

## 第一篇

# NetBeans 基础篇

第 1 章 NetBeans 工具介绍

第 2 章 编辑与重构源代码

第 3 章 调试 Java 应用程序

## 第 1 章



### 内容摘要 | Abstract

NetBeans 是一个集成、开源、跨平台的可视化开发工具。提供了一个纯 Java (TM) 的技术开发环境窗口,可以在 Windows、Linux 和 Solaris(TM)等多个操作系统下运行。在 NetBeans 的环境中,可以进行编写、编辑、调试和部署 Java 程序。同时 NetBeans 具有很好的扩展性,利用第三方插件能够支持常见的语言技术,例如 C、C++、XML、HTML 和 Java Web 技术等,几乎是任何人们能想到的技术。

本章介绍 NetBeans 的基础知识,例如 NetBeans 的产生和发展、下载和安装时常用窗口的作用和用法、源代码编辑器的使用等;同时重点介绍如何在 NetBeans 中创建项目、项目中常见的操作以及如何配置、运行项目等。



### 学习目标 | Objective

- 了解 NetBeans 的产生和发展
- 掌握下载和安装 NetBeans 的方法
- 了解 NetBeans 常用窗口的作用和用法
- 熟练掌握 NetBeans 源代码编辑器的使用
- 掌握 NetBeans 中常见的的项目操作
- 理解配置和运行项目

## 1.1 NetBeans 概述和安装

NetBeans 是由 Sun 公司 (Sun Microsystems) 建立的开源的软件开发工具,现在最新的版本是 Netbeans 6.1。在 NetBeans Platform 平台中,应用软件是用一系列的软件组件 (modular software components) 构建出来。而这些组件是一个 jar 文档 (Java archive file),有赖于组件化带来的好处,用组件来构建的应用程序只要加上新的组件就能进一步扩充。由于组件可以独立地进行开发,所以由 NetBeans 平台开发出来的应用程序就能利用第三方软件,非常容易及有效率地进行扩充。

### 1.1.1 NetBeans 发展介绍

NetBeans IDE 作为一个教学项目于 1996 年启动,当初的名称并非 NetBeans,而是叫做

Xelfi。Xelfi 项目的目标正是开发一个类似于 Delphi 的 Java IDE，而且为了证明 Java 的无所不能，这个 IDE 本身的开发语言毫无悬念地选择了 Java 语言。很快，Xelfi 第一个预览版本在 1997 年发布，虽然是个不成熟的产品，却给当时的 Java IDE 领域带来一股清新的空气。要知道，那时还是字符界面 IDE 的王国，尤其是对 Java 语言来说，Xelfi 是第一个图形化的 Java IDE，其意义重大而深远。



Xelfi 最初商业目标是开发具有网络功能的 JavaBeans 组件。Xelfi 的架构设计者 JardaTulach 给这种组件命名为 NetBeans，IDE 的目标就是协助开发 NetBeans——开发为网络使用的 Java Beans。

第一个正式采用 NetBeans 这个名称的是 NetBeans 2.0 版本。1999 年春，NetBeans Developer X2 发布，支持 Swing 技术。不过，这时 NetBeans 一直受到性能问题的困扰，主要是因为 Java 平台本身性能不高。直到 1999 年秋，当 Sun 公司发布 JDK 1.3，大幅度优化了 Java 平台效率后，NetBeans Developer X2 才成为 Java IDE 一个可以考虑的选择。这年夏天，NetBeans 开发小组对 NetBeans Developer X2 结构进行了重新规划。这个更模块化的 NetBeans 结构，成为此后 NetBeans 发展的基础。

2000 年 6 月，NetBeans 迎来了其发展历程上的又一重大机遇，即 NetBeans 开源。在客观上，开源将 NetBeans 带入了一个快速发展的阶段。NetBeans 是 Sun 公司资助的第一个开源项目，Sun 公司负责开源组织架构的管理和维护，并且专门为 NetBeans 建立了一个网站，那就是 NetBeans.org。开源后 NetBeans 势头迅猛，版本更新速度加快，越来越多的工具包被开发出来并加入 NetBeans 中，而 NetBeans 一词也逐渐成为出版物中出现频度最高的 Java 开发工具之一。Sun 公司的开源决策帮助 NetBeans 取得了出乎意料的成功。



NetBeans 过人之处，是时刻紧跟 Java 标准和版本。

当 JDK 1.5 发布时，NetBeans 发布了 5.0 版本来支持它，当 Java EE 5 发布时，NetBeans 又发布了 5.5 版本呼应。当 JDK 1.6.X 发布时，NetBeans 发布了 6.X 版本来支持它。

NetBeans 6.X 是在 NetBeans 5.5 之后的一个新版本，在继承了 NetBeans 5.5 优秀功能基础上，NetBeans 6.0 又提供了如下功能及改进。

- ❑ **功能强大的代码编辑器** 编辑器支持缩进、自动补全和高亮显示。代码编辑器可以自动分析代码、自动匹配单词和括号、标注代码错误、显示提示和 Javadoc。还提供了被定制化或者横向纵向切分，同时提供集成的代码重构、调试和 JUnit 测试。
- ❑ **增强的图形用户界面编辑器** NetBeans 6.0 可以通过拖曳设计基于 Swing 组件的图形用户界面，只需从组件面板中选择组件并拖曳到 GUI Builder 里，再双击就可以直接编辑 JLabel、JButton 和 JTextField 的文字属性。
- ❑ **全面支持 Java 企业版应用程序** NetBeans 6.0 全面支持 GlassFish、WebLogic 9.0、JBoss 5 以及 Tomcat 服务器等，同时支持 J2EE 和 Java EE 5，包括 JAX-RPC 和 JSR 109 标准。NetBeans 6.0 还提供了对 Java EE API 的代码自动完成和 Javadoc 功能支持。
- ❑ **Web 程序开发** NetBeans 6.0 支持 JSF、JSP、Struts、Ajax 和 JSTL 等技术，同时提供

编辑、部署描述符的可视化编辑器和调试 Web 应用的 HTTP 监视器。

- ❑ **协同开发** 使用 NetBeans Developer Collaboration (可以从更新中心下载), 开发人员可以通过网络实时共享整个项目和文件。这样就可以让其他同事在远程进行修改。协同开发还可以检查其他伙伴的工作并发送消息。而这消息可以是普通文本、或者有高亮支持的 XML、HTML 和 Java 代码。
- ❑ **Ruby 和 Rails 支持** 从 NetBeans 6.0 开始加入对 Ruby 开发的支持, 提供了功能强大的 Ruby 编辑器 (具有代码自动完成功能) 和调试器, 并全面支持 Rails。
- ❑ **对版本控制 CVS 和 Subversion 支持** 版本控制已经被紧密地集成到 NetBeans 开发环境的工作流当中。开发人员可以选择使用内建的 CVS, 或者从更新中心下载 Subversion 模块。
- ❑ **可视化的手机程序开发** NetBeans 6.0 提供了可视化的手机程序开发, 利用其提供的可视化设计器, 可以通过拖曳方式在手机界面上添加组件。

### 1.1.2 下载 NetBeans IDE

NetBeans 是一个可视化集成开发工具, 安装时会需要相应的硬盘空间, 运行时需要相应的内存空间, 所以在使用 NetBeans 开发工具之前, 需要了解 NetBeans 工具对系统的最低要求, 如表 1-1 所示。

表 1-1 NetBeans 对系统最低要求

操作系统	处理器	内存	硬盘空间
Microsoft Windows 操作系统	500MHz Intel Pentium III 工作站 或与之等效的处理器	512MB	850MB 可用硬盘空间
Linux 操作系统	500MHz Intel Pentium III 工作站 或与之等效的处理器	512MB	450MB 可用硬盘空间
Solaris OS (SPARC)	UltraSPARC II 450MHz	512MB	450MB 可用硬盘空间
Solaris OS (x86/x64 Platform Edition)	AMD Opteron 100 系列 1.8GHz	512MB	450MB 可用硬盘空间
Macintosh OS x 操作系统	PowerPC G5	512MB	450MB 可用硬盘空间



提示

下载 NetBeans 开发工具非常简单, 可以到其官方网站 <http://www.netbeans.org> 或者 Sun 中国技术社区 <http://developers.sun.com.cn/> 下载。对于不习惯访问英文网站的读者, 可以到 Sun 中国技术社区下载, 并且 Sun 中国技术社区还提供了一些中文学习资料和参考文档。

目前 NetBeans 最稳定的版本就是 NetBeans 6.1, 所以本书将以此版本为例介绍 NetBeans 开发工具的各个知识点。图 1-1 为 NetBeans 下载操作界面, 根据需要可以在【语言】下拉列表中选择【简体中文】选项, 在【平台】下拉列表中选择 Windows 2000/XP/Vista 选项。同时用户可以根据自己需要, 下载有不同开发包组成的 NetBeans 开发工具包。此处单击最右侧的【下载】超链接按钮, 下载具有所有开发包的 NetBeans 工具。

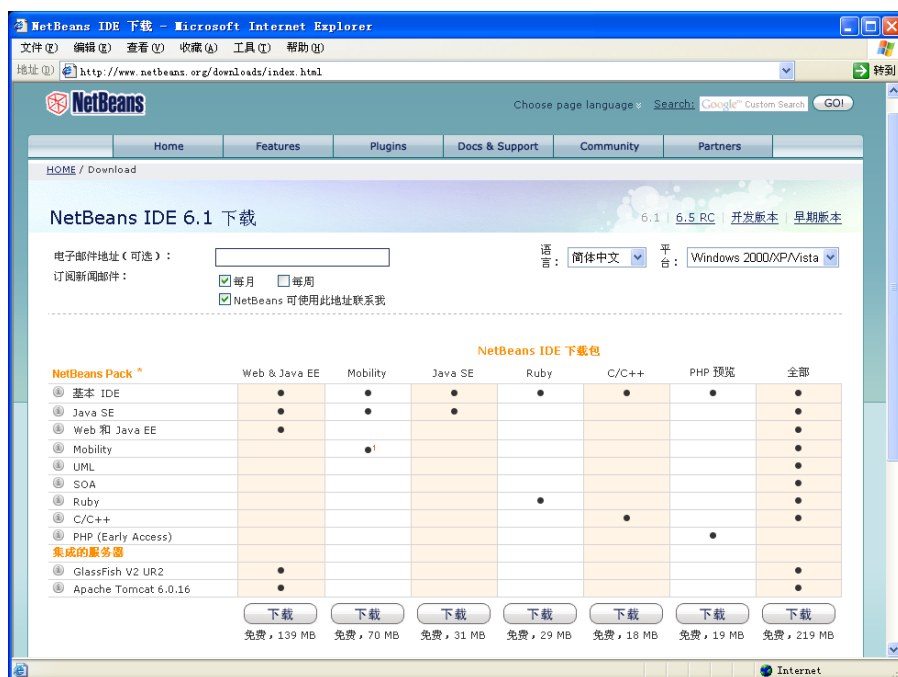


图 1-1 NetBeans 官方网站下载

图 1-1 中最左侧一列表示 NetBeans 6.1 各个开发工具包, 各个开发包主要功能介绍如表 1-2 所示。

表 1-2 开发包说明

名称	说明
基本 IDE	NetBeans 集成开发环境最基本功能
Java SE	用于开发 Java SE 标准程序的开发包
Web 和 Java EE	提供开发 Web 应用程序、可视化的 JSF 应用程序, 以及 Java EE 企业级应用程序的功能。可以通过拖放控件的方式, 以可视化的方式构造 Web 应用程序。在 NetBeans 集成开发环境中, 添加了以可视化快速生成基于标准的 Web 应用程序所需的所有功能, 其中包括 Ajax 和 JSF 组件支持
Mobility	可以开发基于 CLDC/MIDP 或者 CDC 的手机应用程序, 使用可视化工具快速生成手机程序的图形界面
UML	提供了创建 UML 项目的功能, 可以进行 UML 双向建模
SOA	加入了开发专业 SOA 应用的所有功能。开发人员可以用 XML、BPEL 和 Java Web 服务编写、测试和调试 SOA 应用
Ruby	提供了开发 Ruby 应用程序的功能, 提供了功能强大的 Ruby 编辑器、调试器, 并全面支持 Rails
C/C++	提供了在 NetBeans 开发环境中创建 C/C++ 应用程序的能力。可以为很多不同的平台编写 C/C++ 的代码, 包括 Microsoft Windows、Linux 和 Solaris
PHP	提供了用于开发 PHP Web 应用程序的开发包

在图 1-1 的【下载】按钮下方显示了安装不同 NetBeans 开发包所需要的空间。

### 1.1.3 安装 NetBeans IDE

NetBeans 可以在多种不同操作系统上运行，如 Linux、Solaris OS2 等，这里以 Windows 操作系统为例进行安装介绍。从官方网站下载 NetBeans 中文简体版后，就可以直接安装了。双击 NetBeans 安装文件，打开如图 1-2 所示的窗口。

在图 1-2 中，单击【定制】按钮，在打开的对话框中启用所有的复选框，安装所有软件包。然后单击【确定】按钮，返回到图 1-2 所示的窗口。



在图 1-2 中，读者根据需要也可以直接单击【下一步】按钮，选取默认安装。如果读者想取消其中一个软件包的安装，也可以单击图 1-2 中的【定制】按钮，在打开的对话框中禁用要取消安装的软件包复选框。

然后单击【下一步】按钮后，在打开窗口中，启用【我接受许可证协议中的条款】复选框。设置完成后，单击【下一步】按钮，打开如图 1-3 所示的设置安装路径窗口。

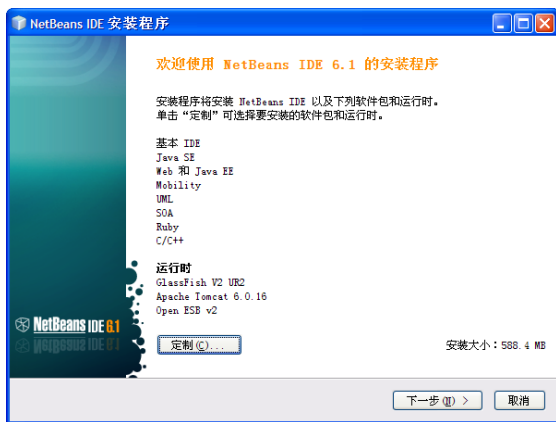


图 1-2 显示 NetBeans 安装欢迎信息

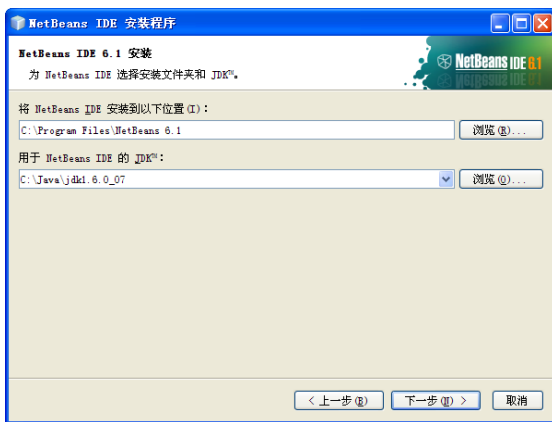


图 1-3 设置安装路径

在图 1-3 中，读者可以设置 NetBeans 的安装路径，这里选择默认；当然，也可以单击第一个【浏览】按钮进行设置。如果操作系统中安装多个 JDK 软件工具包，那么可以单击第二个【浏览】按钮，选择 NetBeans 要运行在哪个 JDK 之上。



NetBeans 运行需要 JDK 5.0 或更高版本。一般情况下，安装 NetBeans 之前，需要先安装一个 JDK 开发工具包，最好使用 JDK 6.0 或以上版本。

在图 1-3 窗口中路径选择完毕后，单击【下一步】按钮，会打开 GlassFish V2 UR2 安装窗口，如图 1-4 所示。

在图 1-4 中，单击第一个【浏览】按钮，选择 GlassFish V2 服务器安装路径。在【用于 GlassFish 应用服务器的 JDK】下拉列表中选择 GlassFish 应用服务器使用的 JDK 版本。【管理员用户名】是服务器管理员账户，默认值为 admin。



图 1-4 GlassFish V2 安装对话框



【HTTP 端口】指定使用 HTTP 协议访问服务器时使用的端口号，默认值为 8080。  
【HTTPS 端口】指定使用 HTTPS 协议访问服务器时使用的端口号，默认值为 8181。  
【管理端口】指定使用 GlassFish 管理控制台对服务器进行管理时使用的端口号，默认值为 4848。

在图 1-4 中，接受所有默认值，单击【下一步】按钮，打开软件包安装信息的窗口。在该窗口中单击【安装】按钮开始安装。安装完毕后，打开安装成功窗口。

NetBeans 启动完毕后，显示 NetBeans 操作窗口，如图 1-5 所示。

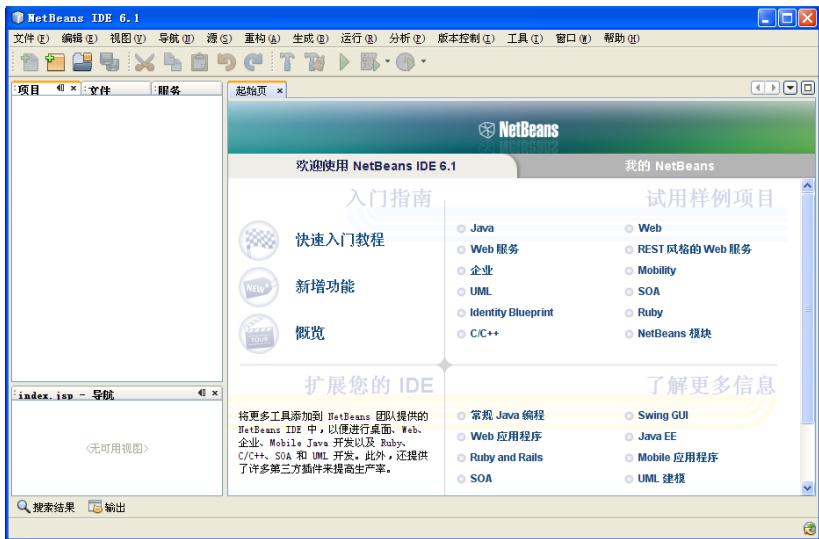


图 1-5 NetBeans 工作主界面

在 NetBeans 主界面中，最上面的是菜单栏，菜单栏下面是工具栏，最下面是 NetBeans 的各个窗口。其中，右下角是欢迎窗口，在第一次运行 NetBeans 时都会默认打开欢迎窗口。在欢迎窗口中，开发人员可以快速打开项目，访问 NetBeans 的快速入门教程、新增功能和概览等。

NetBeans 相比较 Eclipse 开发工具来说，最大优点是汉化程度非常高，其帮助文档都是使用中文编写，这对于初学者而言是一个极大的帮助。单击图 1-5 中的【快速入门教程】超链接，可以在线链接 NetBeans 学习文档，如图 1-6 所示。



图 1-6 NetBeans 快速入门教程

通过 NetBeans 提供的在线入门教程，初学者可以很快掌握 NetBeans 基础知识。同样，也可以在图 1-5 中的菜单栏上执行【帮助】|【帮助内容】命令，打开如图 1-7 所示的窗口。



图 1-7 NetBeans 帮助文档

通过图 1-7 提供的知识，可以快速掌握 NetBeans 开发工具的各个应用。NetBeans 主界面中各个菜单项和工具栏上的按钮，与其他开发工具的菜单项和按钮基本相同，这里就不再介绍了，读者可以使用帮助文档学习。

## 1.2 NetBeans 常用窗口

NetBeans 集成开发环境中比较重要的就是各种不同功能的窗口，如【项目】窗口、【服务】窗口、【文件】窗口、【输出】窗口和【导航】窗口等。通过这些窗口可以实现不同的功

能操作，本节将主要介绍这些常用窗口，对于其他窗口会在后面相关章节中介绍。

### 1. 【项目】窗口

【项目】窗口列出当前打开的所有项目，是项目源的入口点。展开某个项目会看到项目中使用项目内容的逻辑实体，如图 1-8 所示。该图是项目 JavaApplication1 的【项目】窗口。

【项目】窗口中所包含的各个选项，参见表 1-3。

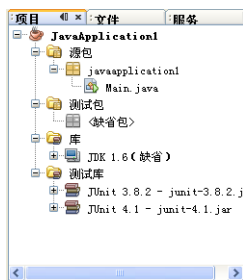


图 1-8 【项目】窗口

表 1-3 【项目】窗口中节点选项表

名称	说明
源包	包括源代码中的源代码文件，双击某个源文件节点可以在源代码编辑器中打开该文件
测试包	编写的单元测试代码将会包含在这个包中
库	项目需要使用的库文件
测试库	编写的测试程序使用的测试库

默认情况下，打开 NetBeans 时，将在集成开发环境的左上角打开【项目】窗口，也可以通过执行菜单栏上的【窗口】|【项目】命令，或者使用快捷键 Ctrl+1 打开【项目】窗口。

【项目】窗口中每个节点都具有快捷菜单，其中包含可以对该节点操作的命令选项。通过右击某个节点打开该节点的快捷菜单。项目节点的弹出式菜单中包含所有主要项目命令，例如【生成】、【运行】和【调试】命令等，还包含该项目的【属性】命令，可以使用该命令来配置项目。



在 NetBeans 集成开发环境中，可以同时打开多个项目，但是同一时间只能有一个主项目，可以在【项目】窗口中进行主项目设置。

### 2. 【文件】窗口

【文件】窗口和【项目】窗口有些类似，用于显示基于目录的项目视图，其中包括【项目】窗口中未显示的文件和文件夹，如图 1-9 所示。

默认情况下，打开 NetBeans 时，将在集成开发环境的左上角打开【文件】窗口，也可以通过执行菜单栏上的【窗口】|【文件】命令，或者使用快捷键 Ctrl+2 打开【文件】窗口。

【文件】窗口中的每个节点也都有快捷菜单，其中包含可以对该节点运行的命令，例如【复制】、【删除】和【查找】等。通过右击某个节点打开该节点的快捷菜单。

### 3. 【服务】窗口

【服务】窗口是项目运行时资源的主入口点，显示项目资源的逻辑视图，如在可视化环境中注册服务器、数据库和 Web 服务，如图 1-10 所示。

默认情况下，打开 NetBeans 时，将在集成开发环境的左上角打开【服务】窗口，也可以通过执行菜单栏上的【窗口】|【服务】命令，或者使用快捷键 Ctrl+5 打开【服务】窗口。

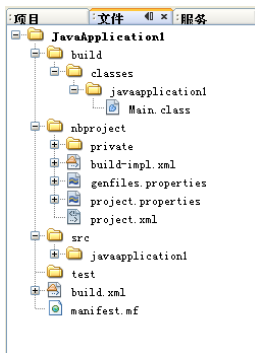


图 1-9 【文件】窗口

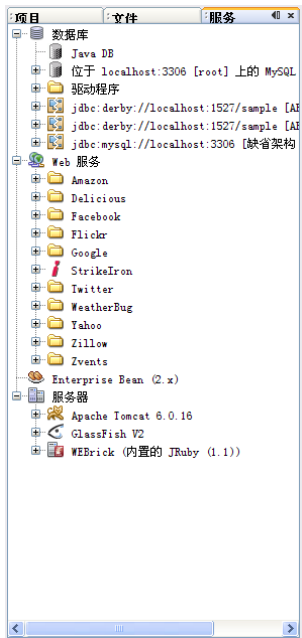


图 1-10 服务窗口

【服务】窗口中各个节点的含义如表 1-4 所示。

表 1-4 服务创建节点选项

名称	说明
服务器	服务器节点给出了当前项目可以注册的所有服务器，例如 GlassFish 和 JBoss 服务器
数据库	给出了项目当前可以使用的数据库和连接数据库的驱动程序。右击图 1-10 中的【驱动程序】节点，在弹出菜单中执行【添加驱动程序】命令，可以为 NetBeans 添加其他的数据库启动程序
Web 服务	用于管理当前所有的 Web 服务

#### 4. 【输出】窗口

【输出】窗口是一个多标签窗口，显示来自集成开发环境的消息。在出现编译错误、调试程序和生成 Javadoc 文档等情况下，将自动显示此窗口，如图 1-11 所示。【输出】窗口在运行和调试时非常实用，不仅显示了程序的运行信息，还能够显示通过 `System.out.println` 方法输出的信息以及一些异常信息。

如果要打开【输出】窗口，可以执行【窗口】|【输出】命令，或者使用快捷键 `Ctrl+4` 将其打开。有些 Java 程序运行时可能需要一些参数，此时【输出】窗口中将显示一个新标签，并且光标将停留在此标签中。可以在【输出】窗口中输入信息，此信息与在命令行中输入的信息一样。

在编译程序时，如果程序中有错误，错误信息显示为带下划线的蓝色文本，并链接到产生错误的源代码行上，如图 1-12 所示。单击该链接可以快速定位到发生错误的代码行。

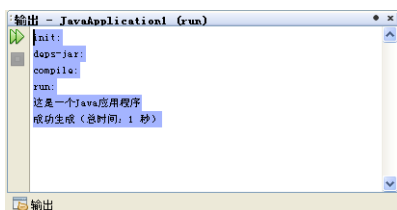


图 1-11 【输出】窗口

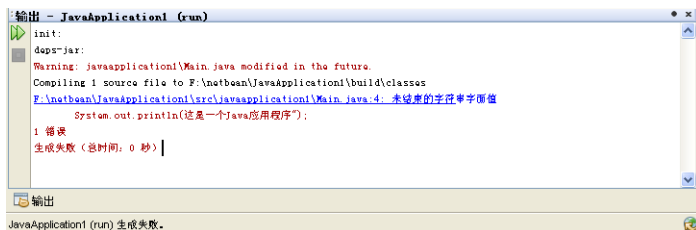


图 1-12 显示编译错误

## 5. 【导航】窗口

【导航】窗口提供了当前选定文件的简洁视图，并且简化文件不同部分的导航。如果要在源代码编辑器中打开某个文件，那么会在【导航】窗口中显示构造函数、方法和字段，如图 1-13 所示。

通过将鼠标光标停在成员的节点上，可以为成员显示 Javadoc 文档。在【导航】窗口中双击某个成员可以在源代码编辑器中快速地定位该成员。

一般情况下，将在 NetBeans 集成开发环境的左下角打开【导航】窗口。也可以通过执行【窗口】|【导航】命令，或者使用快捷键 Ctrl+7 打开【导航】窗口。要在【导航】窗口中显示源文件，请在源代码编辑器、【项目】窗口、【文件】窗口或【收藏夹】窗口中选择该文件。

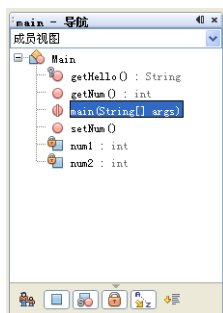


图 1-13 【导航】窗口

## 6. 【任务列表】窗口

在程序开发过程中，有些操作需要在以后实现，可以使用如图 1-14 所示的“//TODO”标记添加一个待做事项。“//TODO”文字表示待做事项的说明。

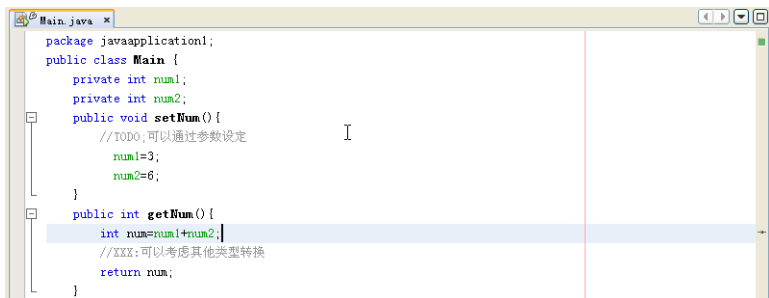


图 1-14 添加待做事项

【任务列表】窗口中可以显示项目中待做任务的列表。默认情况下，【任务列表】窗口不会自动打开，通过执行【窗口】|【任务列表】命令，或者使用快捷键 Ctrl+6 打开该窗口，如图 1-15 所示。

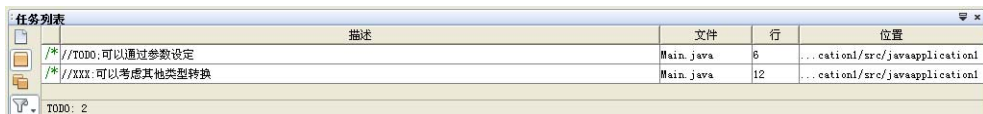



图 1-15 【任务列表】窗口

在【任务列表】窗口中可以执行以下操作，在源代码编辑器中跳转到一个待做任务。

- ❑ 双击某一个任务。
- ❑ 右击一个待做任务，然后在弹出菜单中执行【显示源】命令。

通过单击任务列表的列标题，可以对任务列表进行排序。可以使用【任务列表】窗口的工具栏来修改和处理待做任务。在【任务列表】窗口工具栏中，选定【当前文件】按钮，集成开发环境将扫描当前在源代码编辑器中打开的文件，并显示在该文件中找到的待做任务。在编辑器中打开新文件时，将扫描该文件，并且仅显示该文件中的待做任务。




在【待做任务】窗口工具栏中，如果选定【打开的项目】按钮，集成开发环境将扫描所有打开的项目，并显示这些项目中的所有待做任务。如果选定【主项目】按钮，集成开发环境将扫描当前主项目中的所有待做任务。在【待做任务】窗口工具栏中，单击【编辑过滤器】按钮，打开【任务列表过滤器】对话框，如图 1-16 所示。在该对话框中可以添加和修改过滤器，以过滤【任务列表】窗口中显示的待做任务。



图 1-16 【任务列表过滤器】对话框

下面几种方法都会创建一个待做任务，并自动被添加到【任务列表】窗口中。

- ❑ 包含 TODO、XXX 或 FLXME 的代码注释。
- ❑ 模式<<<<<<<<，它在 CVS 和其他版本控制系统中表示合并冲突。
- ❑ Java 项目中，包含@todo 关键字的 Javadoc 注释。
- ❑ Java 项目中，登记编译器错误时所在的行。

NetBeans 集成开发环境支持添加或者编辑待做任务标识。执行【工具】|【选项】命令，打开【选项】对话框。单击【其他】按钮，再打开【ToDo 任务】标签，如图 1-17 所示。在这里用户可以根据自己的习惯设置待做任务的标识，以这些标识开头的内容将会自动添加到【任务列表】窗口中。

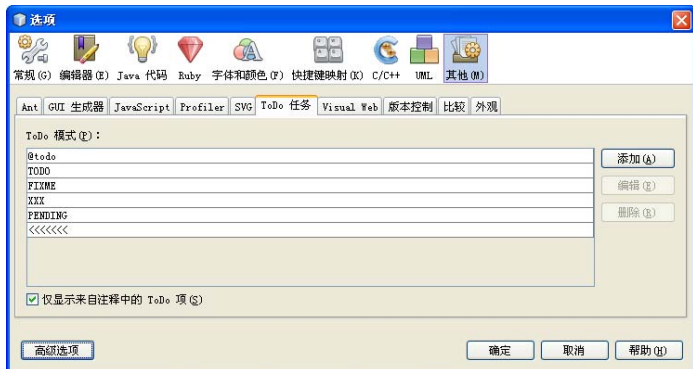


图 1-17 待做任务标签选项

## 7. 【收藏夹】窗口

使用【收藏夹】窗口，开发人员可以访问计算机或网络上的任何文件，而无须为该文件创建项目。首次打开【收藏夹】窗口时，只包含计算机的 Home 目录。对于 Windows 系统，如果登录用户为 Robin，那么这个目录默认为 C:\Documents and Settings\Robin。

默认情况下，【收藏夹】窗口不会自动打开，可以通过执行【窗口】|【收藏夹】命令，或者使用快捷键 Ctrl+3 将其打开，如图 1-18 所示。

如果要将任何文件夹添加到【收藏夹】窗口中，可以右击窗口，在弹出菜单中执行【添加到收藏夹】命令。【收藏夹】窗口中不提供任何常规项目命令，可以从【收藏夹】窗口中运行 Ant 脚本，就像在集成开发环境中的任何其他地方运行 Ant 脚本一样。

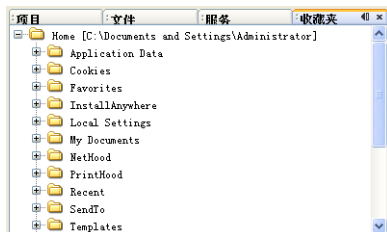


图 1-18 收藏夹窗口

## 1.3 源代码编辑器

编写程序代码是 NetBeans 集成开发工具的主要任务之一。NetBeans 集成开发环境中使用最多的就是源代码编辑器，该编辑器提供了各种可以使编写代码更简单、更快捷的功能，如代码完成、突出显示编译错误、代码重构和代码搜索等。

### 1. 快速编写代码

在 NetBeans 中输入代码时不必手工输入很长的 Java 类名、表达式或者方法名，源代码编辑器提供的代码完成功能可以帮助用户快速输入。在用户输入字符后，源代码编辑器就可以显示一个提示菜单，给出包含可能的类、方法和变量等列表，以使用户选择输入，如图 1-19 所示。

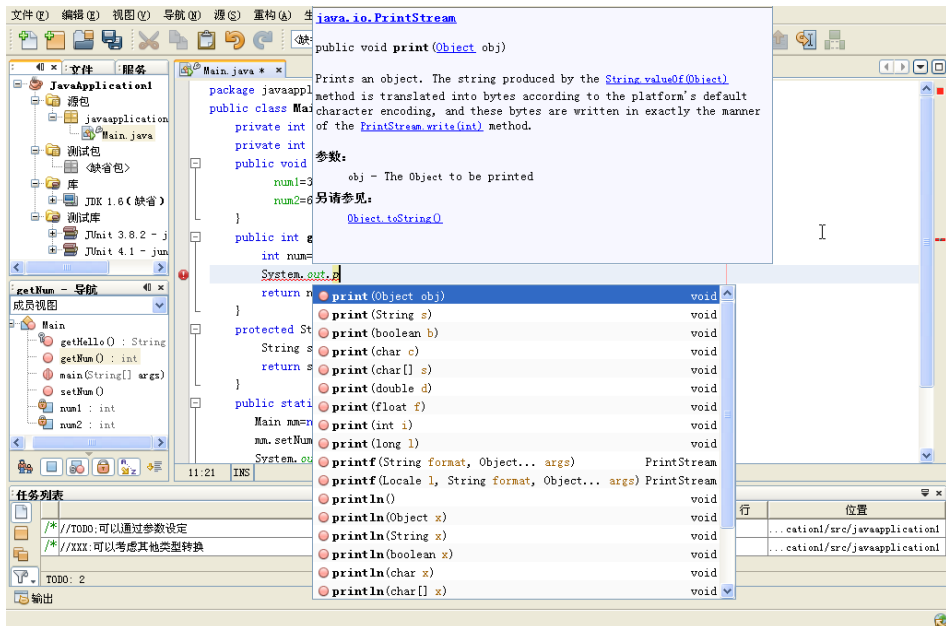


图 1-19 代码自动完成功能



另外，用户输入某些字符后，按 **Ctrl+**组合键，也会显示与输入的字符相关的提示信息以及 Javadoc 文档。

## 2. 代码跳转

14

代码跳转就是从代码一个位置快速移动到另外一个位置。NetBeans 源代码编辑器维护一个跳转列表，列表中每一个条目均代表文件中用户曾经添加或删除过文本的一个节点，或者代表曾使用查找功能或转至导航快捷键找到的一个点。用户可以根据需要使用快捷键跳到文件中先前工作过的某个位置。代码跳转可以通过【导航】菜单实现。代码跳转的详细内容将在第 2 章中介绍。

## 1.4 NetBeans 项目基础

在 NetBeans IDE 中，开发工作都将在项目内完成，NetBeans IDE 中的项目由一组源文件以及用来完成、运行和调试这些源文件的设置组成。这些设置包括哪些内容适合放在类路径中、如何生成和运行项目等。使用 NetBeans 集成开发环境可以快速地创建 Java 应用程序。

### 1.4.1 创建 Java 应用程序项目

在 NetBeans 集成开发环境中创建 Java 项目非常简单，只需要遵循 Java 项目创建向导即可。现在创建一个 FirstProject 项目，其创建步骤如下所示。

(1) 启动 NetBeans 集成开发环境，在菜单栏上执行【文件】|【新建项目】命令，打开【新建项目】对话框，如图 1-20 所示。

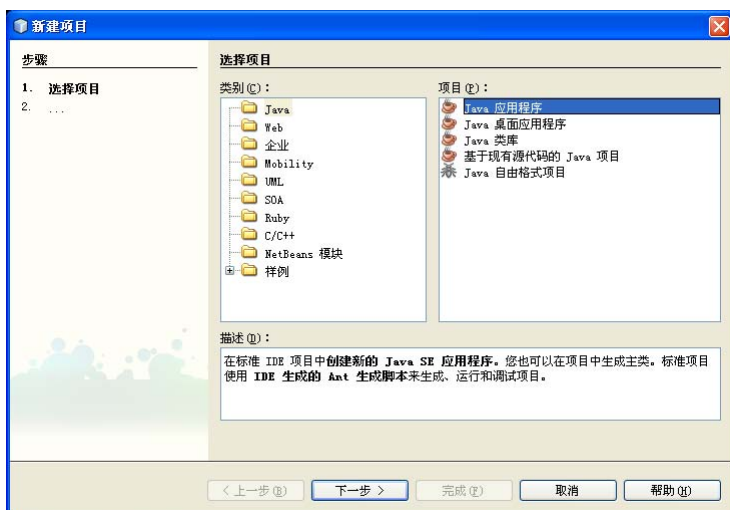


图 1-20 【新建项目】对话框

在图 1-20 中，单击【类别】列表框中的 Java 选项节点，在【项目】列表框中显示如下几

个选项节点。

- ❑ **Java 应用程序** 通过【新建应用程序】向导，创建 Java 标准应用程序的项目。
- ❑ **Java 桌面应用程序** 创建一个基于 Swing 组件的桌面应用程序框架。此模板提供基本的应用程序基础结构，如菜单栏、永久保持窗口状态以及状态栏。利用此模板，还可以生成代码以便为数据库表创建 GUI 界面。
- ❑ **Java 类库** 通过【新建 Java 类库】向导，创建不包含主类的新标准项目，该项目主要用来创建 Java 标准类库。
- ❑ **基于现有源代码的 Java 项目** 通过【新建包含现有源的 Java 项目】向导，可以使用现有的 Java SE 源文件夹创建新的标准项目。
- ❑ **Java 自由格式项目** 将现有的 Java 应用程序及其生成脚本导入 IDE 项目中。在自由格式项目中，IDE 命令使用提供 Ant 脚本来运行、编译和调试项目。对生成过程的任何修改必须直接在 Ant 脚本中执行。



要简化对项目的长期维护，请考虑使用“基于现有源代码的 Java 项目”项目模板。

(2) 在图 1-20 中，选中【Java 应用程序】选项节点，然后单击【下一步】按钮，打开如图 1-21 所示对话框。

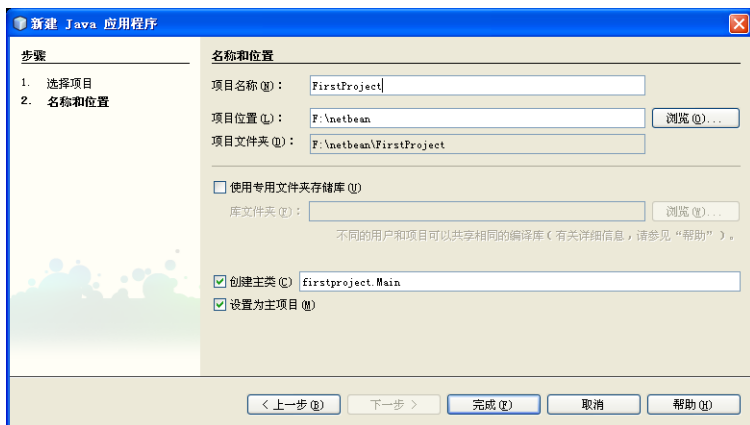


图 1-21 【名称和位置】对话框

(3) 在【项目名称】文本框中输入项目名称 FirstProject；并启用【设置为主项目】复选框，把该项目设置为主项目，其他选择默认。



用户可根据需要在【项目位置】文本框中更改项目存放的位置。如果在创建项目的同时，需要创建一个 Java 主类，可以启用【创建主类】复选框，并在后面的文本框中输入主类的路径和名称。

(4) 上述设置完成，单击【完成】按钮，完成 Java 标准项目的创建。至此创建了一个简单的 Java 标准应用程序。创建的 Java 标准项目在【项目】窗口的显示如图 1-22 所示，该项目包含源包、测试包、库和测试库等几个部分。

NetBeans 集成开发环境将创建的项目信息存储在指定文件中，如图 1-23 所示。

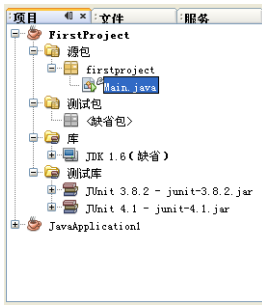


图 1-22 Java 标准项目逻辑视图

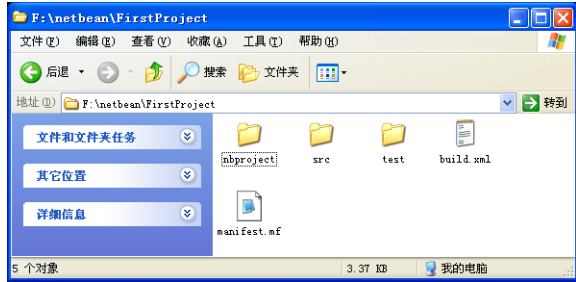


图 1-23 Java 标准项目文件夹

其中 `nbproject` 文件夹中包括 Ant 生成脚本、控制生成和运行设置的属性文件，以及若干个 XML 配置文件（例如，将 Ant 目标映射到 NetBeans IDE 命令的 `project.xml` 文件）。源目录一般包含在项目文件夹中，目录名称为 `src`，但这并不是必需的。`test` 目录通常用于保存项目的测试包。

在源代码编辑器中打开 `FirstProject` 项目中自动生成的 `Main.java` 文件，可以看到 NetBeans 自动生成的内容如代码 1.1 所示。

代码 1.1 NetBeans 自动生成内容

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package firstproject;

/**
 *
 * @author Administrator
 */
public class Main {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {           //自动生成的 main 方法
        // TODO code application logic here
    }
}


```

## 1.4.2 项目常见操作

NetBeans 中应用程序是以项目作为基本单位的，所以 NetBeans 提供了多个操作对于 NetBeans 项目，如打开项目、关闭项目、重命名项目和复制项目等。本节将会介绍常见的项目操作。

## 1. 打开项目

在 NetBeans 中可以根据需要打开一个现存项目或关闭一个已经存在的项目。在 NetBeans 开发环境中，打开项目可以使用下面的方式。

执行 NetBeans 菜单栏上的【文件】|【打开项目】命令（快捷键为 Ctrl+Shift+O），或者单击工具栏上的【打开项目】图标，将会打开如图 1-24 所示的【打开项目】对话框。

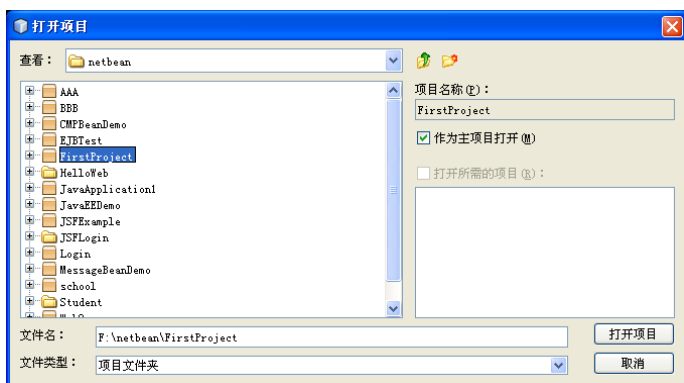


图 1-24 【打开项目】对话框

在弹出的对话框中，选择要打开项目的根文件夹。如果希望打开项目作为主项目，需要启用【作为主项目打开】复选框。单击【打开项目】命令，即可打开选中项目。

如果要打开最近刚刚关闭的项目，可以在 NetBeans 中菜单栏上，执行【文件】|【打开近期的项目】命令，打开最近打开的项目级联菜单。通过单击该级联菜单中选项可以再次打开某个项目。

## 2. 关闭项目

如果要关闭某个项目，可以通过下面的两种方法完成。

- ❑ 在【项目】窗口中选中要被关闭项目的根节点或者其子节点（即选中必须在被关闭的项目范围内），然后在菜单栏中执行【文件】|【关闭 XXX】命令。其中“XXX”为被选中项目的名称，根据项目的不同，其显示内容也不相同。
- ❑ 在【项目】窗口中右击该项目的根节点，然后执行【关闭】命令。

## 3. 重命名项目

项目创建完成后，有时可能需要重新修改项目名称。此时可以在【项目】窗口中右击项目根节点，然后执行【重命名】命令，打开如图 1-25 所示的【重命名项目】窗口。

在【项目名称】文本框中输入项目的新名称。如果需要在重新命名项目时，同时也修改项目所在文件夹名称，需要启用【同时重命名项目文件夹】复选框。设置完成后，单击【重命名】按钮即可完成项目重命名。

## 4. 复制项目

项目创建完成后，如果要对项目进行备份或复制，可以在【项目】窗口中右击项目的根节

点，然后在弹出菜单中执行【复制】命令，打开如图 1-26 所示的【复制项目】对话框。



图 1-25 【重命名项目】窗口



图 1-26 【复制项目】对话框

图 1-26 中，在【项目名称】文本框中输入新项目名称，在【项目位置】文本框中输入新项目保存位置，或者单击【浏览】按钮指定项目保存路径。最后单击【复制】按钮，即可将当前选中项目以指定新名称复制到指定目录。

## 5. 移动项目

在项目创建后如果要移动项目，可以在【项目】窗口中右击项目根节点，然后执行【移动】命令，打开【移动项目】对话框，如图 1-27 所示。

图 1-27 中，可以在【项目位置】文本框中输入项目新位置，或者单击【浏览】按钮指定项目新路径，然后单击【移动】按钮完成项目移动。



在项目移动或复制到新位置时，新文件夹必须已存在于计算机上。如果不存在，可以通过【浏览】按钮来创建文件夹。

## 6. 删除项目

如果要删除已有项目，需要在【项目】窗口中右击项目根节点，然后执行【删除】命令，打开【删除项目】对话框，如图 1-28 所示。

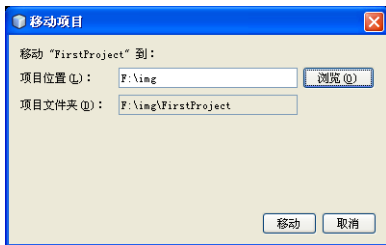


图 1-27 【移动项目】对话框

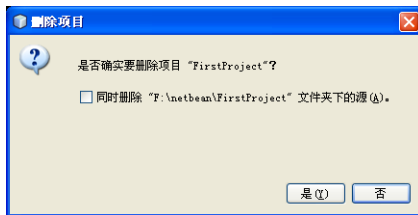


图 1-28 【删除项目】对话框

如果在删除项目的同时也要删除源代码，可以启用图 1-28 中的复选框。确认无误后，单击【是】按钮删除指定项目。

### 1.4.3 创建项目中 Java 包和类

项目创建完成之后，就可以创建类文件了。如果创建的是一个较大的项目，可能会出现大

量 Java 类源文件。为了区别这些类文件，通常需要创建 Java 包。本节将介绍使用 NetBeans 工具创建 Java 包和 Java 类。

## 1. 创建 Java 包

在 Java 应用程序中，Java 包主要解决类之间的命名冲突和类的类别划分。NetBeans 集成开发环境提供两种方式创建 Java 包，如下所示。

(1) 在 NetBeans 的【项目】窗口中，右击该项目的【源包】或【测试包】节点，在弹出菜单中执行【新建】|【Java 包】命令，如图 1-29 所示。执行命令后，打开【新建 Java 包】对话框，如图 1-30 所示。

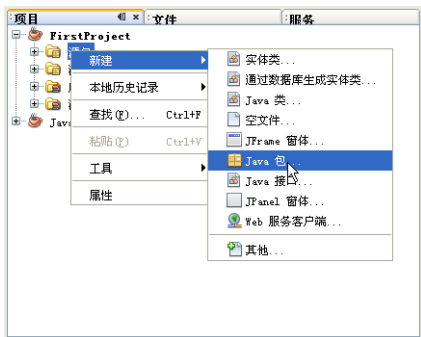


图 1-29 添加 Java 包

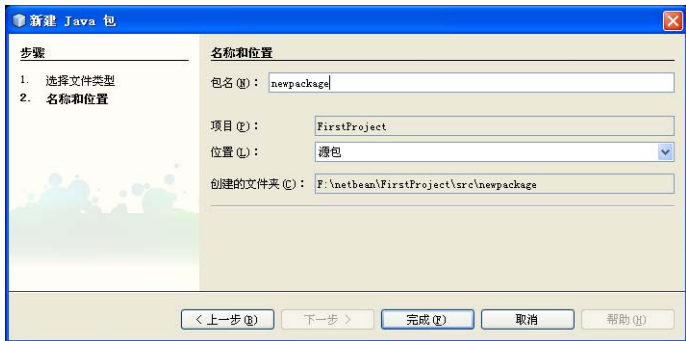


图 1-30 【新建 Java 包】对话框

在【新建 Java 包】对话框中，输入要创建 Java 包的名称和位置。在【包名】文本框中输入任意有效的名称，然后在【位置】下拉列表框中选择 Java 包创建位置。单击【完成】按钮，即可在指定位置创建一个 Java 包。

(2) 把要添加 Java 包的项目设置为主项目，然后执行【文件】|【新建文件】命令，打开如图 1-31 所示的【新建文件】对话框。



图 1-31 【新建文件】对话框

在【新建文件】对话框中，单击【类别】列表中的【Java】节点选项，然后再选择【文件

类型】列表中的【Java 包】节点。单击【下一步】按钮，打开如图 1-30 所示的【新建 Java 包】对话框。

## 2. 创建 Java 主类

Java 主类就是带有 main 方法（程序执行的入口）的 Java 类，main 方法允许 Java 类作为控制台应用程序运行的起点。在创建 Java 标准项目时，如果在图 1-21 中启用了【创建主类】复选框，那么会在创建项目的同时创建一个 Java 主类。如果没有启用该复选框，可以通过下面步骤在创建 Java 标准项目以后再创建 Java 主类。

(1) 确认要添加 Java 主类的项目是否为主项目。如果不是主项目，则在项目窗口中右击该项目根节点，然后在弹出菜单中执行【设置为主项目】命令，将该项目设置为主项目。

(2) 在 NetBeans 集成开发环境中，执行【文件】|【新建文件】命令，打开【新建文件】对话框，如图 1-31 所示。该对话框中提供了很多模板，可以帮助开发人员快速地创建原型文件。

(3) 在【类别】列表中，选择【Java】节点选项，在右边【文件类型】列表中显示常用文件类型供用户选择使用。这些常用文件类型说明如表 1-5 所示。

表 1-5 常用文件类型

名称	说明
Java 类	创建新的 plain Java 类。此模板用于创建新的非可视类
Java 接口	创建新的 Java 接口
Java 枚举	创建新的 Java 枚举文件。要能够使用枚举文件，项目必须使用 JDK 5.0 或兼容版本
Java 标注类型	创建新的 Java 标注类型。要能够使用标注类型文件，项目必须使用 JDK 5.0 或兼容版本
Java 异常	创建带有可选详细消息的新 Exception 子类
Java 包信息	创建新的 Java 包信息。要使用包标注，项目必须使用 JDK 5.0 或兼容版本
JApplet	创建新的 JFC (Swing) Applet。Applet 是可在任一已启用 Java 的浏览器上运行的 Java 类。注意：此模板不包含允许在窗体编辑器中对 Applet 进行可视化设计的窗体代码。要进行可视化设计，请从“Java GUI 窗体”下的 JApplet 模板入手
Applet	创建新的 AWT (Abstract Window Toolkit, 抽象窗口工具包) Applet。Applet 是可在任一已启用 Java 的浏览器上运行的 Java 类。注意：此模板不包含允许在窗体编辑器中对 Applet 进行可视化设计的窗体代码。要进行可视化设计，请从“Java GUI 窗体 AWT 窗体”下的 Applet 模板入手
Java 主类	创建带有 main 方法的新 Java 类，此方法允许 Java 类作为控制台应用程序运行。如果要设计可视应用程序，则最好使用“Java GUI 窗体”下的 JFrame 模板，或者使用“Java GUI 窗体 样例窗体”下的应用程序框架
空 Java 文件	创建空的 Java 源代码文件。除了生成必需的包语句外，不生成任何代码。使用此模板可以从头开始创建类
Java 包	创建用于保存 Java 源代码文件的包。此包的 actual 形式为磁盘上的空文件夹

(4) 这里选择【文件类型】列表中的【Java 主类】节点选项，然后单击【下一步】按钮，打开【名称和位置】选项卡，如图 1-32 所示。

在【类名】文本框中输入 Java 主类的类名，在【包】下拉列表中输入包的名称。

(5) 单击【完成】按钮，即可创建一个 Java 主类。



图 1-32 【名称和位置】选项卡

## 1.4.4 项目配置

项目创建完成后，可以进行项目配置，以便项目能够按照设想运行。例如可以修改项目的配置参数，也可以为项目指定另外的源代码目录、测试目录、指定编译选项、设置编译运行库和设置运行参数等。

### 1. 设置源代码选项

在【项目】窗口中右击要配置的项目节点，在弹出菜单中选择【属性】命令。或者在项目窗口中选中要配置的项目，执行【文件】|【选择的项目名称属性】命令，打开【项目属性】对话框，在左边【类别】列表选中【源】节点，如图 1-33 所示。

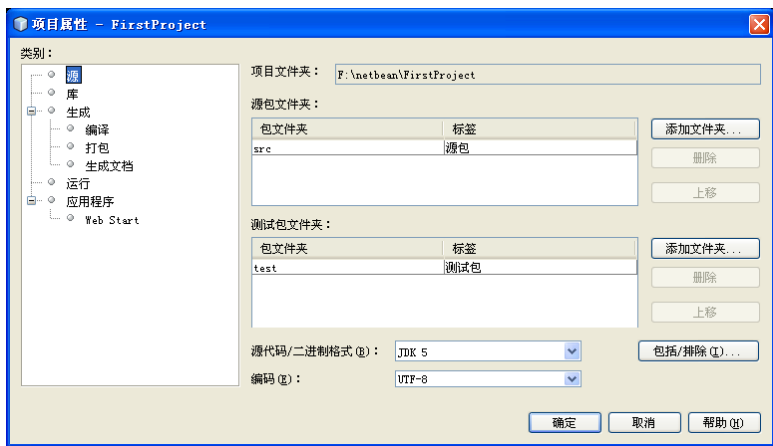


图 1-33 【项目属性】中源代码对话框

NetBeans 的标准项目允许有多个源代码文件夹，在【源包文件夹】列表中列出了这些源代码文件（默认情况下只有 src 文件夹）。可以单击【源包文件夹】列表旁的【添加文件夹】按钮，添加其他的源代码文件夹。NetBeans 标准项目也允许有多个测试代码文件夹，在【测试包文件夹】列表中列出了这些测试代码文件夹（默认情况下只有 test 文件夹）。可以单击【测试包文件夹】列表旁边的【添加文件夹】按钮，添加其他的测试代码文件夹。【源代码/二进制格式】下拉列表框用于选择 JDK 的版本。【编码】下拉列表框用于选择代码的编码格式。

## 2. 设置库选项

在【项目属性】对话框中，选中【类别】列表中【库】节点，可以设置应用程序编译和运行时的库，如图 1-34 所示。如果在项目中使用了其他 Java 库，或者引用了其他项目，那么需要在这里添加相关库文件。

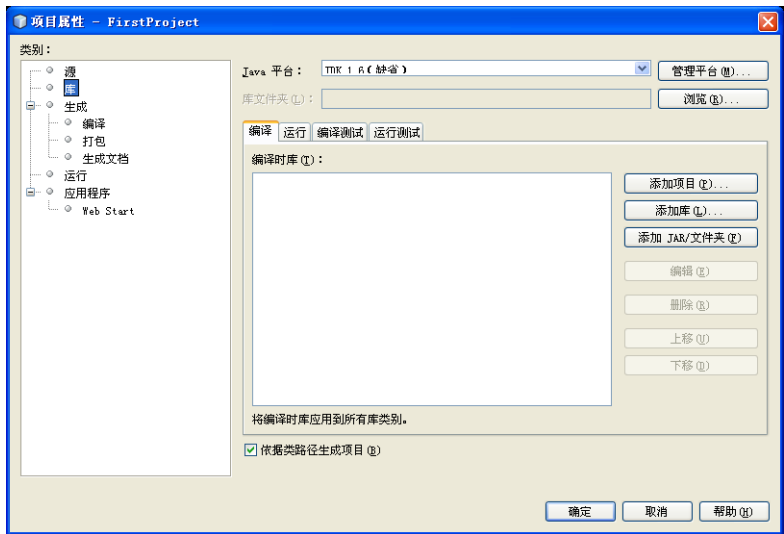


图 1-34 设置编译库

在图 1-34 中，【Java 平台】下拉列表框用来指定编译和运行时使用的 JDK 版本，如果要添加其他 Java 平台，可以单击【管理平台】按钮（或者在 NetBeans 集成开发环境中选择【工具】菜单下的 Java 平台管理器命令），打开【Java 平台管理器】对话框，如图 1-35 所示。在对话框中可以添加/删除 Java 平台。

如果项目中引用了其他项目，那么可以在 1-34 所示的对话框中单击【添加项目】按钮，引入其他项目；如果需要添加 Jar、Zip 或者包含.class 文件的文件夹，可以单击【添加 JAR/文件夹】按钮；如果要添加 NetBeans 库，可以单击【添加库】按钮，打开【添加库】对话框，如图 1-36 所示。

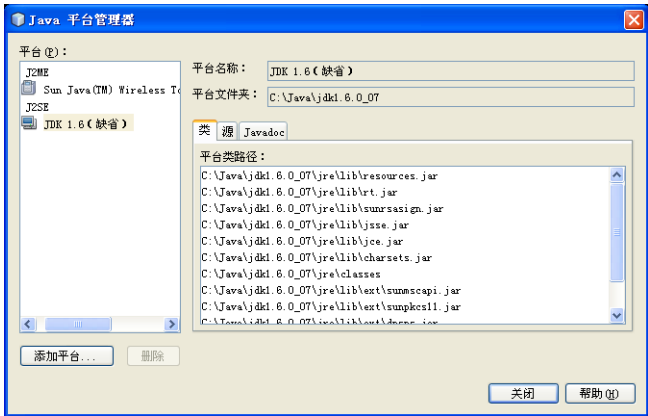


图 1-35 Java 平台管理器



图 1-36 【添加库】对话框

### 3. 设置编译选项

在【项目属性】对话框中，选中【类别】列表中的【生成】|【编译】节点，设置应用程序的编译选项，如图 1-37 所示。

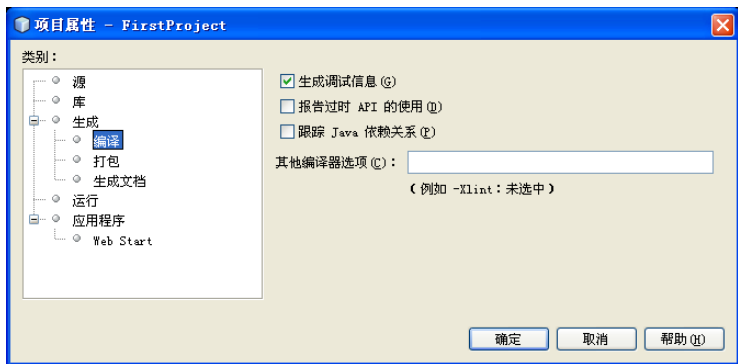


图 1-37 设置编译选项

图 1-37 中，如果启用【生成调试信息】复选框，在编译文件中会包含调试信息，可以进行跟踪调试。在开发和调试阶段应该启用该复选框。全部测试通过后，发布正式的应用程序时，可以禁用该选项。如果启用【报告过时 API 的使用】复选框，当程序中使用了当前 JDK 已经不再建议使用的 API 时，就会提示开发人员。如果在编译时指定其他编译选项，那么可以在【其他编译器选项】文本框中输入。

### 4. 设置运行选项

在【项目属性】对话框中，选中【类别】列表中【运行】节点，可以设置应用程序的运行选项，如图 1-38 所示。

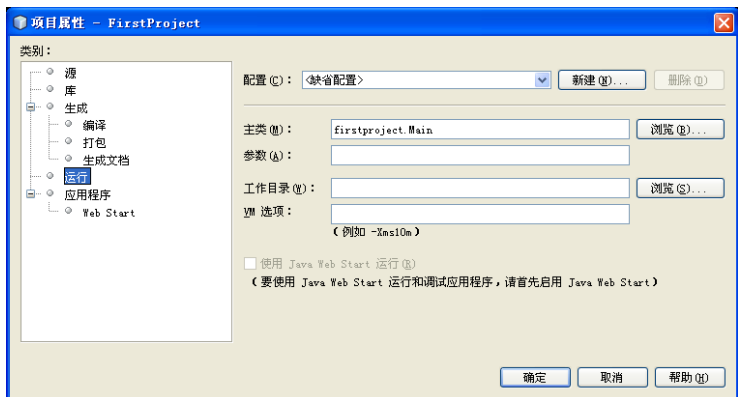


图 1-38 设置项目运行选项

在图 1-38 中，【主类】文本框中可以输入 Java 程序运行的主类，如果项目中有多个主类，可以单击【浏览】按钮，打开【浏览主类】对话框，选择哪个类要被应用程序执行。在【参数】文本框中可以输入应用程序运行时需要的一些参数。【工作目录】文本框用于指定应用程序的工作目录。【VM 选项】用于指定虚拟机的一些选项。


## 1.4.5 运行项目

在 1.4.1 小节中，创建了一个 FirstProject 项目，并且自动生成了一个 Main.java 的主类。在 NetBeans 编辑器中，打开 Main.java 文件进行编辑，其代码内容如代码 1.2 所示。

代码 1.2 Main.java 文件代码

```
package firstproject;
public class Main {
    public void getStr(){
        System.out.println("这是 NetBeans 开发的第一个 Java 程序");
    }
    public static void main(String[] args) {
        Main m=new Main();           //创建 Main 实例
        m.getStr();                   //调用 getStr 方法
    }
}
```

代码 1.2 确认无误后，就可以运行该项目了。在 NetBeans 集成开发环境中，可以通过下面几种方式来运行标准项目。

- ❑ 在【项目】窗口中右击要运行的项目，在弹出菜单中执行【运行】命令。这种方法适合主项目和非主项目。
- ❑ 执行【运行】|【运行主项目】命令（快捷键为 F6）。这种方法只适用于主项目，对于非主项目可以先设置为主项目，然后再使用这种方法。
- ❑ 在【项目】窗口中右击要运行的文件，在弹出菜单执行【运行文件】命令（快捷键为 Shift+F6）。这种方法适合主项目和非主项目。
- ❑ 在【项目】窗口中选择要运行的文件，或者在【编辑器】窗口中激活要运行的文件，然后执行【运行】菜单下的【运行文件】|【运行 XXX.java】命令（快捷键为 Shift+F6）。这种方法适合于主项目和非主项目。
- ❑ 单击工具栏上的【运行主项目】图标。这种方法只适用于主项目，对于非主项目可以先设置为主项目，然后再使用这种方法。

可以用以上任一种方法运行 FirstProject 项目，项目执行完毕后，在调试窗口输出结果，如图 1-39 所示。

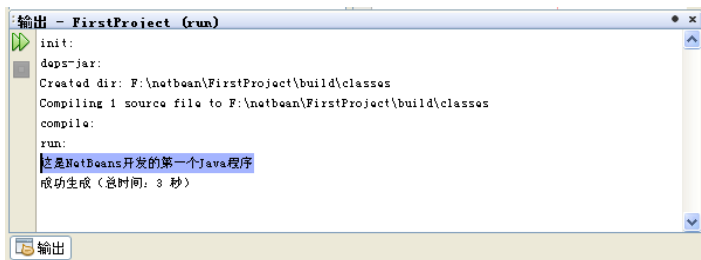


图 1-39 项目执行结果显示