

项目 1 搭建学生信息管理系统开发平台

技能目标

能进行数据的运算并能编写输入/输出数据的程序。

知识目标

- (1) 了解 Java 语言的特性。
- (2) 了解 Java 语言的发展历程及其特性。
- (3) 掌握 JDK 的安装与使用。
- (4) 掌握 MyEclipse 的使用。
- (5) 掌握 Java 两类程序运行的方法。

项目任务

本项目通过两个任务介绍 Java 语言的特点及其在软件开发语言领域中的地位,Java 系统开发平台的选择,MyEclipse 的下载与使用以及 Java 两类程序的运行方式。这两个任务包括搭建系统开发环境和编写第一个 Java 程序。

要完成 Java 欢迎界面的输出程序,第一必须学会搭建 Java 开发环境;第二必须掌握 Java 两类程序的结构、各自的命名机制及运行过程,因此将本项目分成两个任务:搭建系统开发环境和编写第一个 Java 程序。

任务 1.1 搭建系统开发环境

在开发一个系统前,首先要选择一种语言。为什么选择这种语言、这种语言有哪些特点、语言的运行环境与开发工具是什么……带着这些问题,我们来认识一下 Java,并学习简单 Java 程序的编写过程。

1.1.1 Java 语言的发展历程及特点

1. 什么是 Java

Java 是由 Sun 公司于 1995 年发布的编程语言和计算平台。这项基础技术支持最新

的程序,包括实用程序、游戏和业务应用程序。

有许多应用程序和网站只有在安装 Java 后才能正常工作,而且这样的应用程序和网站日益增多。Java 快速、安全、可靠,从笔记本电脑到数据中心,从游戏控制台到科学超级计算机,从手机到互联网,Java 无处不在。

2. Java 简史

Java 最初是为家用消费类电子产品开发分布式代码系统,后来为了使整个系统与平台无关,该项目小组的领导人 James Gosling 决定开发一种新语言,称为 Oak,这就是 Java 语言的前身,后来改名为 Java。随着 Internet 的迅速发展,Web 的应用日益广泛,Java 语言也得到了迅速发展。1994 年,Gosling 用 Java 开发了一个实时性较高、可靠安全、有交互功能的新型 Web 浏览器,它不依赖于任何硬件平台和软件平台。这种新的浏览器称为 HotJava。1995 年在业界发表,引起了巨大轰动,Java 的地位随之而得到肯定。

Java 语言发展非常迅猛。1995 年 3 月,Sun 公司发布了 Java 语言的 Alpha 1.0a2 版本,1996 年 1 月发布了 Java 语言的第一个开发包 JDK V1.0,1997 年 2 月发布了 Java 语言的开发包 JDK V1.1,从而奠定了 Java 语言在计算机语言中的地位。1998 年 12 月,Sun 公司发布了 Java 2 开发平台 JDK V1.2。Java 2 平台是 Java 发展史上的里程碑。1999 年 6 月,Sun 公司重新组织 Java 平台的集成方法,并将企业级应用平台作为 Java 语言发展方向,包含了以下 3 个成员。

- (1) J2ME(Java 2 Micro Edition): 用于嵌入式应用的 Java 2 平台。
- (2) J2SE(Java 2 Standard Edition): 用于工作站、PC 的 Java 2 标准平台。
- (3) J2EE(Java 2 Enterprise Edition): 可扩展的企业级应用的 Java 2 平台。

Java 是天生面向对象的计算机语言。虽然许多面向对象的语言一开始就是严格的过程化语言,Java 却从一开始就被设计为面向对象的语言。

3. Java 程序运行机制和 JVM

计算机高级语言按照程序的运行方式可以分为编译型和解释型两种。

编译型语言使用专门的编译器,针对特定的平台将某种高级语言“翻译”成该平台可以识别的机器码,并包装成该平台可以识别的可执行程序。例如,C、C++ 和 FORTRAN 等高级语言都属于编译型语言。

解释型语言使用专门的解释器将源代码逐行解释成特定平台的机器码并且立即执行,解释型语言不会进行整体的编译和链接处理。现在的 Python、Ruby 等语言都属于解释型语言。

Java 语言的机制比较特殊,Java 编写的程序会经过编译步骤,但是不会编译成特定平台的机器码,而是生成一种与平台无关的字节码——.class 文件。这种字节码是不能直接执行的,需要 Java 的解释器来进行解释执行。因此 Java 不是纯粹的编译型语言或纯粹的解释型语言,它需要先编译,然后解释执行。

Java 语言负责解释执行字节码文件的是 Java 虚拟机,即 JVM(Java virtual machine)。所有平台上的 JVM 向 Java 编译器提供相同的接口,因此编译器只需要面向

虚拟机,生成虚拟机能够理解的代码。要想在不同的平台上运行相同的机器码基本是不可能的。Java 通过 Java 虚拟机很好地解决了移植性问题。用户编写的程序是面对 Java 虚拟机的,至于系统的差异性则由 Java 虚拟机来解决。

Java 虚拟机可以理解为软件模拟的计算机,可以在任何处理器上(无论是在计算机中还是在其他电子设备中)安全并且兼容地执行程序。只要在不同平台上安装相应的 JVM,就可以运行字节码文件,运行 Java 程序。

而在这个过程中,Java 程序没有做任何改变,仅仅是通过 JVM 这一“中间层”,就能在不同平台上运行,真正实现了“一次编译,到处运行”的目的。

Java 从 1.2 版本开始,针对不同的应用领域,分为了 3 个不同的平台: J2SE、J2EE 和 J2ME。它们分别是 Java 标准版(Java standard edition)、Java 企业版(Java enterprise edition)和 Java 微型版(Java micro edition)。Java 标准版是基础,学习 Java 一般都是从标准版开始。本书讲述的就是 Java 标准版的程序设计。

有以下两点需要注意。

(1) Java 5.0 版本后,J2EE、J2SE、J2ME 分别更名为 Java EE、Java SE、Java ME。

(2) 每个版本名称中都带有一个数字 2,这个 2 是指 Java 2: 自从 Java 1.2 发布后,Java 改名为 Java 2(不过平时仍然称为 Java)。

4. Java 语言的特点

Java 是一种编程语言,除了具有所有编程语言的共同特点之外,由于 Java 主要用于网络编程,使得 Java 语言具有其他编程语言所不具有的诸多特点,可以概括为以下 5 点。

1) Java 是完全面向对象的编程语言

Java 是完全面向对象的编程语言,在 Java 中一切都是类。Java 利用类和对象的机制将数据和对数据的操作封装在一起,并通过统一的接口与外界实现交互,使程序中的各个类彼此独立、自治又能继承,大大提高了程序的可维护性和可复用性,同时大大提高了开发效率。

Java 的编程过程就是设计类、继承类、实现类和定义、调用类的属性、方法的过程。

2) Java 是编译解释型的编程语言

以往的编程语言可分为编译型语言和解释型语言两种。

编译型语言的优点是编译成可执行的 EXE 文件后不再需要编程环境的支持,但是正因为如此,编译型语言编译时往往将许多函数放在 EXE 文件之中或者需要带有函数库文件。编译生成的 EXE 文件加上函数库文件往往很大,在网络上传输比较困难。

解释型语言的执行离不开解释器,这种解释器往往就是开发程序的编程环境。用户要执行解释程序就必须安装相应的编程环境。从网络传输的角度来讲,解释型语言传输的是源代码文件,虽然源代码文件比 EXE 文件加上函数库文件小许多,但丧失了程序的保密性。

也有些语言在编程阶段采用解释运行,程序编制完成后再编译成可执行程序。此种方式兼具了两者的优点,但在网络传输方面其缺点依然存在。

Java 是编译解释型的编程语言。Java 的源程序不能解释执行,必须使用编译器

javac.exe 进行编译。但是 Java 并不把源程序编译成可执行的 EXE 文件,而是编译成比 EXE 文件小很多的字节码文件,即扩展名为.class 的一个或多个文件,这种很小的字节码文件极其有利于在网络上传输。用户在网络上获取这种字节码文件之后不能直接执行,需要一种“解释器”。然而只要计算机安装了网络浏览器(例如 Microsoft 的 IE)就同时安装了这种“解释器”,所以以 Java 的字节码文件形式出现的 Java 程序在网络上畅通无阻的。

3) Java 是跨平台的编程语言

由于在网上传播的是 Java 的字节码,对字节码解释执行的任务交由浏览器负责,所以 Java 程序有良好的跨平台特性,即在一种系统下编制的 Java 程序一经编译成字节码文件,就可以不加修改地在任何系统中运行。这种普遍适用的程序大大降低了程序开发、维护、管理的成本。

4) Java 是适合在网上运行的编程语言

如前所述,在网上传输的只是 Java 程序的字节码,而且 Java 程序中的每一个类都单独编译成一个字节码文件,所以传输量小、传输速度快,因此 Java 是适合在网上运行的编程语言。

5) Java 是支持多线程的编程语言

多线程技术是指在同—个应用程序中有两个或更多个执行线程,即一个应用程序能够同时做两件以上的事情,这就满足了程序对某些复杂功能(如动画)的要求。Java 语言内置了多线程功能,提供了语言级的多线程支持,预先定义了一些用于建立、管理多线程的类和方法,使得开发多线程应用程序变得简单、方便而且有效。

1.1.2 Java 开发工具的选择

Java 的应用越来越广泛,学习 Java 的人也越来越多。学过程序设计的人都知道,使用 Basic 进行程序设计,可以使用 QBasic、Visual Basic 等开发工具;使用 C 语言进行程序设计,可以使用 Turbo C、Visual C++、C++ Builder 等开发工具。这些开发工具集成了编辑器和编译器,是集成开发工具,很方便使用。学习 Java 程序设计,同样需要方便易用的开发工具。Java 的开发工具很多,而且各有优缺点,初学者往往不知道有哪些常用的开发工具,或者由于面临的选择比较多而产生困惑。下面对初学者常使用的 Java 开发工具进行介绍,有助于初学者了解 Java 常用开发工具并做出选择。

1. JDK(Java development kit,Java 开发工具集)和文本编辑器

从初学者角度来看 Java 开发工具,采用 JDK 开发 Java 程序能够很快理解程序中各部分代码之间的关系,有利于理解 Java 面向对象的设计思想。JDK 的另一个显著特点是随着 Java(J2EE、J2SE 以及 J2ME)版本的升级而升级。

但它的缺点也是非常明显的,比如从事大规模企业级 Java 应用开发非常困难,不能进行复杂的 Java 软件开发,也不利于团体协同开发。

2. JBuilder

JBuilder 是一个 Java 集成开发环境,它能够满足很多方面的应用,尤其是对于服务器方以及 EJB 开发者。JBuilder 环境开发程序方便,它是纯的 Java 开发环境,适合企业的 J2EE 开发;缺点是往往一开始人们难以把握整个程序各部分之间的关系,对机器的硬件要求较高,比较耗内存,这时运行速度显得较慢。

3. Visual Age for Java

Visual Age for Java 是一个非常成熟的开发工具,它的特性对于 IT 开发者和业余的 Java 编程人员来说都是非常有用的。它提供对可视化编程的广泛支持,支持利用 CICS 连接遗传大型机应用,支持 EJB 的开发应用,支持与 Websphere 的集成开发,方便的 Bean 创建和良好的快速应用开发(RAD)支持和无文件式的文件处理。

Visual Age for Java 独特的管理文件方式使其集成外部工具非常困难,用户无法让 Visual Age for Java 与其他工具一起联合开发应用。

4. JCreator

JCreator 是一个 Java 开发工具,也是一个 Java 集成开发环境。不论是要开发 Java 应用程序还是网页上的 Applet 都难不倒它。在功能上与 Sun 公司所公布的 JDK 等文字模式开发工具相比,其操作更加容易,还允许使用者自定义操作窗口界面及无限 Undo/Redo 等功能。

JCreator 为用户提供了相当强大的功能,如项目管理功能,项目模板功能,可个性化设置语法高亮属性、行数、类浏览器、标签文档、多功能编译器,向导功能以及完全可自定义的用户界面。通过 JCreator,不用激活主文档就可以直接编译或运行 Java 程序。

JCreator 能自动找到包含主方法的文件或包含 Applet 的 HTML 文件,然后它会运行适当的 Java 开发工具。在 JCreator 中,可以通过一个批处理同时编译多个项目。JCreator 的设计接近 Windows 界面风格,用户对它的界面比较熟悉。其最大特点是与机器中所装的 JDK 完美结合。它是一种初学者很容易上手的 Java 开发工具,其缺点是只能进行简单的程序开发,不能进行企业 J2EE 的开发应用。

5. IDEA

IDEA 全称 IntelliJ IDEA,是用于 Java 语言开发的集成环境(也可用于其他语言),IntelliJ 在智能代码助手、代码自动提示、重构、J2EE 支持、Ant、JUnit、CVS 整合、代码审查、创新的 GUI 设计等方面的功能比较强大。

6. Eclipse 与 MyEclipse

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。Eclipse 只是一个框架和一组服务,用于通过插件组件构建开发环境。Eclipse 附带了一个标准的插件集,包括 Java 开发工具(Java development tools, JDT),还包括插件开发环境(plug-in development

environment, PDE), 这个组件主要针对希望扩展 Eclipse 的软件开发人员, 因为它允许开发人员构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每样东西都是插件, 对于给 Eclipse 提供插件, 以及给用户提供一个一致和统一的集成开发环境而言, 所有工具开发人员都具有同等的发挥场所。

注: IDEA 功能强大, 配置项繁多, 更多尊重开发者自己的选择, 专注感更强, 写代码的体验极佳, 更适合开发熟手。Eclipse 使用简单, 大局观更佳, 更适合初学者和代码管理者, 所以本书采用的是 Eclipse/MyEclipse 开发工具。

1.1.3 JDK 的下载与安装

1. Java 开发工具包 JDK 的下载

JDK 是整个 Java 的核心, 包括 Java 运行环境 JRE(Java runtime environment)、Java 工具和 Java 基础类库。

可以通过 Java 的官方网站: <http://Java.sun.com/Javase/downloads/index.html> 下载所需的 Java 版本的 JDK。

2. JDK 的安装

(1) 双击下载后的 JDK 软件包如 `jdk_1.8.0.0_64.exe`, 开始安装。

(2) 安装程序首先解开压缩, 解压后选择接受许可协议, 然后单击“下一步”按钮。

(3) 接下来, 为 JDK 指定安装目录。如果想指定安装目录, 则单击“更改”按钮, 选择指定目录。如果没有特殊需要可以不作改动, 如图 1-1 所示。



图 1-1 为 JDK 指定安装目录

(4) 自定义安装确认之后, JDK 开始安装至硬盘中, 稍等几分钟即可。

(5) 完成后, 单击“下一步”按钮完成安装。

完成安装后,在安装目录下应该有两个子目录: jdk1.8.0_31 与 jre1.8.0_31。以 jdk1.8.0_31 目录为例,每个目录的含义如下。

- ① C:\Program Files\Java\jdk1.8.0_31\bin 包括 Java 的一些常用开发工具。
- ② C:\Program Files\Java\jdk1.8.0_31\lib 包括 Java 的一些开发库。
- ③ C:\Program Files\Java\jdk1.8.0_31\demo 包括一些演示实例。
- ④ C:\Program Files\Java\jdk1.8.0_31\include 包含一些头文件(以.head 为文件扩展名)。

3. 环境配置

在 Windows 7 操作系统中,打开“控制面板”→“系统”→“高级”→“环境变量”或者右击“我的电脑”,选择“属性”→“系统高级设置”,然后单击“环境变量”,进入“环境变量”对话框,如图 1-2 所示。单击“新建”按钮,创建变量 javahome,变量值为 D:\Program Files (x86)\Java\jdk1.8.0_31,即 JDK 的安装路径,如图 1-3 所示。

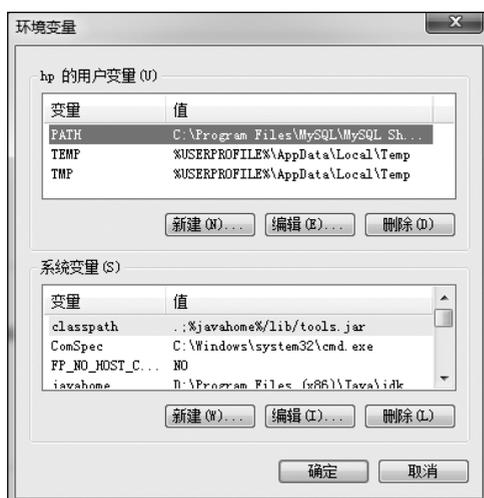


图 1-2 环境配置：环境变量



图 1-3 环境配置：变量名 javahome

然后需要创建另外一个环境变量 classpath,变量值为.;%JAVA_HOME%\lib\tools.jar。注意前面的句点不能省略。该变量指明了 Java 程序在运行时需要用到类的路径,如图 1-4 所示。

PATH 环境变量原来 Windows 里面就有,只需修改一下,使它包含指向 JDK 的 bin 目录,这样在控制台(DOS 界面)下面编译、执行程序时就不需要再输入完整路径了。设置方法是:保留原来的 PATH 的内容,在原有内容末尾先添加分号,然后加上%JAVA_HOME%\bin,如图 1-5 所示。

4. JDK 常用命令

在“运行”对话框中输入 cmd,单击“确定”按钮,从而进入 DOS 界面,输入 javac 命令,若如图 1-6 所示,则表示已经安装成功,否则没有成功。

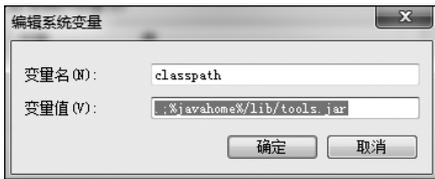


图 1-4 环境配置：变量名 classpath



图 1-5 环境配置：变量名 Path

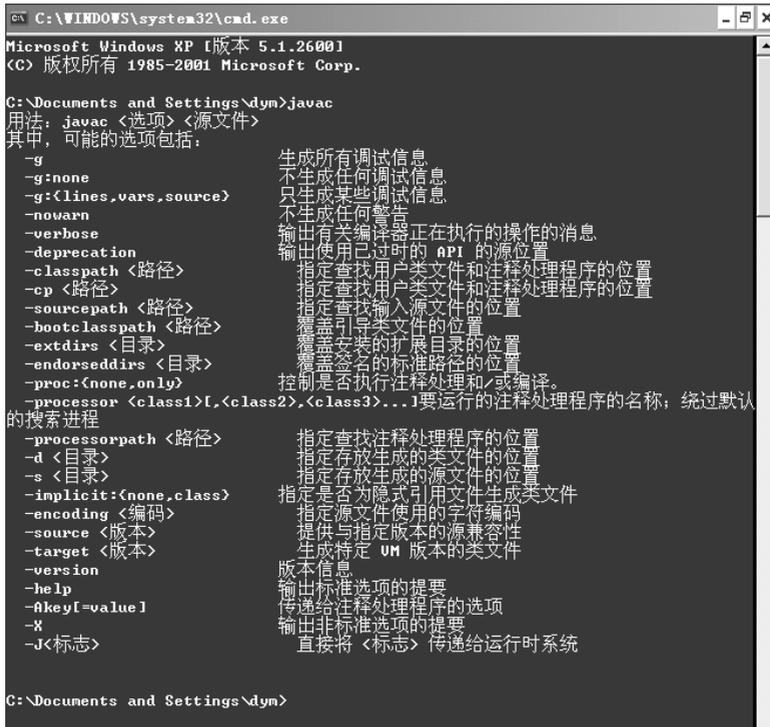


图 1-6 JDK 安装成功

JDK 的常用命令有以下几个。

1) Java 编译器——javac

javac 命令的使用格式如下。

```
javac [<编译选项>] <Java 源程序文件列表>
```

2) Java 解释器——java

java 命令作为运行环境中的解释器，负责解释并执行编译产生的类。它的使用格式如下。

```
java [<功能选项>] <主类名> [<参数列表>]
```

3) Applet 浏览器——appletviewer

appletviewer 程序的功能是下载 HTML 文档中包含的 Applet 小程序，然后在自身

的浏览器窗口中予以执行。该程序的使用格式如下。

```
appletviewer <选项><HTML 文档或网址>
```

4) Java 调试器——jdb

jdb 是一个基于命令行的调试工具,使用这一工具可以实施逐行执行程序、设置断点、检查变量的当前值等操作。jdb 命令的使用格式如下。

```
jdb <功能选项><类名><参数>
```

1.1.4 MyEclipse 的下载与使用

1. Eclipse 安装

Eclipse 软件包可以到官方网站下载,下载链接为 <http://www.eclipse.org/downloads/>。在 Windows 下运行 Eclipse,除了需要 Eclipse 软件包外,还需要 JDK 的支持。

2. MyEclipse 下载与安装

MyEclipse 安装过程比较简单,只须接受默认安装选项即可。

3. MyEclipse 的基本使用

当下载并成功安装 MyEclipse 后,就可以正式使用了,其基本的使用方法如下。

1) 设置工作空间(workspace)

若是第一次启动并使用 MyEclipse,系统会提示用户选择工作空间(workspace,实质上就是项目文件所在的空间)的路径,如 D:\2020\MyEclipse 10,如图 1-7 所示。

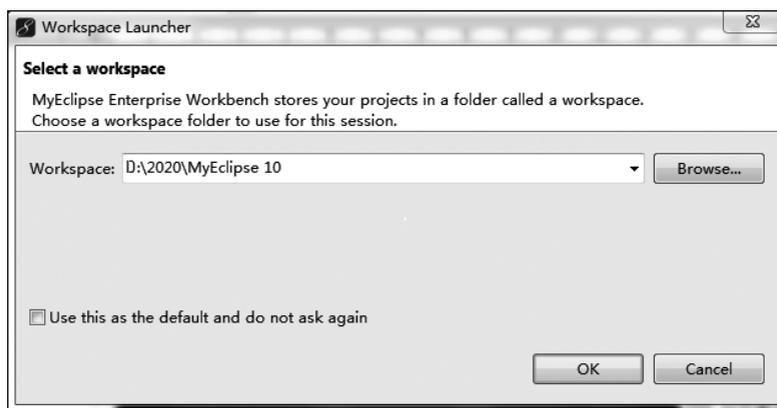


图 1-7 MyEclipse 工作空间路径设置界面

当成功建立工作空间之后,系统会在指定的路径下创建文件夹,以后就会在 Workspace 文件夹下面存放项目文件。注意,每个项目是一个独立的文件夹。

2) 创建项目(project)

MyEclipse 为开发者提供了合理且强大的项目管理功能,它以树状形式很好地展现了项目中各个文件及文件夹(包)之间的关系。当用户创建了自己的工作空间之后,系统会自动进入工作界面,如图 1-8 所示。

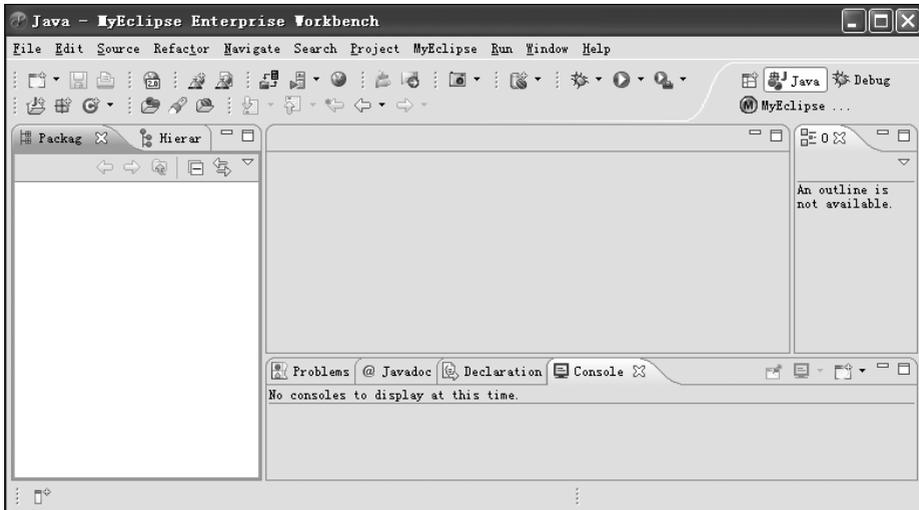


图 1-8 MyEclipse 工作界面

此时,工作空间中还没有任何内容,因为还没有创建一个项目,创建项目的步骤如下。

(1) 打开创建项目的界面,具体操作如图 1-9 所示。

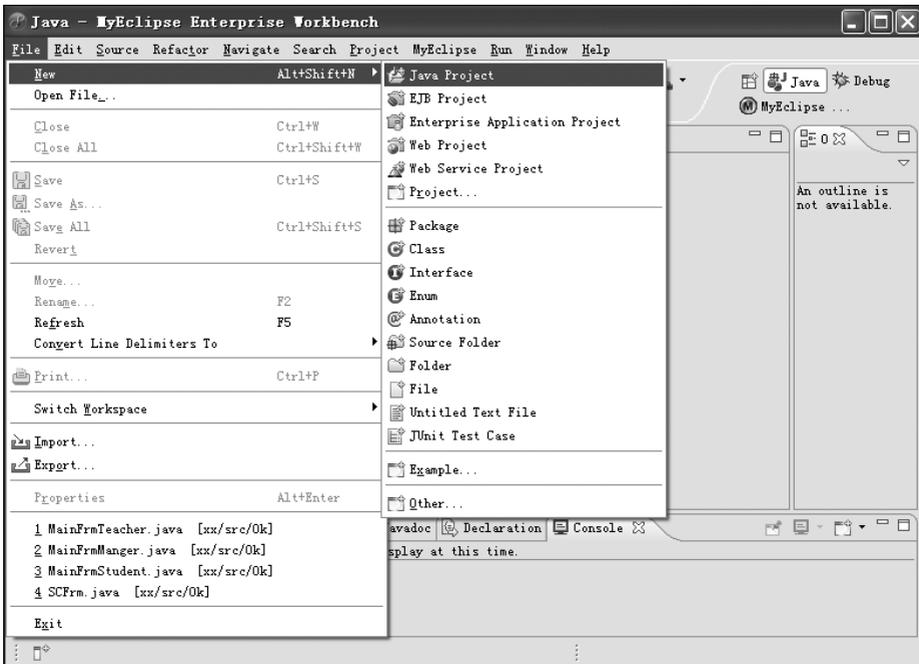


图 1-9 创建项目