

---

# 基于 J2SE 的 Java 程序设计毕业论文

## 目录

<a href="#">引言</a>	1
<a href="#">1、开发环境概述</a>	1
<a href="#">1.1 开发平台及开发工具概述</a>	1
<a href="#">1.1.1 Java 介绍</a>	1
<a href="#">1.1.2 Java 的特点</a>	2
<a href="#">1.1.3 Java 平台介绍</a>	3
<a href="#">1.1.4 类</a>	3
<a href="#">1.1.5 方法</a>	4
<a href="#">1.2 Swing</a>	4
<a href="#">1.2.1 Swing 介绍</a>	4
<a href="#">1.2.2 Swing 的原理</a>	4
<a href="#">1.2.3 Swing 与 AWT 的关系</a>	5
<a href="#">1.2.4 创建图形界面步骤</a>	5
<a href="#">1.3 需求概述</a>	6
<a href="#">1.4 简历生成程序的功能要求</a>	6
<a href="#">1.5 运行环境</a>	6

---

<u>2、环境搭建</u>	6
<u>2.1 安装 JDK</u>	6
<u>2.2 设置环境变量</u>	7
<u>2.3 HelloWorld 程序</u>	11
<u>2.3.1 编写源程序</u>	11
<u>2.3.2 编译程序代码并运行</u>	11
<u>3、设计规划</u>	13
<u>3.1 程序说明</u>	13
<u>3.2 总体设计说明</u>	13
<u>4、详细设计</u>	14
<u>4.1 界面设计</u>	14
<u>4.2 信息输入模块</u>	15
<u>4.3 信息预览模块</u>	18
<u>4.4 信息操作模块</u>	19
<u>5、结论</u>	22
<u>6、致谢</u>	23
<u>7、参考文献</u>	24

---

## 1、开发环境概述

### 1.1 开发平台及开发工具概述

开发平台：J2SE

开发工具：Java Version 6 Update 11 、 Swing

#### 1.1.1 Java 介绍

Java 是 1995 年 6 月有 Sun 公司引进到这个世界的革命性的编程语言，它被美国的著名杂志《PC Magazine》评为 1995 年十大优秀科技产品。之所以称 Java 为革命性编程语言，是因为传统的软件往往与具体的实现环境有关，一旦环境有所变化就需要对软件进行一番改动，耗时费力，而 Java 编写的软件能在执行码上兼容。这样，只要计算机提供了 Java 解释器，Java 编写的软件就能在其上运行。

Java 的出现是源于对独立于平台语言的重要，希望这种编程语言能编写出嵌入各种家用电器等设备的芯片上、且易于维护的程序。但是，人们发现当时的编程语言，比如 C、C++等都有一个共同的缺点，那就是针对 CPU 芯片进行编译。这样，一旦电气设备更换了芯片就不能保证程序正确运行，就可能需要修改程序并针对新的芯片重新进行编译。1990 年 Sun 公司成立了有 James Gosling 领导的开发小组，开始致力于开发一种可移植的、跨平台的编程语言，该编程语言能生成正确运行与各种操作系统、各种 CPU 芯片上的代码。他们的精心专研和努力促成了 Java 的诞生。Java 的快速发展得利于 Internet 和 Web 的出现，Internet 上各

WORD 版本.

---

种不同的计算机，它们可能使用完全不同的操作系统和 CPU 芯片，但仍希望运行相同的程序，Java 的出现标志着真正的分布式系统的到来。

### 1.1.2 Java 的特点

1、Java 语言是简单的。Java 语言的语法与 C 语言和 C++ 语言很接近，使得大多数程序员很容易学习和使用 Java。另一方面，Java 丢弃了 C++ 中很少使用的、很难理解的、令人迷惑的那些特性，如操作符重载、多继承、自动的强制类型转换。特别地，Java 语言不使用指针，并提供了自动的废料收集，使得程序员不必为存管理而担忧。

2、Java 语言是一个面向对象的。Java 语言提供类、接口和继承等原语，为了简单起见，只支持类之间的单继承，但支持接口之间的多继承，并支持类与接口之间的实现机制（关键字为 implements）。Java 语言全面支持动态绑定，而 C++ 语言只对虚函数使用动态绑定。总之，Java 语言是一个纯的面向对象程序设计语言。

3、Java 语言是分布式的。Java 语言支持 Internet 应用的开发，在基本的 Java 应用编程接口中有一个网络应用编程接口（java.net），它提供了用于网络应用编程的类库，包括 URL、URLConnection、Socket、ServerSocket 等。Java 的 RMI(远程方法激活)机制也是开发分布式应用的重要手段。

4、Java 语言是健壮的。Java 的强类型机制、异常处理、废料的自动收集等是 Java 程序健壮性的重要保证。对指针的丢弃是 Java 的明智选择。Java 的安全检查机制使得 Java 更具健壮性。

---

5、Java 语言是安全的。Java 通常被用在网络环境中，为此，Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外，Java 对通过网络下载类具有一个安全防机制（类 `ClassLoader`），如分配不同的名字空间以防替代本地的同名类、字节代码检查，并提供安全管理机制（类 `SecurityManager`）让 Java 应用设置安全哨兵。

6、Java 语言是体系结构中立的。Java 程序（后缀为 `java` 的文件）在 Java 平台上被编译为体系结构中立的字节码格式（后缀为 `class` 的文件），然后可以在实现这个 Java 平台的任何系统中运行。这种途径适合于异构的网络环境和软件的分发。

7、Java 语言是可移植的。这种可移植性来源于体系结构中立性，另外，Java 还严格规定了各个基本数据类型的长度。Java 系统本身也具有很强的可移植性，Java 编译器是用 Java 实现的，Java 的运行环境是用 ANSIC 实现的。

8、Java 语言是解释型的。如前所述，Java 程序在 Java 平台上被编译为字节码格式，然后可以在实现这个 Java 平台的任何系统中运行。在运行时，Java 平台中的 Java 解释器对这些字节码进行解释执行，执行过程中需要的类在联接阶段被载入到运行环境中。

9、Java 是高性能的。与那些解释型的高级脚本语言相比，Java 的确是高性能的。事实上，Java 的运行速度随着 JIT(Just-In-Time) 编译器技术的发展越来越接近于 C++。

10、Java 语言是多线程的。在 Java 语言中，线程是一种特殊的对象，它必须由 `Thread` 类或其子（）类来创建。通常有两种方法来创建线程：其一，使用型构为 `Thread(Runnable)` 的构造子将一个实现了 `Runnable` 接口的对象

---

包装成一个线程，其二，从 **Thread** 类派生出子类并重写 **run** 方法，使用该子类创建的对象即为线程。值得注意的是 **Thread** 类已经实现了 **Runnable** 接口，因此，任何一个线程均有它的 **run** 方法，而 **run** 方法中包含了线程要运行的代码。线程的活动由一组方法来控制。**Java** 语言支持多个线程的同时执行，并提供多线程之间的同步机制（关键字为 **synchronized**）。

11、**Java** 语言是动态的。**Java** 语言的设计目标之一是适应于动态变化的环境。**Java** 程序需要的类能够动态地被载入到运行环境，也可以通过网络来载入所需要的类。这也有利于软件的升级。另外，**Java** 中的类有一个运行时刻的表示，能进行运行时刻的类型检查。

### 1.1.3 **Java** 平台介绍

**Sun** 公司要实现“一次写成，处处运行”（**write once,run anywhere**）的目标，就必须提供相应的 **Java** 运行平台，目前 **Java** 运行平台主要分为下列 3 个版本。

- 1、**J2SE**: 称为 **Java** 标准版或 **Java** 标准平台。**J2SE** 提供了标准的 **SDK** 开发平台（以前称为 **JDK** 开发平台）。利用该平台可以开发 **Java** 桌面应用程序和低端的服务器应用程序，也可以开发 **Java Applet** 程序。
- 2、**J2EE** 称为 **Java** 企业版或 **Java** 企业平台。使用 **J2EE** 可以构建企业级的服务器应用程序，**J2EE** 平台包含了 **J2SE** 平台，并增加了附加类库，以便支持目录管理、交易管理和企业级消息处理等功能。
- 3、**J2ME** 称为 **Java** 微型版或 **Java** 小型平台。**J2ME** 是一种很小的 **Java** 运行环境，

WORD版本.

---

用于嵌入式的消费产品中，如移动、掌上电脑或其它无线设备等。

#### 1.1.4 类

类是组成 Java 程序的基本要素。类封装了一类对象的状态和方法。类是用来定义对象的模板。类的实现包括两部分：类声明和类体。基本格式为：

```
class 类名
{
    类的容
}
```

`class` 是关键字，用来定义类。“`class 类名`”是类的声明部分，类名必须是合法的 Java 标识符。两个大括号以及之间的容是类体。

编写类的目的是为了描述一类事物共有的属性和功能，描述过程由类体来实现。类体的容有两部分构成：一部分是变量的定义，用来刻画属性；另一部分是方法的定义，用来刻画功能。

#### 1.1.5 方法

一个类的类体由两部分组成：变量的定义和方法的定义。方法的定义包括两部分：方法声明和方法体。一般格式为：

```
方法声明部分
{
    方法体的容
}
```

WORD版本.

---

## 1.2 Swing

### 1.2.1 Swing 介绍

Swing 是一个用于开发 Java 应用程序用户界面的开发工具包。它以抽象窗口工具包 (AWT) 为基础使跨平台应用程序可以使用任何可插拔的外观风格。Swing 开发人员只用很少的代码就可以利用 Swing 丰富、灵活的功能和模块化组件来创建优雅的用户界面。

Swing 组件中大部分组件是轻量组件，没有同位体。Swing 组件的轻量组件在设计上和 AWT 完全不同。轻量组件把与显示组件有关的许多工作和处理组件事件的工作交给相应的 UI 代表来完成，这些 UI 代表是用 Java 语言编写的类，这些类被增加到 Java 的运行环境中，因此，组件的外观不依赖平台，不仅在不同平台上的外观是相同的，而且较重量组件而言有更高的性能。如果 Java 编程环境或运行环境低于 1.2 版本，就不能使用 Swing 组件或运行含有 Swing 组件的程序。

### 1.2.2 Swing 的原理

Swing 组件类的设计原理是建立在称为模型 - 视图 - 控制器 (Model-View-Controller) 结构基础上，也称为 MVC。MVC 的设计思想并非源于 Java，MVC 是 Trygve Reenskaug 教授于 1978 年最早开发的一个设计模板或基本结构，其目的是以会话形式提供方便的 GUI 支持。MVC 设计模式首先出现在 Smalltalk 编程语言中，MVC 是一种先进的组件设计模式，Java 在 MVC 的基础上设计了 Swing 组件。

MVC 是一种通过 3 个不同部分构造一个组件的理想办法：

WORD 版本.

- 1、 模型（**model**）用于存储定义该组件的数据。
- 2、 视图（**view**）用模型中的数据生成该组件的可视化表示。
- 3、 控制器（**controller**）处理用户对该组件的交互操作，对于用户的操作做出响应。

从面向对象的角度看，**MVC**结构可以使程序更具有对象化特性，也更容易维护。因为视图和控制器之间有依存关系，**Swing**组件的开发者们将**MVC**设计思想中的视图和控制器合并成一个独立整体，称为**Swing**组件的**UI代表**（**UI delegate**），如图1所示。

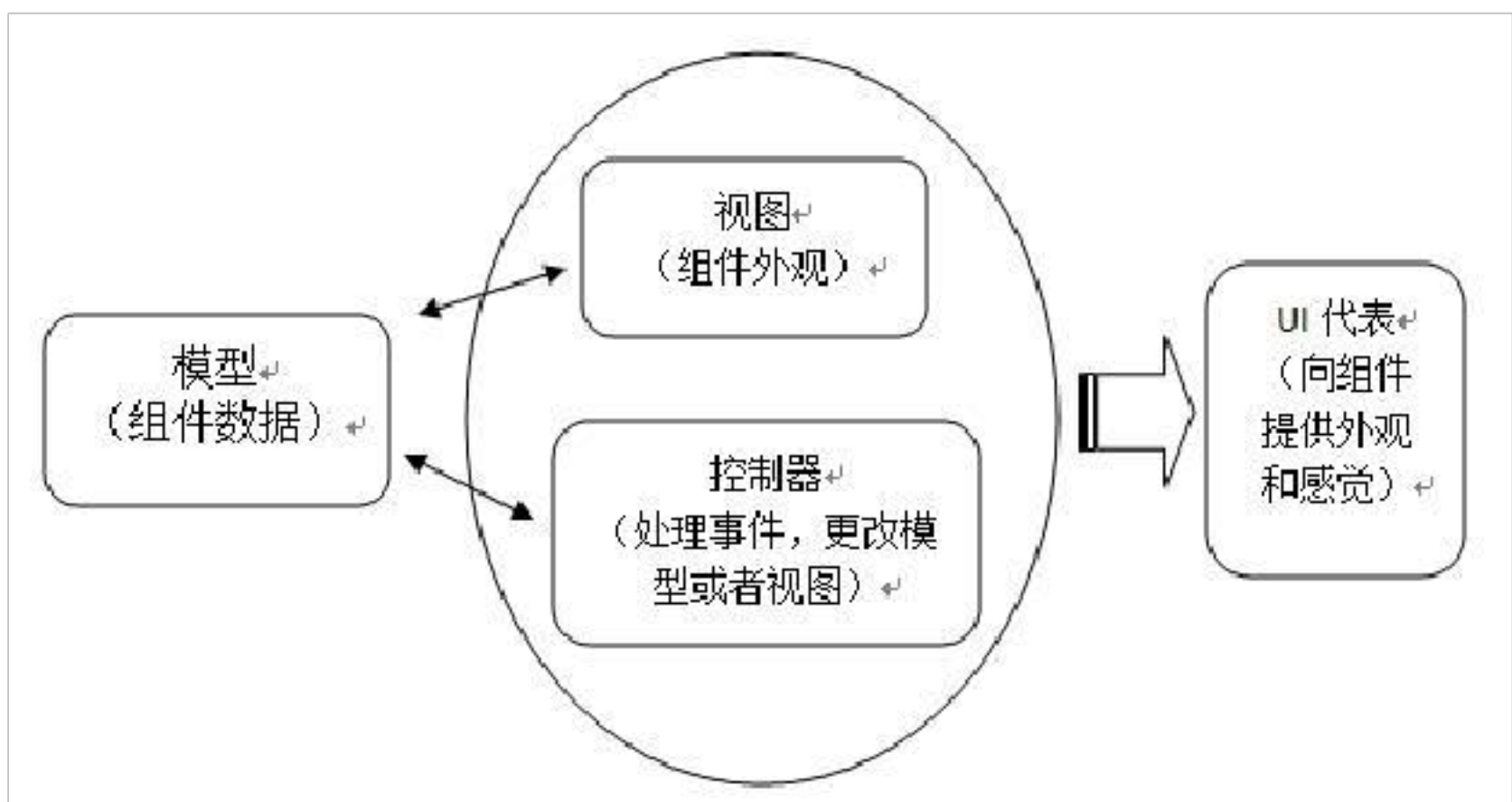


图 1 MVC结构示意图

### 1.2.3 Swing 与 AWT的关系

Java 的早期版本以来，抽象视窗工具包为用户界面组件提供了平台独立的API。在AWT中，每个组件都由一个原生的等同组件生成和控制，这个原生组件是当前的图形视窗系统决定的。与此相对，**Swing**组件经常被描述为轻量级的，因为它们不需要操作系统本身所带视窗工具包的原生资源来生成。**Swing API**的大部分

WORD版本.

---

AWT的补充扩展而不是直接的代替。Swing 用来绘制轻量级组件的核心渲染功能是由 Java2D 提供的，这是 AWT的一部分。然而，轻量级和重量级组件在同一个应用中使用会导致 Z-order 不兼容。

#### 1.2.4 创建图形界面步骤

- (1) 导入 Swing 包
- (2) 选择界面风格
- (3) 设置顶层容器
- (4) 设置按钮和标签
- (5) 将组件放到容器上
- (6) 为组件增加边框
- (7) 处理事件
- (8) 辅助技术支持

### 1.3 需求概述

求职者在寻找工作之前，都会花费大量的时间用来建立简历，这样缩短了寻找工作的时间，降低了找工作的效率。简历自动生成程序，可以帮助求职者花费少量的时间，就能建立一份完整的简历，提高他们找工作的效率。

---

## 简历生成程序的功能要求

### (1) 信息输入

由求职者将信息输入到程序中，这是本程序操作的第一步，也是后续步骤的基础。本程序只有一个界面，可使求职者更直观、更方便的操作。

### (2) 信息预览

将输入的信息整合起来，显示在预览框，使求职者查看输入到本程序中的信息。

### (3) 信息保存

将输入到本程序中的信息，以txt 格式保存到计算机上。

### (4) 重置信息

清除已输入到本程序中的信息，使求职者可以从新输入信息。

## 1.5 运行环境

运行服务器：JDK1.5 版本或更高

操作系统：windows 系列，Linux 系列或 Unix 系列

硬件需求：VGA显示器或液晶显示器，存最低为 256MB 硬盘空间最少为 20G

键盘及鼠标。

---

## 、环境搭建

### 2.1 安装 JDK

- (1) 打开“<http://java.sun/>”网页，根据本地计算机的操作系统，选择并下载适合此系统的 JDK 版本。
- (2) 找到安装程序后，双击 `jdk-6-windows-i586.exe`，运行安装程序进行安装。
- (3) 等程序初始化完成后，选择接受许可证协议，继续安装。
- (4) 选择要安装的组件，点击“下一步”按钮继续安装，一般选择默认组件直接进入下一步。
- (5) 等待程序安装完成后，显示安装完成界面，至此 JDK 安装完成。

### 2.2 设置环境变量

- (1) 在桌面右键点击“我的电脑”图标，在出现的菜单中选择“属性”选项，出现的界面如图 2 所示。

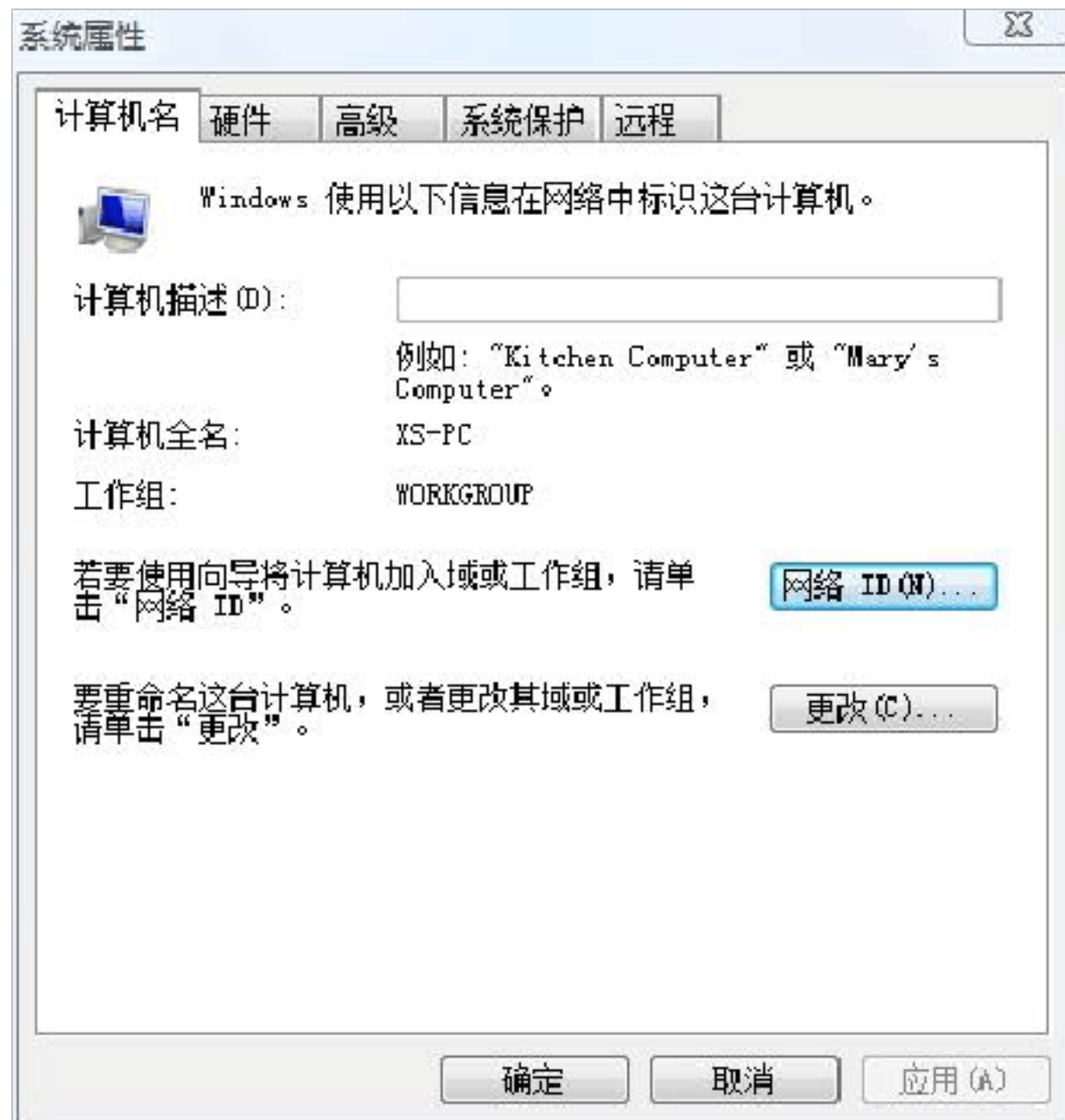


图 2 系统属性

(2) 选择“高级”选项，如图 3 所示，并单击“环境变量”按钮，弹出“环境变量”对话框，如图 3 所示。



图3 “高级”界面



图4 “环境变量”对话框

3) 在“系统变量”对话框里找到变量名“Path”，并选出，再单击“编辑”按钮进入“编辑系统变量”对话框，如图5所示。



图5 “编辑系统变量”对话框

(4) 在所有的变量值前面输入“”，注意有分号，路径根据情况进行设置。

(5) 在“系统变量”对话框里点击“新建”按钮，弹出“新建系统变量”对话框，如图6所示。

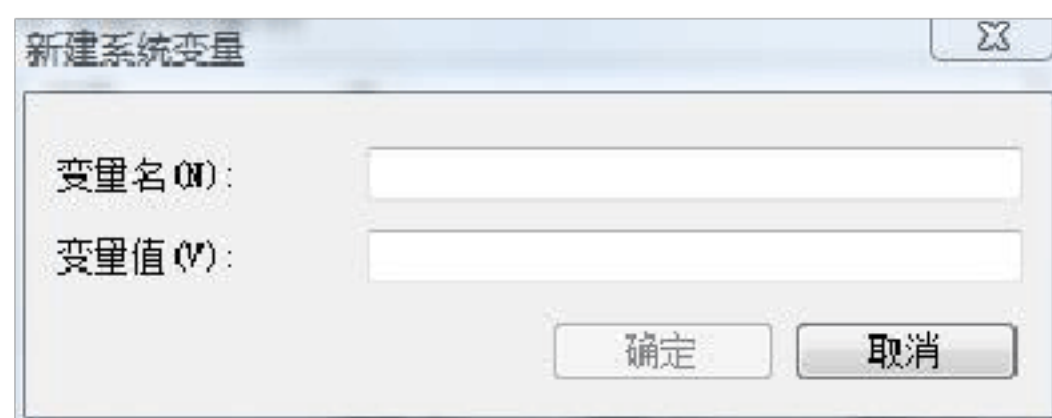


图6 “新建系统变量”对话框

(6) 设置变量名为“classpath”，变量值为

“

”，路径根据情况进行设置，如图7所示。

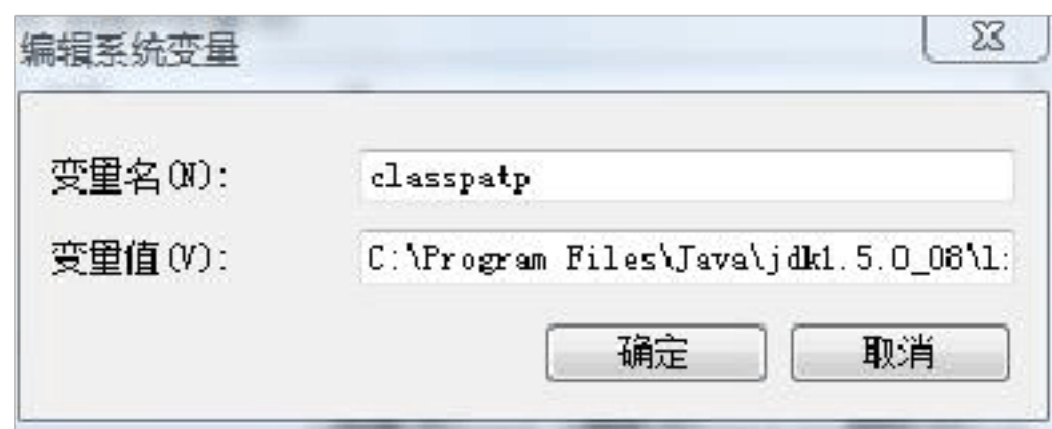


图7 填入新值

---

配置完成后，需要测试是否配置正确。下面给出了详细的测试步骤，按照这个步骤操作可以很轻松地完成测试。

(1) 单击“开始”按钮，在弹出的开始菜单中再单击“运行”菜单项，将出“运行”对话框。在“运行”对话框里输入“cmd”，并单击“确定”按钮，如图 8 所示。

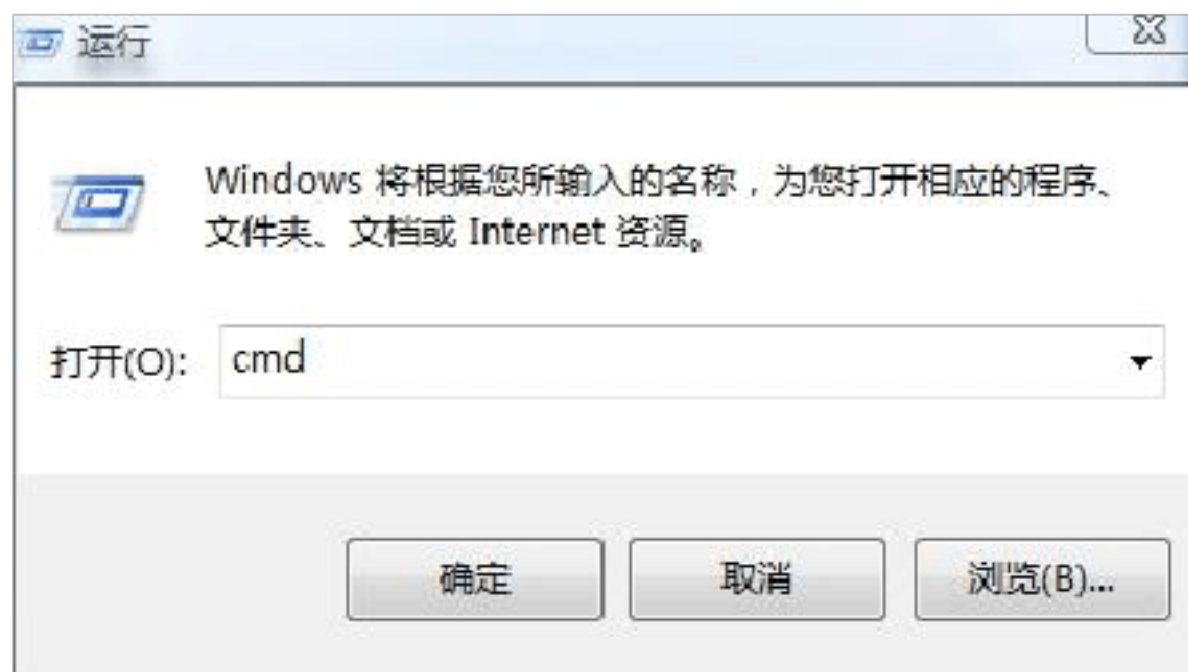


图 8 “运行”对话框

(2) 在命令提示符窗口里输入“javac”和“java”，观察是否出现 javac 或 java 的用法提示，如图 9 所示。

```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.0.6001]
版权所有 (C) 2006 Microsoft Corporation。保留所有权利。

C:\Users\XS>javac
用法: javac <选项> <源文件>
其中, 可能的选项包括,
-g 生成所有调试信息
-g:none 不生成任何调试信息
-g:<lines,vars,source> 只生成某些调试信息
-nowarn 不生成任何警告
-verbose 输出有关编译器正在执行的操作的消息
-deprecation 输出使用已过时的 API 的源位置
-classpath <路径> 指定查找用户类文件和注释处理程序的位置
-cp <路径> 指定查找用户类文件和注释处理程序的位置
-sourcepath <路径> 指定查找输入源文件的位置
-bootclasspath <路径> 覆盖引导类文件的位置
-extdirs <目录> 覆盖安装的扩展目录的位置
-endorseddirs <目录> 覆盖签名的标准路径的位置
-proc:<none,only> 控制是否执行注释处理和/或编译。
-processor <class1>[,<class2>,<class3>... ] 要运行的注释处理程序的名称; 绕过默认
的搜索进程
-processorpath <路径> 指定查找注释处理程序的位置
-d <目录> 指定存放生成的类文件的位置
-s <目录> 指定存放生成的源文件的位置
-implicit:<none,class> 指定是否为隐式引用文件生成类文件
-encoding <编码> 指定源文件使用的字符编码
-source <版本> 提供与指定版本的源兼容性
-target <版本> 生成特定 VM 版本的类文件
-version 版本信息
-help 输出标准选项的提要
-Akey[=value] 传递给注释处理程序的选项
-X 输出非标准选项的提要
-J<标志> 直接将 <标志> 传递给运行时系统

C:\Users\XS>

```

图 9 配置正确的环境变量信息

## 2.3 HelloWorld 程序

### 2.3.1 编写源程序

打开文本文件编辑器, 如 Windows 的记事本, 也可以使用更高级的编写工具。如 Eclipse、JBuilder、NetBeans 等, 这些工具具有更加强大的功能, 但不推荐初学者使用, 不利于初学者打下良好的编程基础。首先, 在记事本里添加如下代码。

例 1,

WORD 版本.

---

```
//定义一个类名称为 HelloWorld

Public class HelloWorld

{

    //类的主入口函数

    Public static void main (String args[])

    {

        //System.out.println    为打印语句，用来显示结果

        System.out.println(    “ 欢迎使用 Java 来编写程序! ” );

    }

}
```

在本程序中，首先定义了一个类，类的名字为 **HelloWorld**。在这个类里有一个 **main** 方法，这是 **Java** 程序的入口，只要能执行的程序都有这个方法。

**System.out.println** 方法能执行打印操作，还能打印其它类型的数据。

在编写后把这个文本文件保存为 **HelloWorld.java**，并注意大小写问题。

### 2.3.2 编译程序代码并运行

编写完 **Java** 程序的源代码后就可以对该程序进行编译，**Java** 程序源代码的编译有如下几个步骤。

(1) 单击“开始按钮”，在菜单中选择“运行”菜单项，将出现“运行”对话框。

在“运行”对话框里填写“cmd”，并单击“确定”按钮。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/635022204344011321>