统的设置	

### 2.1.7 工具栏

Eclipse 的工具栏位于菜单栏的下方, 这与大多数软件的布局格式相同。工具栏中的按钮都是菜单命令对应的快捷图标, 在打开不同的编辑器时还会动态地添加与编辑器相关的新工具栏按钮。另外, 除了菜单栏下面的工具栏称为主工具栏之外, Eclipse 中还有视图工具栏、透视图工具栏和快速视图工具栏等多种工具栏。

#### 1. 主工具栏

主工具栏就是位于 Eclipse 菜单栏下方的工具栏, 其内容将根据不同的透视图和不同类型的编辑器显示相关工具按钮, 如图 2.20 所示



图 2.20 Eclipse 主工具栏

#### 2. 视图工具栏

Eclipse 界面中包含多种视图, 这些视图各有不同的用途, 根据视图的功能需求会在视图的标题栏位置添加相应的视图工具栏。例如, “控制台”视图用于输出程序运行中的输出结果和运行时异常信息, 其工具栏如图 2.21 所示。

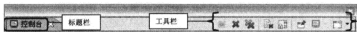


图 2.21 “控制台”视图的标题栏和工具栏

#### 3. 透视图工具栏

透视图工具栏主要包括切换已经打开的不同透视图的缩略按钮、选择打开其他视图的按钮。在相应的工具按钮上单击鼠标右键会弹出透视图的管理菜单, 实现透视图的定制、关闭、复位、布局位置、是否显示文本等操作, 如图 2.22 所示。

#### 4. 快速视图工具栏

快速视图工具栏包含了一组快速视图的图标按钮，该工具栏中的视图会以图标按钮形式隐藏，当单击指定工具按钮时显示指定的视图，而该视图失去焦点时会自动隐藏到快速工具栏。默认情况下，快速视图工具栏出现在工作台的左下角，如图 2.23 所示。



图 2.22 透视图工具栏



图 2.23 Eclipse 中的快速视图工具栏

### 2.1.8 “包资源管理器”视图

该视图用于浏览项目结构中的 Java 元素，包括包、类、类库的引用等，但最主要的用途还是操作项目中的源代码文件。“包资源管理器”视图的界面如图 2.24 所示。

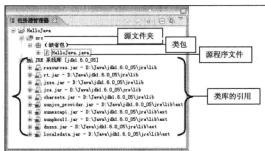


图 2.24 拆分后的“包资源管理器”视图

### 2.1.9 “控制台”视图

“控制台”视图用于显示程序运行时的输出结果和运行时的异常信息（Runtime Exception），在学习 Swing 程序设计之前，必须使用控制台实现与程序的交互，例如为方便某个方法的调试，在方法执行前后分别输出“方法开始”和“方法结束”信息。“控制台”视图的界面如图 2.25 所示。

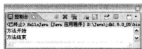


图 2.25 拆分后的“控制台”视图

## 2.2 使用 Eclipse

#### 视频讲解：光盘\TM\lx\2\使用 Eclipse.exe

现在读者对 Eclipse 工具应该有个大体的认识了，本节将介绍如何使用 Eclipse 完成 HelloJava 程序

的编写和运行。

## 2.2.1 创建 Java 项目

在 Eclipse 中编写程序，必须先创建项目。Eclipse 中有很多种项目，其中 Java 项目用于管理和编写 Java 程序。创建该项目的步骤如下：

(1) 选择“文件”/“新建”/“项目”命令，打开“新建项目”对话框。该对话框包含创建项目的向导，在向导中选择“Java 项目”节点，单击“下一步”按钮。

(2) 在弹出的“新建 Java 项目”对话框的“项目名”文本框中输入“HelloJava”，在“项目布局”栏中选中“创建单独的源文件夹和输出文件夹”单选按钮，如图 2.26 所示，单击“完成”按钮。

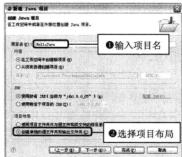


图 2.26 “新建 Java 项目”对话框

## 2.2.2 创建 Java 类文件

创建 Java 类文件时，自动打开 Java 编辑器。创建 Java 类文件可以通过“新建 Java 类”向导来完成。在 Eclipse 菜单栏中选择“文件”/“新建”/“类”命令，打开“新建 Java 类”向导对话框，如图 2.27 所示。



图 2.27 “新建 Java 类”向导对话框

使用该向导对话框创建 Java 类的步骤如下：

(1) 在“源文件夹”文本框中输入项目源程序文件夹的位置。通常向导会自动填充该文本框，没有特殊情况，不需要修改。

(2) 在“包”文本框中输入类文件想使用的包名，这里暂时默认为空，不输入任何信息，这样就会使用 Java 工程的“缺省包”。

(3) 在“名称”文本框中输入新建类的名称，如 HelloJava。



**注意**

虽然 HelloJava 类名与 Java 项目同名，但是它们分别代表类文件和 Java 项目，读者要注意区分它们的含义。

(4) 选中 `public static void main(String[] args)` 复选框，向导在创建类文件时，会自动为该类添加 `main()` 方法，使该类成为可以运行的主类。

## 2.2.3 使用编辑器编写程序代码

编辑器总是位于 Eclipse 工作台的中间区域，该区域可以重叠放置多个编辑器。编辑器的类型可以不同，但是它们的主要功能都是完成 Java 程序、XML 配置等代码编写或可视化设计工作。本节将向读者介绍如何使用 Java 编辑器和其他代码辅助功能快速编写 Java 程序。

### 1. 打开 Java 编辑器

在使用向导创建 Java 类文件之后，会自动打开 Java 编辑器编辑新创建的 Java 类文件。除此之外，打开 Java 编辑器最常用的方法是在“包资源管理器”视图中双击 Java 源文件或者右击 Java 源文件，在弹出的快捷菜单中选择“打开方式”/“Java 编辑器”命令。Java 编辑器的界面如图 2.28 所示。

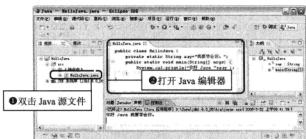


图 2.28 Java 编辑器的界面

从图 2.28 中可以看到，Java 编辑器以各种样式和不同的颜色突出显示 Java 语法。这些突出显示的语法如下：

- ☒ 程序代码注释
- ☒ Javadoc 注释
- ☒ Java 关键字



- ☒ 变量
- ☒ 字符串

## 技巧

在 Java 编辑器左侧单击鼠标右键，在弹出的快捷菜单中选择“显示行号”命令，可以开启 Java 编辑器显示行号的功能。

## 2. 编写 Java 代码

Eclipse 的强大之处并不在于编辑器能突出显示 Java 语法，而是它强大的代码辅助功能。在编写 Java 程序代码时，可以使用 Ctrl+Alt+/ 快捷键自动补全 Java 关键字，也可以使用 Alt+/ 快捷键启动 Eclipse 代码辅助菜单。使用 Eclipse 的代码辅助功能编写 HelloJava 类的步骤如下：

在使用向导创建 HelloJava 类之后，向导会自动构建 HelloJava 类结构的部分代码，并且已经建立了 main() 方法，程序开发人员需要做的就是将代码补全，为程序添加相应的业务逻辑。本程序的完整代码如图 2.29 所示。

在 HelloJava 程序代码中，第 1、3、5、6 行是由向导创建的，完成这个程序只要编写第 2 行和第 4 行代码即可。

首先来看一下第 2 行代码。它包括 private、static、String 3 个关键字，这 3 个关键字如果在“记事本”中手动输入可能不会费多长时间，但是无法避免出现输入错误的情况，例如将 private 关键字输入为 privat，缺少了字母 e，这个错误可能在程序编译时才会被发现。这只是简单的关键字，如果是名称更长、更复杂的关键字，就更容易出现错误。在 Eclipse 的 Java 编辑器中可以输入关键字的部分字母，然后使用 Ctrl+Alt+/ 快捷键自动补全 Java 关键字，如图 2.30 所示。

```
1 public class HelloJava {
2     private static String say="我要学编程。";
3     public static void main(String[] args) {
4         System.out.println("你好 Java "say);
5     }
6 }
```

图 2.29 HelloJava 程序代码

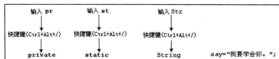


图 2.30 使用快捷键补全关键字

其次是第 4 行的程序代码。它使用 System.out.println() 方法输出文字信息到控制台，这是程序开发时最常使用的方法之一。当输入“.”操作符时，编辑器会自动弹出代码辅助菜单，也可以在输入部分文字之后使用 Alt+/ 快捷键调出代码辅助菜单，完成关键语法的输入，如图 2.31 所示。

## 注意

System.out.println() 方法在 Java 编辑器中可以通过输入“syso”和按 Alt+/ 键完成快速输入。

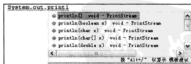



图 2.31 代码辅助菜单



### 技巧

将光标移动到 Java 编辑器的错误代码位置，按 **Ctrl+I** 键可以激活代码修正菜单，进而选择菜单中合适的修正方法。

## 2.2.4 运行 Java 程序

HelloJava 类包含 `main()` 主方法，它是一个可以运行的主类。在 Eclipse 中运行 HelloJava 程序，可以在“包资源管理器”视图中右击 HelloJava 文件，在弹出的快捷菜单中选择“运行方式”/“ Java 应用程序”命令。运行结果如图 2.32 所示。

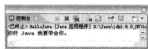


图 2.32 HelloJava 程序在控制台的输出结果

## 2.3 程序调试



### 视频讲解：光盘\TM1x\2\程序调试.exe

读者在程序开发过程中会不断体会到程序调试的重要性。为验证 Java 单元的运行状况，经常在某个方法调用的开始和结束位置分别使用 `System.out.println()` 方法输出状态信息，根据这些信息判断程序执行状况，这是非常古老的程序调试方法，而且经常导致程序代码混乱（导出都是 `System.out.println()` 方法）。

本节将简单介绍 Eclipse 内置的 Java 调试器的使用方法，使用该调试器可以设置程序的断点、实现程序单步执行、在调试过程中查看变量和表达式的值等调试操作，这样可以避免在程序中编写大量的 `System.out.println()` 方法输出调试信息。

使用 Eclipse 的 Java 调试器需要设置程序断点，然后使用单步调试分别执行程序代码的每一行。

### 1. 断点

设置断点是程序调试中必不可少的有效手段，Java 调试器每次遇到程序断点时都会将当前线程挂起，即暂停当前程序的运行。

可以在 Java 编辑器中显示代码行号的位置双击添加或删除当前行的断点，或者在当前行号的位置单击鼠标右键，在弹出的快捷菜单中选择“切换断点”命令实现断点的添加与删除，如图 2.33 所示。

### 2. 以调试方式运行 Java 程序

在 Eclipse 中调试 HelloJava 程序，可以在“包资源管理器”视图中右击 HelloJava 文件，在弹出的快捷菜单中选择“调试方式”/“Java 应用程序”命令。图 2.33 中在第 4 行代码设置了断点，调试器将在该断点处挂起当前线程，使程序暂停，如图 2.34 所示。



图 2.33 Java 编辑器中的断点

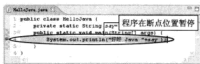


图 2.34 程序执行到断点后暂停

### 3. 程序调试

程序执行到断点被暂停后，可以通过“调试”视图工具栏上的按钮，执行相应的调试操作，如运行、停止等。“调试”视图如图 2.35 所示。

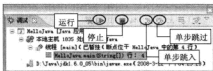




图 2.35 “调试”视图

(1) 单步跳过。在“调试”视图的工具栏中单击  按钮或按 F6 键，将执行单步跳过操作，即运行单独的一行程序代码，但是不进入调用方法的内部，然后跳到下一个可执行点并暂停线程。



**说明**

不停地执行单步跳过操作，会每次一行执行程序代码，直到程序结束或等待用户操作。

(2) 单步跳入。在“调试”视图的工具栏中单击  按钮或按 F5 键，执行该操作将跳入调用方法或者对象的内部单步执行程序并暂停线程。

## 2.4 本章小结

本章主要介绍了 Eclipse 开发工具，包括它的配置与启动、菜单栏、工具栏，以及常用的“包资源管理器”和“控制台”视图；另外还介绍了 Eclipse 的使用方法，包括创建 Java 项目、创建 Java 类、编写程序代码和运行程序；最后还讲解了 Java 程序调试器的简单使用方法。

[illegible]