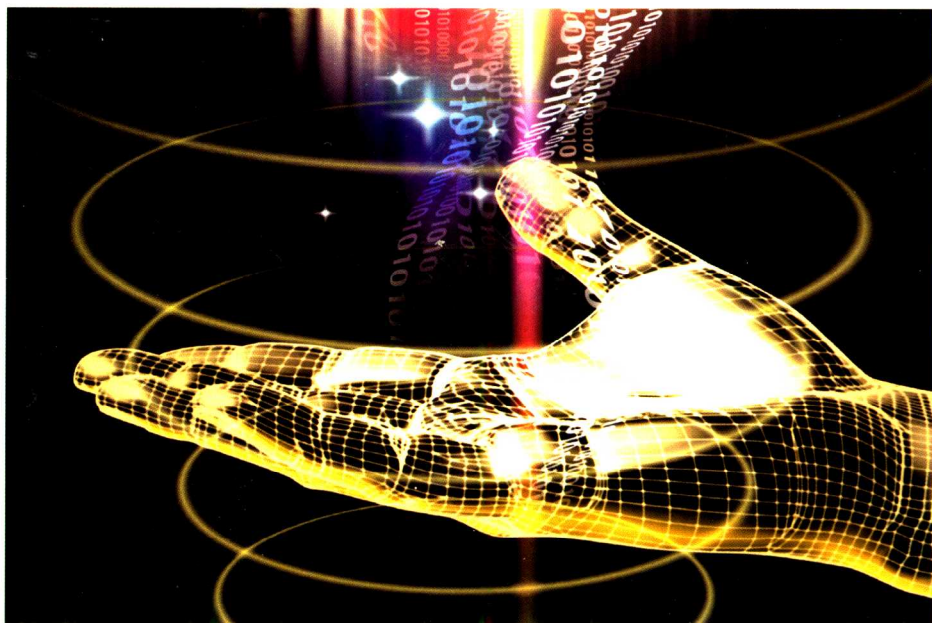


Java 程序设计

课程设计

辛运伟 高 宁 徐文拴 等编著



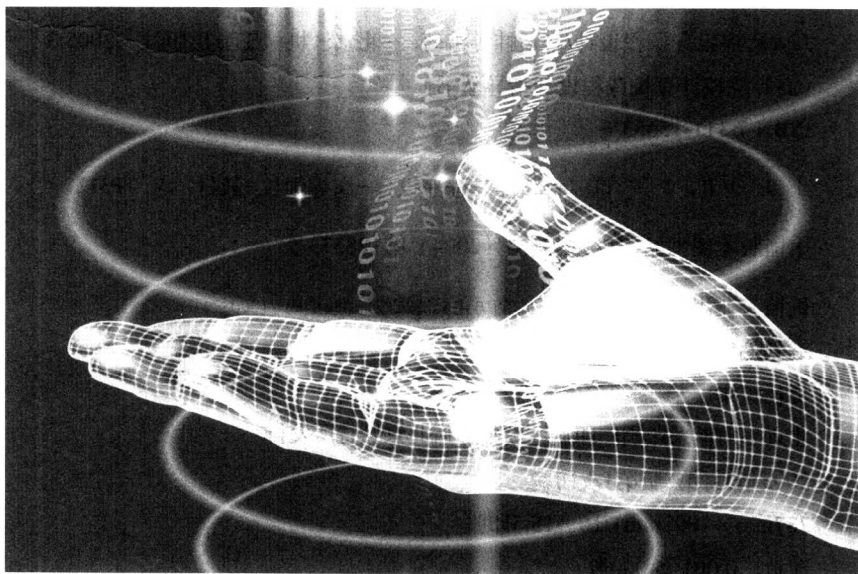
机械工业出版社
China Machine Press

高等院校计算机课程设计指导丛书

Java程序设计

课程设计

辛运伟 高宁 徐文拴 等编著



机械工业出版社
China Machine Press

本书是配合Java程序设计课程的课程设计指导教材，独立于任何具体的Java语言教科书，重点放在Java语言的基本特征上，涵盖Java语言的重要基础知识。通过详细的实例，循序渐进地启发学生完成课程设计，培养学生实际分析问题、编程和动手的能力，提高学生综合素质。

本书适合作为高等院校计算机专业的Java语言课程设计指导用书，也可供广大计算机编程人员参考。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目（CIP）数据

Java程序设计课程设计/辛运韩等编著. —北京：机械工业出版社，2005.3

（高等院校计算机课程设计指导丛书）

ISBN 7-111-15653-6

I. J… II. 辛… III. Java语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字（2004）第128333号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

策划编辑：温莉芳

责任编辑：华 章

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2005年3月第1版第1次印刷

787mm × 1092mm 1/16 · 15.5印张

印数：0 001- 5 000册

定价：23.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

丛书序言

近年来,我国在计算机应用、计算机软件和电子类相关专业的人才培养方面,取得了长足的进展,每年的毕业生都有数十万人。但是这些毕业生走进企业、公司、政府机构或研究单位之后,往往深刻地感觉到缺乏实际开发设计项目的经验,不善于综合运用所学理论,对知识的把握缺乏融会贯通的能力。

综合考察目前高等院校教学大纲、课程设置以及内容安排等方面的情况,多数学校还是比较重视训练学生的实际设计能力。但是,从安排设计实践的内容上看,基本上是围绕相关课程教学内容而展开的,不能够构成对实际问题的解决方案;从配套程序的规模上看,一般只是几十行到几百行的源代码,或者是一个单独电路的设计,远远小于一个小型项目的规模;从设计的结构上看,由于设计实践是围绕着课程教学内容而进行的,问题已经高度抽象,学生很难得到有关综合运用所学知识的整体训练机会。而且,这些内容相对简单、问题域已经高度抽象、规模较小的设计实践一人基本上就能完成,学生几乎无法通过这些设计实践,去真正获得有关项目管理和团队协作等方面的基本训练和工作经验。

由此可以看出,大多数学校对学生实际设计能力的训练与国外知名大学和国内精品课程相比较,还是存在一些差距的。为此,机械工业出版社华章分社和一批高等院校的教师,针对当前高等院校计算机硬件、软件和电子类相关课程教学中存在的问题,参考国内外知名大学相关课程成功的教学经验,设计编写了这套“高等院校计算机课程设计指导丛书”,其目的就是通过课程设计的一系列训练,把知识获取和项目实践两个方面有机地结合起来。

在这套“高等院校计算机课程设计指导丛书”中的每一门课程设计里,都安排了由多个子项目组成的一个课程设计项目。学生们可以在教师的指导下,逐步设计实现这些子项目,并最终完成一个功能相对完整,可以运行的系统,其代码可以是数千行,甚至上万行。通过这种设计课程,学生一方面可以结合课程的教学内容循序渐进地进行设计方面的实践训练,另一方面,在参与一系列子项目的实践过程中,还能提高如何综合运用所学知识解决实际问题的能力,以及获得有关项目管理和团队合作等等众多方面的具体经验,增强对相关课程具体内容的理解和掌握能力,培养对整体课程知识综合运用和融会贯通能力。

参加丛书编写的各高等院校的教师都有着丰富的教学、科研,以及与企业合作开发项目等多方面的经验。每个课程设计中的子项目和整体项目,都来自教师们具体的科研和设计开发实践,所选设计项目与教学内容配合紧密,项目的难度与规模适宜。

最后,感谢机械工业出版社华章分社编辑们的大力支持,使出版有关这套丛书的计划,从单纯的构想演化成带有油墨芳香的真实。

丛书写作组

高等院校 计算机课程设计指导丛书

专家指导委员会

(以姓氏拼音为序)

- | | |
|-----|------------|
| 陈向群 | (北京大学) |
| 戴 葵 | (国防科技大学) |
| 何钦铭 | (浙江大学) |
| 林 闯 | (清华大学) |
| 刘振安 | (中国科技大学) |
| 马殿富 | (北京航空航天大学) |
| 齐 勇 | (西安交通大学) |
| 宋方敏 | (南京大学) |
| 汤 庸 | (中山大学) |
| 王立福 | (北京大学) |
| 吴功宜 | (南开大学) |
| 赵一鸣 | (复旦大学) |

联络人 温莉芳

前言

计算机教学的改进虽然没有严格遵从著名的摩尔定律，但其变化的速度与其他学科相比仍处于领先地位。不论是教学形式、教学手段、教学内容，乃至课程体系在近些年都有很大的变化。翻开高等院校计算机专业学生的教学大纲，比较一下20年前后学生的学习课程目录，这些变化跃然纸上。即使是名称相同的课程，其内容的差异也与日俱增。可喜的是，国内教材的编写和出版一直都努力跟踪国外最先进的潮流。

计算机专业的教学既有理论学习，也必须配合实际操作，课程设计教材正是为了体现这种特点而编写的。本课程设计是为了配合Java程序设计课程、帮助学生深入学习掌握Java语言、熟练运用这个工具而编写的，它并不象通常的Java教材那样系统而详尽地介绍语法知识，而是在每章给出的知识点一节中，针对课程设计题目有针对性地回顾知识，帮助学生复习这些内容，顺利地完各个设计项目。

本教材的目的旨在通过课程设计各个项目的综合训练，培养学生实际分析问题、编程和动手能力，提高学生的综合素质。本课程设计尝试使用一些较生动的示例和设计项目，激发学生的学习兴趣，引导学生主动学习，正确理解和接受需要重点掌握的知识点，为参加项目开发及更深入的学习做好准备。

本课程的内容符合教育部教学指导委员会制订的课程要求，注意吸取国外著名大学相关课程教学实习的经验和方法。根据Java课程的特点，本课程设计并没有针对具体的教科书，而是可以独立成为一门课程。课程设计的内容基本覆盖了Java课程设计所要讲授的内容，教师可以根据本学校的教学计划及学生的具体情况和特点，选择相应的课程设计内容。

本教材中，课程设计选择的设计项目比一般的课程要求学生实现的内容更接近于实用，每个课程设计的答案都不是唯一的，可以有多种实现方法。有些项目设计直接要求学生使用两种或两种以上的方式实现。这些项目都是开放式的，教师可以根据自己的计划改变项目目标，要求学生在基本内容之上增加功能，并鼓励学生完善设计。

本书共分10章，从第2章至第9章是基本的项目设计题目。第10章是一个综合练习，它将前面几章的题目集成起来完成一个简单而全面的系统，着重训练学生的综合能力。各章都给出简单的设计思路，有些题目还详细说明重点语句，最后都给出了参考程序。在实际考核时，应重点考核学生调试程序的能力及扩展功能的能力。每个设计题目都建议学生在已有代码基础之上，扩展一些功能，并完美地实现自己的设想。教师可以根据题目的内容和教学计划，对原来的设计目标做适当修改。

本书在编写过程中，得到了许多人的大力支持。北京大学计算机系的陈向群教授向作者介绍了很多国外相关教材的编写与使用情况，并毫无保留地将自己的经验介绍给作者，作者受益匪浅。华中科技大学计算机系的刘振安教授耐心细致地审阅了本书全稿，提出了许多非常中肯的意见和建议，甚至不放过一个语法及文字错误。正是得益于这么多的教师和同行的帮助，才使本书顺利出版，作者在此表示深深的谢意。在感谢的同时，作者更多地是学到了各位老师的工作精神，这对作者是更大的精神财富。

VI

本书的编写还得到了南开大学信息学院院长吴功宜教授的大力支持，吴老师工作繁忙，但仍不忘询问编写进展情况，当遇到困难时，吴老师悉心指导，指点迷津。本书还得到了笔者的导师、南开大学信息学院卢桂章教授的帮助，在此一并表示感谢。

参加本书编写的除封面署名外，还有：熊国龙、徐小兵、王丽丽，他们调试了大部分的程序，并为本书的编写做了很多工作。由于作者水平有限，时间仓促，书中难免有错误和不妥之处，恳请广大读者特别是同行专家批评指正。

辛运伟

2004年12月于南开园

目 录

丛书序言

前言

第1章 课程学习目标1

1.1 概述1

1.2 课程设计结构1

1.3 评价标准3

第2章 Java的基本语法5

2.1 基础知识回顾5

2.1.1 Java的基本数据类型、运算符
与表达式5

2.1.2 流控制结构9

2.1.3 Java语言中的类和对象16

2.1.4 Java语言中的异常19

2.2 项目设计21

2.2.1 设计题目21

2.2.2 设计实现23

2.2.3 评分标准30

第3章 Java类库的基本知识和使用31

3.1 基础知识回顾31

3.1.1 Java包与类层次结构31

3.1.2 Java类的查找方法33

3.2 项目设计33

3.2.1 设计题目一33

3.2.2 设计实现34

3.2.3 评分标准37

3.2.4 设计题目二37

3.2.5 设计实现39

3.2.6 评分标准43

第4章 数组45

4.1 基础知识回顾45

4.1.1 Java的数组简介45

4.1.2 排序算法的介绍49

4.1.3 Java中如何实现链表53

4.2 项目设计56

4.2.1 设计题目56

4.2.2 设计实现59

4.2.3 评分标准74

第5章 Java的图形用户界面的设计77

5.1 基础知识回顾77

5.1.1 Java的GUI简介77

5.1.2 基于AWT的GUI设计77

5.1.3 基于Swing的GUI设计78

5.1.4 GUI事件处理78

5.1.5 布局管理器79

5.2 项目设计80

5.2.1 设计题目一80

5.2.2 设计实现81

5.2.3 评分标准83

5.2.4 设计题目二84

5.2.5 设计实现84

5.2.6 评分标准86

5.2.7 设计题目三86

5.2.8 设计实现89

5.2.9 评分标准96

第6章 文件操作97

6.1 基础知识回顾97

6.1.1 File类98

6.1.2 InputStream和OutputStream类100

6.1.3 FileInputStream和FileOutput
Stream类102

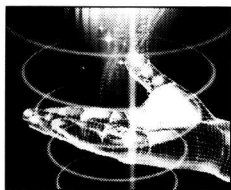
6.1.4 RandomAccessFile类103

6.1.5 示例103

6.2 项目练习104

6.3 项目设计107

6.3.1 设计题目	107	9.1 基础知识回顾	141
6.3.2 设计实现	108	9.1.1 Java的网络技术	141
6.3.3 评分标准	117	9.1.2 Java的Socket编程概述	142
第7章 Java多线程技术	119	9.1.3 使用多线程实现网络连接示例	143
7.1 基础知识回顾	119	9.2 项目设计	148
7.1.1 线程和多线程	119	9.2.1 设计题目	148
7.1.2 线程的状态	119	9.2.2 设计实现	149
7.1.3 创建线程	120	9.2.3 评分标准	178
7.1.4 线程的控制	121	第10章 综合练习	179
7.1.5 线程组	124	10.1 项目设计	179
7.1.6 线程的优先级	124	10.1.1 设计目标	179
7.1.7 同步问题	124	10.1.2 设计思路	179
7.2 项目设计	124	10.2 项目实现	181
7.2.1 设计题目	124	10.2.1 Application.java的实现	181
7.2.2 设计实现	125	10.2.2 Communication.java的实现	185
7.2.3 改善动画质量	126	10.2.3 Media.java的实现	189
7.2.4 评分标准	131	10.2.4 book.java的实现	192
第8章 Java Applet	133	10.2.5 cd.java的实现	196
8.1 基础知识回顾	133	10.2.6 tape.java的实现	200
8.1.1 Java Applet	133	10.2.7 Calculator.java的实现	203
8.1.2 Java Applet类方法	133	10.2.8 CalculatorAD.java的实现	205
8.1.3 Java Applet和HTML语言集成 程序设计	134	10.2.9 ChessInternalFrame.java的实现	212
8.2 项目设计	134	10.2.10 MainPanel.java的实现	222
8.2.1 设计题目	134	10.2.11 ChessModel.java的实现	226
8.2.2 设计实现	135	10.2.12 SocketThread.java的实现	234
8.2.3 设计实现改进	137	10.2.13 Sound.java的实现	236
8.2.4 评分标准	139	10.3 评分标准	237
第9章 Java的Socket编程	141	参考文献	238



第1章

课程设计目标

1.1 概述

本课程设计是为配合Java程序设计课程，帮助学生深入学习掌握Java语言，熟练运用这个工具而编写的。通过课程设计各个项目的综合训练，培养学生实际分析问题、编程和动手能力，提高学生的综合素质。

本课程设计尝试使用一些较生动的示例和设计项目，激发学生学习兴趣，引导学生主动学习，正确理解、接受需要重点掌握的知识点，为参加项目开发及更深入的学习做好准备。

本课程的内容符合教育部教学指导委员会制订的课程要求，注意吸取国外知名大学相关课程教学实习的经验和方法。根据Java课程的特点，本课程设计并没有针对某一具体的教科书，而是可以独立成为一门课程。本书内容基本覆盖了Java课程设计所要讲授的内容，教师可以根据本学校的教学计划及学生的具体情况和特点，选择相应的课程设计内容。

本课程设计选择的设计项目比一般的课程要求学生实现的内容更接近于实用，每个课程设计的答案都不是唯一的，可以有多种实现方法。有些项目设计直接要求学生使用两种或两种以上的方式实现。这些项目都是开放式的，教师可以根据自己的计划改变项目目标，要求学生在基本内容之上增加功能，并鼓励学生完善设计。

1.2 课程设计结构

本书共分10章，从第2章至第9章是基本的项目设计题目。每章第一小节列出相关的知识点，这些内容可以帮助学生复习巩固学习过的内容，也是设计本章项目时所要用到的知识。第二小节是具体的基本设计题目，这些题目针对Java语言中相关的知识点而设计，题目之间有联系，按层次逐步深入，后面的题目可以借助于前面题目中设计的内容。第10章是一个综合练习，更是将前面几章的题目集成起来完成一个简单而全面的系统，训练学生的综合能力。

1. Java的基本语法（第2章）

要进行Java程序设计必须先全面掌握基本语法，该章的练习即针对此目的。Java是面向对象的语言，类是其中最重要的元素，该章选择三种出版物，分别是书、CD及磁带，要求学生根据书、CD或磁带的特性来设计相关的类。由于这三个类同属于出版物，所以要求学生为出版物设计一个父类，书、CD和磁带都是从这个父类继承下来的子类。父类中可以封装公共的方法，子类还可以重写这些方法。

要求学生学会从键盘读入数据，并对输入的数据进行合法性检查，保证数据的正确性。对

输入格式不正确的数据进行异常处理。同时也要求学生学会在屏幕上显示信息，并注意信息的输出格式。

程序的流程控制可以使用分支和循环，并从这个练习开始，注意程序的书写风格，培养良好的编程习惯。

2. Java类库的基本介绍和使用（第3章）

在了解了Java语言的语法之后，还需要掌握Java给我们提供的类库。类库中提供的包、类及方法可以极大地方便我们的编程工作，降低工作强度，简化代码。Java中的包和类非常庞大，该章的项目选择了数学类和日历类的包进行练习。要求学生掌握相关的方法，在实际编程时，可以熟练地查找所需的包及类。

该章中安排了两个设计项目，一个是求素数问题，另一个是计算日期问题。其中计算日期用到了Java中的日历类，这和第2章中自己计算日期的方法有所不同，让学生针对同一个问题采用不同的方法实现，增强他们解决问题与处理问题的能力。此外，该练习中也增加了数组的定义和使用，下一章的项目设计中将针对数组进行全面的训练。

教师可以根据学生的具体情况，对学生提出不同的要求，建议要求学生必须完成第一个设计，第二个设计可以做为选作题，与第5章的计算器程序合并完成。

3. 数组（第4章）

数组是高级程序设计语言中不可缺少的内容，这个环节的练习对学生非常重要。该章的项目设计是对第2章项目设计的扩充，可以使用第2章中定义的出版物类，在此基础上，使用数组存储从键盘输入的信息。

对数组中的数据进行排序可以方便随后的查找。要排序必然要进行数据的移动，从而带来一定的系统开销。更重要的问题是，如果要求按不同的数据域作为关键字进行排序时，会导致更大的麻烦。这一章我们介绍使用引用来实现链表，完成了数组元素的排序，实现了不使用指针的Java链表。

该章项目设计中只将一个数据域作为关键字，实现时教师可根据实际情况增加出版物的数据项，同时除主关键字域之外，还可以指定多个次关键字域，并要求学生按各关键字完成相应的排序，如按出版物名称排序、按演唱者或作者名称排序、按出版社名称排序等。相应地，针对这些关键字值可以进行组合查找，查找的条件可以是多个。

建议学生必须完成本项目。

4. Java的图形用户界面GUI的设计（第5章）

Java可以实现美观方便的图形用户界面。图形界面有两种实现方式，分别是传统的AWT组件方式和新的Swing方式。该章设计的计算器要求分别使用两种方式实现。

另外，为了扩充计算器的功能，要求计算器不但可以完成数值量的计算，还可以实现日期的计算，例如计算两个日期之间相差的天数。

建议教师要求学生必须完成数值计算器的设计。根据学生的具体情况，教师可以在此基础上

上, 增加计算器的其他功能, 例如可以实现一个更复杂的科学计算器, 可以增加计算幂运算等这样的若干函数计算, 可以增加存储器功能, 还可以增加弧度计算、二进制运算等。当然, 计算器实现的功能越多, 学生相应的分数也应该越高。

5. 文件操作 (第6章)

该章设计了一个通讯录小程序, 具有添加、查找等功能, 并使用文件保存已输入的通讯录。目的在于训练学生掌握文件的使用技巧。

Java中的输入输出流非常丰富, 教师应该要求学生熟练掌握。可能的话, 要扩充通讯录的功能, 例如可以将通讯录导出到磁盘文件系统中保存起来, 可以导入已保存的通讯录文件等, 文件名也可以要求用户随时输入, 而不必保存到固定文件中。

该项目是必选完成的。

6. Java多线程技术 (第7章)

线程是Java中的重要概念, 也是实际编程中需要掌握的重要技巧。该章利用多线程方法设计了一个动画, 通过这个项目的设计, 学生可以了解一个线程的生命周期, 从而能够更加自如地控制程序的运行。

该章实现的动画程序运用了多种技术消除动画闪烁现象, 教师可根据具体情况提出更高要求, 要求学生完善这个设计, 很好地实现动画程序。

7. Java Applet (第8章)

Java可以实现页面设计, 这通过Applet来完成。Applet是和前几章设计的应用程序完成不同的—类小应用程序。该章利用Applet技术实现网页中的一个动画, 要求学生初步掌握相关的知识。

建议必须完成该项目设计。

8. Java的Socket编程 (第9章)

网络编程中必须要用到Socket概念。该章设计了一个游戏, 可以实现本地通信, 也可以实现网络通信。游戏的规则不是主要的, 相关的策略也不一定要求非常完美。该项目的重点在于能正确实现基于服务器-客户端的通信机制, 实现网络环境下的编程。

9. 综合练习 (第10章)

该章将前面几个项目集成起来, 形成一个小系统, 并使用图形窗口, 实现了菜单操作。这样的—个系统已经非常接近实际的应用程序了。

由于在前几章的设计中, 各个项目自成一体, 每个都有主函数入口。该章集成起来时需要进行相应的修改, 因此, 该章的工作主要集中在这方面。

另外, 该章在有些菜单项上又扩充了功能, 例如, 可以打开已有的文件等。

1.3 评价标准

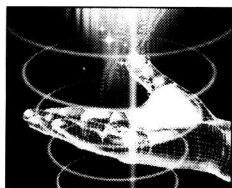
各章都给出简单的设计思路, 有些题目还详细说明重点语句, 最后都给出了参考程序。因

此在实际考核时，应重点考核学生调试程序的能力及扩充功能的能力。每个设计题目都建议学生在已有代码基础之上，扩充一些功能，并能完美地实现。教师可以根据题目的内容和教学计划，对原来的设计目标做适当修改。

如果只完成了课程设计的基本目标，并能调试成功，则只能得到75至80分的基本分。如果能够添加功能，并正确实现，一般要视功能的强弱及实现的好坏给予适当加分，鼓励学生创新性的工作。

如果学生没有完成规定的内容，或是程序有比较大的错误，则不能及格。学生可以用一些创新性的工作代替项目设计中的一些基本功能。

教师在题目选择及评分方面有充分的自主权。



第2章

Java的基本语法

2.1 基础知识回顾

2.1.1 Java的基本数据类型、运算符与表达式

1. Java基本语法单位

Java中的基本语法单位包括空白、注释、语句、标识符及关键字。

(1) 空白

在Java程序中，换行符及回车键都可表示一行的结束，它们可被看作是空白。另外，空格键、水平定位键（tab）亦是空白。为了增加可读性，Java程序的元素之间可插入任意数量的空白，执行时将忽略掉多余的空白。

(2) 注释

Java中的注释不能插入在一个标识符或关键字之中，程序中允许加空白的地方就可以写注释，注释不影响程序的执行结果。Java有3类注释，其形式及对应的含义如下：

```
// 在一行的注释，从这个标识到行尾之间都是注释；
```

```
/* 一行或多行的注释，这两个标识之间都是注释 */
```

```
/** 文档注释 */
```

(3) 语句

Java中最小的执行单位是语句。Java各语句间以分号“;”分隔。一个语句可写在连续的若干行内。花括号“{”和“}”包含的一系列语句称为语句块，简称为块。语句块可以嵌套，即语句块中可以含有子语句块。在词法上，块被当作一个语句看待。

(4) 标识符

标识符是程序中的那些中间不带空格的单词，它们是以字母、下划线或者美元符号\$开头，由字母、数字、下划线（_）或美元符（\$）组成的字符串。标识符区分大小写，长度没有限制，但是不能以数字作为开头。在程序中，标识符可用作变量名、方法名、接口名、类名等。

(5) 关键字

有些特殊的标识符称为保留字或者关键字，它们是Java语言中有固定含义的标识符。Java中的关键字包括：

abstract	boolean	break	byte	case	cast
catch	char	class	const	continue	default
do	double	else	extends	false	final
finally	float	for	future	generic	goto
if	implements	import	inner	int	instanceof
interface	long	native	new	null	operator
outer	package	private	protected	public	rest
return	short	static	super	switch	synchronized
this	throw	throws	transient	true	try
var	void	volatile	while		

2. 简单数据类型

Java的数据类型共分为两大类，一类是简单类型，一类是引用类型。简单数据类型包括逻辑类型、字符类型、整数类型和浮点类型。引用类型包括数组、类和接口等。

表2-1列出了所有简单数据类型的关键字、范围/格式，并给出了简单说明。

表2-1 Java简单数据类型

	类 型	范围/格式	说 明
整数型	byte	8位二进制补码	字节型整数
	short	16位二进制补码	短整数
	int	32位二进制补码	整数
	long	64位二进制补码	长整数
浮点型	float	32位IEEE754规范	单精度浮点数
	double	64位IEEE754规范	双精度浮点数
其他类型	char	16位Unicode字符集	单字符
	boolean	true或false	布尔值

3. 复合数据类型

大多数程序设计语言如PASCAL、C、C++等都提供类型变量的概念。类型变量是一个值，可以是整数或浮点数或一个字符。每种语言本身都有几种内置类型，除此之外，大多数程序设计语言还为用户提供了定义新类型的手段，这大大扩展了语言的处理能力。一般情况下，我们称用户定义的新类型为复合数据类型。Java是面向对象的程序设计语言，它为用户提供的复合数据类型就是引用数据类型，即类、接口和数组。

4. 各类型数据间的转换

整型、浮点型、字符型数据可以混合运算。运算时，不同类型的数据先转换为同一类型，

然后再进行运算。转换的一般原则是位数少的类型转换为位数多的类型，这称作自动类型转换。转换规则如表2-2所示。

表2-2 自动类型转换规则

操作数1类型	操作数2类型	转换后的类型
byte或short	int	int
byte或short或int	long	long
byte或short或int或long	float	float
byte或short或int或long或float	double	double
char	int	int

5. 运算符与表达式

(1) 运算符

Java语言中的常量只有简单数据类型和String类型。变量是存储数据的基本单元，它在使用之前要先说明，变量说明的基本格式为：

类型 变量名1[= 初值1][, 变量名2 [= 初值2]]…;

变量说明的地方有两处，一处是在方法内，另一处是在类定义内。方法内定义的变量称作自动变量，也称为局部变量、临时变量或栈变量。类中定义的变量就是它的成员变量。

变量的作用域是指可访问该变量的代码范围。类中定义的成员变量的作用域是整个类。方法中定义的局部变量的作用域是从该变量的说明处开始到包含该说明的语句块结束处，块外是不可使用的。

块内说明的变量将屏蔽其所在类定义的同名变量。但同一块中如果定义两个同名变量则引起冲突。

Java中的运算符包括算术运算符、关系运算符、逻辑运算符和位运算符等。

算术运算符包括通常的加(+)、减(-)、乘(*)、除(/)、取模(%), 完成整型或浮点型数据的算术运算。算术运算符除了具备通常的意义外，Java对此有所扩展，它允许对浮点数进行取模操作。此外，算术运算符还包括“++”、“--”两种，分别表示加1和减1操作。

关系运算符用来比较两个值，包括大于(>)、大于等于(>=)、小于(<)、小于等于(<=)、等于(==)、不等于(!=) 6种。关系运算符都是二元运算符，运算的结果是一个逻辑值。

Java允许“==”和“!=”两种运算用于任何数据类型。

逻辑运算符包括逻辑与(&&)、逻辑或(||)和逻辑非(!)。前两个是二元运算符，后一个是一元运算符。

位运算符用来对二进制位进行操作，其结果均为整型量。位运算符分为：移位运算符和逻辑位运算符，它只能对整型和字符型数据进行操作。

其中，向右移位运算符有二个，>>与一般意义下的移位运算类似，例如：

256 >> 4 得到 16
-256 >> 4 得到 -16

此外，还有一个逻辑右移或叫无符号右移运算符>>>，它只对位进行操作，而没有算术含义，它用0填充左侧的空位，例如：

0xa2 >>> 2 得到 40
(byte) 0xa2 >> 2 得到 -24
(byte) 0xa2 >>> 2 得到 1073741800

表2-3总结了位运算符。

表2-3 位运算符

运算符	用 法	操作说明
>>	op1>>op2	将op1右移op2位
>>>	op1>>>op2	将op1右移op2位(无符号的)
<<	op1<<op2	将op1左移op2位
&	op1&op2	按位与
	op1 op2	按位或
^	op1^op2	按位异或
~	~op1	按位求补

此外，Java中的运算符还包括扩展赋值运算符（=、+=、-=、*=、/=、%=、&=、|=、^=、>>=、<<=）及（>>>=），条件运算符（?:），点运算符（.），实例运算符（instanceof），new运算符，数组下标运算符（[]）等。

(2) 运算符的优先次序

在对一个表达式进行计算时，如果表达式中含有多种运算符，则应按运算符的优先顺序依次从高向低进行，同级运算符则从左向右进行。括号可以改变运算次序。见表2-4。

表2-4 运算符的优先次序

运算符	优先次序
后缀运算符	., [], ()
单元运算符	++, --, !, ~
创建运算符	new (type)
乘除运算符	*, /, %
加减运算符	+, -
移位运算符	>>, >>>, <<
关系运算符	<, >, <=, >=, instanceof
相等与不等	==, !=
位运算符	&
位运算符	^
位运算符	