

高等院校规划教材

工业机器人运动仿真编程实践

——基于Android和OpenGL

郇极 著



机械工业出版社
CHINA MACHINE PRESS



高等院校规划教材

工业机器人运动仿真编程实践 ——基于 Android 和 OpenGL

邰 极 著



机械工业出版社

本书介绍使用 Java 语言和 OpenGL 图形编程工具开发 Android 操作系统下工业机器人仿真软件的编程方法和程序实例。学习完本书,读者可以自己编写出一个简单的工业机器人运动仿真程序,使其在 Android 操作系统的手机和平板电脑上运行。

本书对使用 OpenGL 编程工具编写 Android 操作系统下其他图形显示软件也具有参考作用,适合作为各类高等院校相关专业学生、工程技术人员及编程爱好者自学 Android 操作系统下 OpenGL 编程技术的入门教材。

图书在版编目(CIP)数据

工业机器人运动仿真编程实践:基于 Android 和 OpenGL/郇极著. —北京:机械工业出版社, 2018.5

高等院校规划教材

ISBN 978-7-111-59258-7

I. ①工… II. ①郇… III. ①工业机器人—软件仿真—高等学校—教材
②工业机器人—程序设计—高等学校—教材 IV. ①TP242.2

中国版本图书馆 CIP 数据核字(2018)第 038367 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:李文轶 责任编辑:李文轶

责任校对:张艳霞 责任印制:李 昂

北京宝昌彩色印刷有限公司印刷

2018 年 6 月第 1 版·第 1 次印刷

184mm×260mm·8.5 印张·203 千字

0001—3000 册

标准书号:ISBN 978-7-111-59258-7

定价:29.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:(010) 88379833

机工官网:www.cmpbook.com

读者购书热线:(010) 88379649

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面无防伪标均为盗版

金书网:www.golden-book.com

前 言

本书介绍使用 Java 语言和 OpenGL 图形编程工具开发 Android 操作系统下工业机器人仿真软件的编程方法和程序实例,可以帮助读者学习 Android 操作系统下工业机器人仿真软件开发技术以及 OpenGL 编程技术。

平板电脑 (Tablet Personal Computer) 是目前广泛使用的移动多媒体设备。由于其具有丰富的显示、操作、计算和互联网功能,因此逐渐开始被应用于工业化自动控制领域。平板电脑既可以作为具有控制、操作和显示功能的全功能控制器,也可以作为主控制装置的操作和显示单元使用。控制过程的运动图形仿真能为控制系统操作和调试提供很大的方便,目前已经成为许多控制装置的必要功能和基本要求。

作者于 2016 年在北京航空航天大学数控和伺服技术实验室网站上 (www.nc-servo.com) 发布了用于工业机器人教学和编程练习的虚拟工业机器人控制系统程序 (PAD ROBOT)。它是运行在 Android 操作系统平板电脑或手机上的虚拟工业机器人控制系统程序,能够为使用者提供一个便捷、安全的工业机器人运动控制编程练习。使用者可以在平板电脑或手机上编写自己的机器人运动控制程序,然后在该系统上运行,从而学习工业机器人编程技术。该系统具有工业机器人控制系统的基本功能,包括运动控制程序创建、程序译码、关节坐标系运动控制、直角坐标系运动控制、工具姿态控制、直线和圆弧插补、工具补偿、工具更换、运行操作。

虚拟工业机器人控制系统程序 (PAD ROBOT) 是作者使用 Java 语言和 OpenGL 图形编程工具编写的工业机器人运动仿真应用程序。本书将作者开发控制系统 PAD ROBOT 时所编写的典型图形和动画编程程序介绍给读者,帮助读者学习工业机器人仿真软件开发技术及 Android 操作系统下的 OpenGL 编程技术。

本书的主要内容如下:

- 第 1 章为概述,介绍了用平板电脑进行工业机器人仿真编程的意义和本书特色。
- 第 2 章介绍了作者开发的两个机器人运动仿真样机,读者可以把它们安装在平板电脑和手机上,完成操作演示。
- 第 3 章概要介绍了 Java 编程语言,给出了与本书后续内容相关的典型编程示例。
- 第 4 章概要介绍了基于 Android 操作系统的开发环境,给出了与本书后续内容相关的典型编程示例。
- 第 5 章以机器人运动仿真样机的基座部件为例,介绍应用 Android 操作系统和 OpenGL 编程的基本方法,完成一个在平板电脑或手机上运行的机器人运动仿真样机基座部件的 3D 显示示例。
- 第 6 章以机器人运动仿真样机程序为例,完成机器人全部部件的 3D 建模和显示。
- 第 7 章介绍机器人运动仿真操作界面的编程方法和编程示例。
- 第 8 章介绍机器人运动控制的编程方法和编程示例,最终完成机器人的运动演示。

附录 1~7 为本书示例程序的源程序文档。

按照本书的内容顺序和示例程序，读者可以自己编写出一个简单的工业机器人运动仿真程序，在 Android 操作系统下的手机和平板电脑上运行。本书内容对使用 OpenGL 编程工具编写 Android 操作系统下的其他图形显示软件也具有参考作用。

本书配备的电子资源有：PAD ROBOT 安装程序、GL ROBOT 安装程序、_surfaced 类的 Java 源程序、GL_CONST 类的 Java 源程序、ROB_PAR 类的 Java 源程序、JOINT 类的 Java 源程序。读者可在 www.cmpedu.com 上下载。

书中的疏漏和不足之处敬请指导。

作者

目 录

前言	
第 1 章 概述	1
1.1 开发和编程环境	1
1.2 OpenGL 图形编程工具	2
1.3 本书特点	2
第 2 章 工业机器人运动仿真样机介绍	3
2.1 机器人运动仿真应用程序 PAD ROBOT	3
2.2 机器人运动仿真样机 GL ROBOT	4
第 3 章 Java 编程语言	6
3.1 Java 程序设计	6
3.1.1 Java 语言特点	6
3.1.2 开发环境	7
3.2 Java 语言基础	7
3.2.1 Java 程序的符号集	7
3.2.2 Java 程序的基本组成	8
3.2.3 常量与变量	9
3.2.4 数据类型	9
3.2.5 运算符和表达式	10
3.2.6 控制语句	11
3.3 示例程序设计的 Java 语法要点	12
3.3.1 类和对象	12
3.3.2 枚举类型	13
3.3.3 数组	14
3.3.4 String 类	14
3.3.5 异常处理	15
3.3.6 包的应用	16
3.3.7 数学运算	16
第 4 章 Android 应用程序开发环境	18
4.1 Android 开发概述	18
4.1.1 Android 系统框架	18
4.1.2 Android 应用程序开发环境的搭建	19
4.1.3 Android 工程的结构和运行	21
4.2 基于 Android 开发机器人运动仿真软件	23
4.2.1 Activity 和视图布局	23
4.2.2 定时器	26
第 5 章 OpenGL 编程示例	28
5.1 安装 Android 应用程序开发工具 Eclipse	28
5.2 物体表面定义	28
5.3 GL ROBOT 基座 part0 的数据结构	29
5.4 创建 GL ROBOT 基座的顶点和法向量数组	30
5.5 创建显示界面	35
5.6 创建渲染器接口类	36
5.6.1 onSurfaceCreated() 方法	37
5.6.2 onSurfaceChanged() 方法	38
5.6.3 onDrawFrame() 方法	39
5.7 创建视图类	41
5.8 编写主程序	42
5.9 运行示例程序	42
第 6 章 GL ROBOT 部件编程	44
6.1 部件 part1	44

6.1.1 结构	44	7.1 操作界面布局	68
6.1.2 构建转台的顶点和法向量数组	45	7.2 操作按钮编程	71
6.1.3 构建立柱的顶点和法向量数组	51	7.2.1 按钮 A0~A5 编程	71
6.1.4 显示	53	7.2.2 按钮 move 和 stop 编程	73
6.2 部件 part2	54	7.2.3 按钮 F%编程	75
6.2.1 结构	54	7.3 位置、速度和定时器计数显示 编程	76
6.2.2 构建关节的顶点和法向量数组	54	7.4 运行操作界面	77
6.2.3 构建摆杆的顶点和法向量数组	55	第 8 章 GLROBOT 运动仿真编程	78
6.2.4 显示	57	8.1 定时器编程	78
6.3 部件 part3	58	8.2 运动控制编程	79
6.3.1 结构	58	8.2.1 添加全局静态变量	79
6.3.2 构建圆柱的顶点和法向量 数组	58	8.2.2 关节位置计算	79
6.3.3 显示	59	8.2.3 JOINT 变量赋值	80
6.4 部件 part4	60	8.3 运动和显示编程	81
6.4.1 结构	60	附录	83
6.4.2 构建圆柱体侧面顶点和法向量 数组	60	附录 1 _surfaced 类的源程序	83
6.4.3 显示	61	附录 2 GL_CONST 类和 ROB_PAR 类的源程序	92
6.5 部件 part5	62	附录 3 _part 0~_part 6 类的 源程序	93
6.5.1 结构	62	附录 4 activity_main.xml 的 源程序	110
6.5.2 构建摆杆和关节的顶点和法 向量数组	62	附录 5 渲染器接口 viewRenderer 及其 onDrawFrame()方法 的源程序	112
6.5.3 显示	63	附录 6 视图类 MyGLView 的 源程序	119
6.6 部件 part6	64	附录 7 MainActivity 的源程序	120
6.6.1 结构	64	参考文献	130
6.6.2 构建转盘和卡爪的顶点和法 向量数组	64		
6.6.3 显示	66		
第 7 章 GLROBOT 操作界面编程	68		

第1章 概述

工业机器人运动仿真是指机器人控制系统或仿真系统将机器人的控制程序转换成可视的动作过程的动画，将机器人的运动情况在显示器上直观地显示出来。在机器人使用领域，工业机器人运动仿真技术能够用于离线编程控制程序的检查，例如运动路径、速度、碰撞等，或用于工作现场其运行状态显示。在机器人控制软件开发领域，该技术可以用于控制软件功能和控制参数的测试。在机器人结构设计领域，可以用于机器人结构参数的检查和优化，例如，结构布局和工作空间的检查和优化等。操作人员培训也是工业机器人运动仿真技术的一个重要应用领域。操作人员可以在仿真系统上练习操作和编程，提高培训效率，避免操作事故。

平板电脑 (Tablet Personal Computer) 由于其具有丰富的显示、操作、计算和互联网功能，因此逐渐开始被应用于工业化自动控制领域。平板电脑既可以作为具有控制、操作和显示功能的全功能控制器，也可以作为主控制装置的操作和显示单元使用。控制过程的运动仿真能为控制系统操作和调试提供很大的方便，目前已经成为许多控制装置的必要功能和基本要求。

作者于 2016 年在北京航空航天大学数控和伺服技术实验室网站上 (www.nc-servo.com) 发布了用于工业机器人教学和编程练习的虚拟工业机器人控制系统程序 (PAD ROBOT)。它是运行在具有 Android 操作系统的平板电脑或手机上的虚拟工业机器人控制系统程序，为用户提供了一个便捷、安全的工业机器人运动控制的编程练习环境。使用者可以在平板电脑或手机上编写自己的机器人运动控制程序，在该系统中运行来学习工业机器人编程技术。该系统具有工业机器人控制系统的基本功能，包括运动控制程序创建、程序译码、关节坐标系运动控制、直角坐标系运动控制、工具姿态控制、直线和圆弧插补、工具补偿、工具更换、运行操作。

虚拟工业机器人控制系统程序 (PAD ROBOT) 是作者使用 Java 语言和 OpenGL 图形编程工具编写的工业机器人运动仿真应用程序。作者提取出开发 PAD ROBOT 控制系统时的部分图形和动画程序并将其作为编程示例介绍给读者，帮助读者学习工业机器人仿真软件开发技术及 Android 操作系统下的 OpenGL 编程技术。

通过本书的学习，读者可以自己编写出一个工业机器人运动仿真程序，并在手机和平板电脑上运行。本书内容对使用 OpenGL 编程工具编写 Android 操作系统下其他图形显示软件也具有参考作用。

1.1 开发和编程环境

本书所介绍的工业机器人运动仿真编程方法和编程示例是使用 Android (安卓) 应用程序开发工具 Eclipse 和 Java 编程语言编写的，因此可以直接在 Android 平板电脑或手机上运

行。Google 公司提供了一个 Android 集成开发环境，下载网址为 <http://developer.android.com/sdk/index.html>^①。本书编程示例所使用的安装版本为 jdk-6u22-windows-i586.exe。若将该程序安装在 Windows 操作系统中后，可以在 Windows 操作系统的 PC 上开发 Android 应用程序。若将 Android 安装程序下载到平板电脑或手机上，即可在平板电脑或手机上运行。

关于下载和安装 Android 集成开发环境的具体步骤及 Eclipse 相关操作，请读者参考其他相关书籍学习。

1.2 OpenGL 图形编程工具

OpenGL (Open Graphics Library) 是一个开放的图形程序接口，它为开发三维图形应用提供了功能强大的底层图形库。OpenGL 的应用时间已经超过 20 年，是目前使用最广泛的三维图形编程工具之一。

OpenGL ES (OpenGL for Embedded Systems) 是对 OpenGL 裁剪产生的子集，适用于手机、平板电脑、游戏机、家电设备、车载设备等。Android 2.2 以后的版本开始支持 OpenGL ES。

Android 应用程序开发工具 Eclipse 集成了 OpenGL ES 编程接口，本书所提供的图形编程示例都是用 Java 语言和 OpenGL ES 接口完成的。

1.3 本书特点

本书作者于 2016 年在北京航空航天大学数控和伺服技术实验室网站上 (www.nc-servo.com) 发布了 PAD ROBOT 的 Android 应用程序 APK。它是一个 6 自由度关节式虚拟工业机器人的应用程序，在 Android 操作系统的平板电脑或手机上运行，用于工业机器人教学和编程练习。PAD ROBOT 的所有程序都是由本书作者编写的。

本书以 PAD ROBOT 的三维图形运动显示部分的程序为例，介绍使用 OpenGL 编写 6 自由度关节式工业机器人运动仿真程序的方法。所有示例程序均来自 PAD ROBOT 程序。为了突出介绍 OpenGL 工业机器人三维运动仿真编程方法，便于读者理解，删减了与 OpenGL 无关的部分程序，例如译码、插补、坐标变换、速度控制、自动屏幕尺寸适应、视图变换等。

本书的特点是用程序示例的方式介绍 OpenGL 工业机器人运动仿真控制的图形编程方法，在具体编程时，不对所调用的函数或参数做全面的解释，而只讲解在本书的应用。读者可以参考其他书籍来学习函数和参数调用的完整功能。

在开始使用本书示例学习 OpenGL 工业机器人图形编程之前，读者需要基本掌握 Android 应用程序开发工具 Eclipse 和 Java 编程语言。通过本书的学习读者可以自行编写出一个 6 自由度关节式工业机器人运动仿真程序，并在手机和平板电脑上运行。本书内容对使用 OpenGL 编程工具编写 Android 操作系统下其他图形显示软件也具有参考作用。

① 本书中涉及的该类网站可能因为被屏蔽的原因而打不开，读者可查阅相关信息自行解决。

第2章 工业机器人运动仿真样机介绍

本书的示例程序摘自作者编写的工业机器人（PAD ROBOT）运动仿真程序。本书随附的资源提供了 PAD ROBOT 安装程序，可以在 Android 手机或平板电脑上对其安装并运行。读者也可以从 www.nc-servo.com 网站下载该程序。为了便于学习基于 OpenGL 编写工业机器人运动仿真软件的基本方法，对 PAD ROBOT 程序做了简化，重点介绍使用 OpenGL 编写工业机器人 3D 图形显示程序的方法，删减了工业机器人控制技术方面的内容，例如数控程序译码、插补、坐标变换等程序。简化后的机器人运动仿真样机的程序名为 `gl_robot`，本书随附的资源提供了其安装程序。

2.1 机器人运动仿真应用程序 PAD ROBOT

虚拟工业机器人控制系统程序（PAD ROBOT）是运行在 Android 操作系统的平板电脑或手机上的，为使用者提供一个便捷、安全的工业机器人运动控制的编程练习。使用者可以在平板电脑或手机上编写自己的机器人运动控制程序，在该系统上运行以学习工业机器人编程技术。系统具有工业机器人控制系统的基本功能，包括运动控制程序创建、程序译码、关节坐标系运动控制、直角坐标系运动控制、工具姿态控制、直线和圆弧插补、工具补偿、工具更换、运行操作。

将随附资源中的 `pad_robot_k.apk` 安装程序复制到 U 盘并将其安装到 Android 手机或平板电脑。读者也可以从 www.nc-servo.com 网站下载该程序。程序启动后屏幕显示如图 2.1 所示的机器人。



图 2.1 PAD ROBOT 界面

PAD ROBOT 界面包含了 6 个控件、9 个演示程序、4 种工具，可以演示系统的基本功能和操作。演示操作步骤如下：

- 1) 按“启动”键，可以观看第 1 个演示程序运行；
- 2) 按“任务”键可以选择其他 8 个演示程序的运行；
- 3) 按“帮助”键可以了解系统的全部功能，以帮助编写出多种类型数控程序。

2.2 机器人运动仿真样机 GL ROBOT

为了便于读者学习 OpenGL 的基本编程技术，本书删减了 PAD ROBOT 中非 OpenGL 程序部分，例如译码、插补、坐标变换、速度控制、自动屏幕尺寸适应、视图变换等，形成了机器人运动仿真样机 GL ROBOT 来作为本书的编程示例程序。将随附资源中的 gl_robot.apk 安装程序复制到 U 盘并将其安装到 Android 手机或平板电脑，运行后的显示界面如图 2.2 所示。

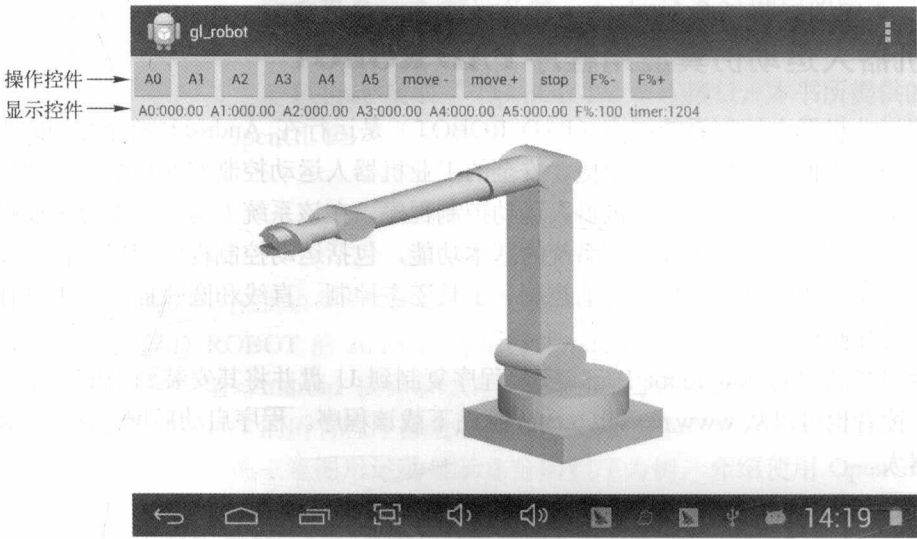


图 2.2 GL ROBOT 运行界面

界面中的操作控件功能如下。

- 1) A0~A5：选中运动的关节，可以多选和解除选中状态；
- 2) move+：启动正方向运动；
- 3) move-：启动负方向运动；
- 4) stop：停止运动；
- 5) F%+/F%-：用于运动速度的调整，是相对于示例程序中给定的最大速度（900°/min）的百分比。

界面中显示控件的含义如下。

- 1) A0~A5：关节位置（°）；
- 2) F%：运动速度调整的显示；

3) timer: 定时器的计数值。

图 2.3 所示是 GL ROBOT 的 6 个关节的名称。

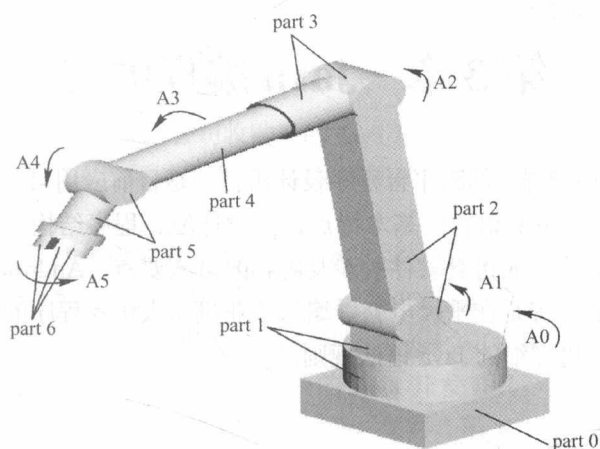


图 2.3 GL ROBOT 关节名称

操作示例如下。

- 1) 按 A0~A5 键，选中关节或解除选中的关节，可以多选；
- 2) 按 move+键，启动正方向运动；
- 3) 按 move-键，启动负方向运动；
- 4) 按 stop 键，停止运动；
- 5) 按 F%-键，减速；
- 6) 按 F%+键，加速。

第3章 Java 编程语言

Java 语言是一种功能强大的跨平台程序设计语言，是目前应用最为广泛的计算机编程语言之一。本章主要介绍 Java 语言的基本特点、开发环境、程序结构、数据类型、表达式和运算符，以及编写工业机器人仿真软件所涉及内容的语法要点。Android 应用程序采用 Java 语言编写，本章介绍的内容是在平板电脑中编写工业机器人仿真程序的基础，读者可以查阅其他相关参考书来了解更详细的 Java 语法规则。

3.1 Java 程序设计

3.1.1 Java 语言特点

Java 是一种面向对象的编程语言，它具有卓越的通用性、高效性、平台移植性和安全性，广泛应用于个人 PC、数据中心、游戏控制台、超级计算机、移动电话和互联网。Java 语言的特点有如下 5 个方面。

1. 面向对象

Java 语言是完全面向对象的语言。Java 提供了类的机制，在对象中封装了成员变量和方法，实现了数据的封装和信息隐藏。类通过继承和多态，实现了代码的复用。

2. 简洁有效

Java 语言省略了 C++ 语言中难以理解、容易混淆的特性，例如头文件、指针、结构、单元、运算符重载、虚拟基础类等。因此，Java 语言更加严谨、简洁。

3. 安全性

Java 语言摒弃了指针，一切对内存的访问都必须经过对象的实例变量来实现，阻止了以不法手段访问对象的私有成员，同时避免了指针操作中容易产生的问题。Java 语言的运行环境提供了字节码校验器、类装载器和文件访问限定功能等内部安全机制，保证了 Java 程序和系统资源的安全性。

4. 操作平台无关性

Java 程序在编译器中被转化成与平台无关的字节码指令，因此，相同的程序不需要更改就可以在各种操作系统上运行。平台无关的特性使得 Java 程序可以被方便地移植到不同的机器上。

5. 多线程

Java 是第一个在语言方面提供内置多线程支持的高级语言，这大大简化了多线程程序的编写。

3.1.2 开发环境

Java 程序的开发环境可以分为开发工具集（Java Development Kit, JDK）和集成开发工具（Integrated Development Environment, IDE）。

1. JDK

JDK 是 Sun 公司（已被 Oracle 公司收购）的 Java 程序开发工具集，它包括了 Java 运行环境、Java 工具和 Java 基础类库，可以免费从 Oracle 公司的网址（<http://www.oracle.com>）下载。

2. IDE

除了 JDK 以外，一些集成开发工具为人们提供了更为方便的交互式开发环境。广泛使用的 IDE 包括 Eclipse、NetBeans、JBuilder、Sun ONE Studio 5 和 IntelliJ IDEA。其中，Eclipse 是 IBM 公司开发的一个开放源代码的、基于 Java 的可扩展开发平台。Eclipse 附带了一个标准的插件集（包括 JDK），它是非常重要的 Java 开发工具。Eclipse 同样是 Android 应用程序的开发工具，本书示例程序全部使用 Eclipse 编写。第 4 章介绍了使用 Eclipse 搭建 Android 开发环境的方法。

3.2 Java 语言基础

3.2.1 Java 程序的符号集

1. 关键字

关键字也称为保留字，是系统预定义的具有专门意义和用途的符号。表 3.1 列出了 Java 语言的全部关键字。表中具有*标记的关键字被保留，表示当前尚未使用。

表 3.1 Java 关键字

首 字 母	关 键 字	首 字 母	关 键 字
a	abstract	e	else
	boolean		extends
b	break	f	false
	byte		final
	byvalue*		finally
c	case		float
	cast*		for
	catch		future*
	char	g	generic*
	class		goto*
	const*	i	if
	continue		implements
d	default		import
	do		inner*
	double		instanceof

(续)

首 字 母	关 键 字	首 字 母	关 键 字
i	int	s	static
	interface		super
l	long		switch
	native		synchronized
n	new	t	this
	null		throw
o	operator*		throws
	outer*		transient
p	package		true
	private		try
	protected	v	var*
	public		void
r	rest*		volatile
	return	w	while
s	short		

2. 标识符

Java 中的包、类、方法、参数和变量的名称，可以由任意的大（小）写字母、数字、下画线 “_” 和符号 “\$” 组成。标识符不能以数字开头，也不允许使用 Java 中的关键字。以下是标识符示例：

```
Student UserNames_style $money val12
```

3. 注释

为程序添加注释可以解释程序中某些语句的作用和功能，提高程序的可读性。Java 的注释可以分为以下 3 种类型。

- 1) 单行注释，其形式为 “//+注释内容”。表示从双斜线 “//” 开始直到此行末尾的部分为注释。
- 2) 多行注释，其形式为 “/*注释内容*/”。表示从 “/*” 开始，直到 “*/” 结束的部分为注释。
- 3) 文档注释，其形式为 “/**注释内容*/”。表示从 “/**” 开始，直到 “*/” 结束的部分为注释。用这种方式注释的内容会被作为正式文档而被囊括到 javadoc 等工具生成的文档里。

3.2.2 Java 程序的基本组成

由 Java 的各种符号可以构成 Java 应用程序。本小节通过一个简单的程序说明 Java 应用程序的基本结构，该示例程序的功能是在屏幕上显示字符串 “Hello world!”，代码如下所示：

```
public class JavaHelloWorld{           //类定义
    public static void main(String args[]){ //定义 main 方法
```

```
System.out.println("Hello world!"); //系统标准输出方法
```

```
}
```

```
}
```

1. 分隔符

示例程序中使用的分隔符包括回车符“Enter”、空格符、制表符、分号“;”和大括号“{}”。其中，大括号表示类和方法的开始与结束，程序中的大括号的数目必须要成对匹配。

2. 类定义

Java 程序都是由类组成的。示例程序第 1 行定义了一个名称为 `JavaHelloWorld` 的类。关键字 `class` 是类的标志；`public` 是用来修饰 `class` 的，说明该类是公共类。`class` 语句后面是一对大括号，其中的内容就是类的成员。本示例中为该类定义了一个 `main()` 方法。

3. `main()` 方法

示例程序的第 2 行定义了 `main()` 方法。它是 Java 程序的执行入口。含有 `main()` 方法的类称为主类。一个 Java 程序中只能包含一个主类。关键字 `static` 表示 `main()` 方法是静态方法，`void` 表示方法无返回值，`String args[]` 是方法的参数。`Main()` 方法声明语句后是一对大括号，其中的内容就是方法的主体。

4. 方法主体

示例程序第 3 行是 `main()` 方法的主体，它调用了系统标准输出方法 `System.out.println()`，向屏幕输出字符串“Hello world!”。

3.2.3 常量与变量

常量是固定不变的量，一旦被定义，它的值就不能再被改变。常量名称通常使用大写字母表示，但这不是硬性要求。常量使用 `final` 修饰符进行声明，以下是常量的声明示例：

```
final int MAX_AXIS = 127;
```

```
final double PI = 3.1415926;
```

变量可为指定的内存空间命名，它的值可以被改变。变量的作用域是指可以访问该变量的程序代码范围。按照作用域的不同，变量可以分为类成员变量和局部变量。类成员变量在类的声明体中声明，它的作用域为整个类；局部变量在方法体或者方法的代码块中声明，它的作用域为它所在的代码块。变量的名称遵循标识符的命名规则，以下是变量的声明示例：

```
float feed_next_block;
```

```
int g0123, g01789;
```

3.2.4 数据类型

基本数据类型是指 Java 语言固有的数据类型，可以分为整数类型、浮点类型、字符型和布尔型。Java 语言的基本数据类型说明如表 3.2 所列。

表 3.2 Java 语言的基本数据类型说明

名 称	功 能	关 键 字	长度/bit	范 围	默 认 值
整数类型	字节型	byte	8	$-2^7 \sim 2^7 - 1$	0
	短整型	short	16	$-2^{15} \sim 2^{15} - 1$	0
	整型	int	32	$-2^{31} \sim 2^{31} - 1$	0
	长整型	long	64	$-2^{63} \sim 2^{63} - 1$	0
浮点类型	浮点型	float	32	$-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$	0.0F
	双精度型	double	64	$-1.7 \times 10^{308} \sim 1.7 \times 10^{308}$	0.0D
字符型		char	16	0~65535 (Unicode 符号)	\u0000
布尔型		boolean	8	true、false	false

Java 语言有严格的数据类型限制。数据类型的转换方式可以分为隐式转换及强制转换。隐式转换分为两种情况：第一，在赋值操作时，如果将较短类型的数据赋给较长类型，则类型转换由编译系统自动完成；第二，在计算过程中，如果一个较短类型的数据与较长类型的数据进行运算，系统会自动把较短类型的数据转换成较长类型的数据，再进行运算。以下是隐式数据类型转换的示例：

```
double x = 100; //整型数据 100 被隐式转化成 double 类型
```

强制转换的语法格式和示例如下：

```
(数据类型) 表达式  
int result = (int)2.45; //浮点型数据 2.45 被强制转化成 int 类型，值为 2
```

3.2.5 运算符和表达式

运算符是执行数学和逻辑运算的标识符。Java 语言的运算符非常丰富。表达式是由常量、变量或其他操作符与运算符所组合而成的语句。表达式是程序组成的基本部分。表 3.3 列举了 Java 语言运算符的优先级、类型，并给出了对应的表达式示例。

表 3.3 Java 语言的运算符和表达式

优先级	类型名称	运 算 符	表达式示例
1	结合运算	()	(a + b) / c
	数组变量标识符	[]	inp.ax.pos[1]
	引用	.	
2	逻辑否	!	!value
	正号、负号	+, -	-i
	按位取反	~	~value
	递增、递减	++, --	i++
3	乘、除、取余	*, /, %	a/b c*5 d%10
4	加、减	+, -	e + f
5	位左移、位右移	<<, >>	value << 8