

“十三五”普通高等教育规划教材

Java程序设计 学习与考试强化指导

Java CHENGXU SHEJI XUEXI YU KAOSHI QIANGHUA ZHIDAO

施 珺 赵雪峰 陈艳艳 主 编
郁洪波 王 霞 纪兆辉 副主编



鼠标连接梦想 键盘写出辉煌

中国铁道出版社有限公司
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

“十三五”普通高等教育规划教材

Java 程序设计学习与考试强化指导

施 珺 赵雪峰 陈艳艳 主 编
郁洪波 王 霞 纪兆辉 副主编

内 容 简 介

本书是针对全国计算机等级考试二级 Java 语言程序设计编写而成的,通过对考试知识点的讲解和总结,帮助学生快速掌握 Java 客观题和编程操作题的相关知识及应试技巧。

全书按照全国计算机等级考试二级 Java 语言程序设计的内容进行合理安排,分为 Java 程序设计知识要点综述、Java 理论知识典型考题精解、Java 程序填空题典型考题精解、二级公共基础知识综述、二级公共基础知识典型例题精解、二级 Java 考试真题汇编、Java 学习与考试方法指导,共 7 章。

本书内容紧扣全国计算机等级考试二级 Java 语言程序设计考试大纲,适合作为准备参加全国计算机等级考试二级 Java 语言程序设计考生的复习资料,尤其适合在校大学生复习应试。

图书在版编目(CIP)数据

Java 程序设计学习与考试强化指导/施珺,赵雪峰,陈艳艳主编. —北京:
中国铁道出版社有限公司,2019.7

“十三五”普通高等教育规划教材

ISBN 978-7-113-25948-8

I. ①J… II. ①施…②赵…③陈… III. ①JAVA 语言-程序设计-高等学校-
教学参考资料 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2019)第 141148 号

书 名: Java 程序设计学习与考试强化指导

作 者: 施 珺 赵雪峰 陈艳艳

策 划: 朱荣荣

编辑部电话: 010-63589185 转 2068

责任编辑: 朱荣荣 包 宁

封面设计: 刘 颖

责任校对: 张玉华

责任印制: 郭向伟

出版发行: 中国铁道出版社有限公司(100054,北京市西城区右安门西街8号)

网 址: <http://www.tdpress.com/51eds/>

印 刷: 北京柏力行彩印有限公司

版 次: 2019年7月第1版 2019年7月第1次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 20.25 字数: 496 千

书 号: ISBN 978-7-113-25948-8

定 价: 48.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

Java 是网络时代最重要的编程语言之一，具有简单、面向对象、分布式、健壮、安全、平台无关、多线程、动态等特点，应用面广。在全国计算机等级考试中，Java 语言是重要的二级考试语种之一。

为了帮助广大学生更好地学习 Java 程序设计的相关知识并顺利通过全国计算机等级考试二级（Java 语言程序设计）（以下简称二级 Java），我们组织编写了这本《Java 程序设计学习与考试强化指导》。本书编写重点放在知识点综述和典型例题解析指导上，将知识点和考点科学地结合起来，通过对典型考题所涉及的知识点进行全面分析、归纳、总结，指导学生掌握解题思路，并能举一反三、融会贯通，帮助学生高效率地掌握二级 Java 考试的相关知识和应试技巧。

二级 Java 采用上机考试方式，由理论知识单选题（40 分）和编程填空题（60 分）两部分组成。其中，公共基础知识单选题 10 分，Java 语言单选题 30 分，Java 基本操作题 18 分，Java 简单应用题 24 分和 Java 综合应用题 18 分。本书参照二级 Java 考试的三大模块（Java 理论知识、Java 编程操作、公共基础知识），分 6 章进行了知识点归纳综述、典型例题精解和真题练习，最后编写了学习方法与应试策略。

由本书编者主讲的江苏省在线开放课程“Java 面向对象程序设计”已经于 2017 年在中国大学 MOOC 平台上线，每年春、秋学期均开课，建议读者在报考二级 Java 之前在线选课学习。

本书所列编程填空题的源代码可以从凌风智慧教育研究所建成的“Java 精品资源共享课”网站“配套教材”栏目中下载，网址是：<http://java.hhit.edu.cn>。该网站还提供了与 MOOC 平台课程配套的 PPT 版全套教学课件、大量综合性的 Java 典型例题和丰富的全国计算机二级考试真题资料。

本书由施珺、赵雪峰、陈艳艳任主编，郁洪波、王霞、纪兆辉任副主编。编写分工如下：第 1 章由施珺、陈艳艳、赵雪峰、纪兆辉共同编写，第 3 章、第 4 章由施珺编写，第 2 章、第 6 章由赵雪峰、陈艳艳、郁洪波、王霞、施珺共同编写，第 5 章由陈艳艳、蔡虹编写，第 7 章由施珺、赵雪峰编写。由于编者水平有限，加上时间仓促，书中难免有疏漏和不足之处，恳请广大读者指正。

愿本书能对读者顺利通过全国计算机二级 Java 的考试有所帮助。

编者

2019 年 5 月

第 1 章 Java 程序设计知识要点综述..... 1	1.5 Java Applet 小程序.....41
1.1 Java 概述..... 1	1.5.1 Applet 的基本工作原理..... 41
1.1.1 Java 开发平台..... 1	1.5.2 Applet 类的主要方法..... 41
1.1.2 Java 程序文件的结构..... 2	1.5.3 Applet 的参数传递..... 42
1.2 Java 程序设计基础..... 4	1.6 输入/输出流和文件处理.....44
1.2.1 Java 标识符和关键字..... 4	1.6.1 字节输入/输出流..... 44
1.2.2 Java 数据类型..... 5	1.6.2 字符输入/输出流..... 45
1.2.3 常量、变量、运算符 与表达式..... 6	1.6.3 文件的处理和随机访问..... 46
1.2.4 流程控制语句..... 9	1.6.4 对象的系列化..... 48
1.2.5 Java 的数组..... 11	1.7 Java 的多线程机制.....50
1.2.6 字符串..... 12	1.7.1 程序、进程、线程的概念..... 50
1.3 Java 面向对象编程..... 13	1.7.2 线程的状态与生命周期..... 50
1.3.1 面向对象程序设计的 基本概念..... 13	1.7.3 Java 中实现多线程的方法..... 51
1.3.2 类的定义与对象的创建..... 14	1.7.4 线程的优先级与调度..... 53
1.3.3 继承与多态..... 19	1.7.5 线程的同步..... 53
1.3.4 Java 的非访问控制符..... 21	1.8 基础工具类和集合框架.....54
1.3.5 Java 的包与访问控制符..... 22	1.8.1 Java 的系统类与工具类..... 54
1.3.6 接口..... 24	1.8.2 集合框架与泛型..... 57
1.3.7 Java 的异常处理..... 25	第 2 章 Java 理论知识典型考题精解... 59
1.3.8 内部类与匿名类..... 27	2.1 Java 程序设计基础.....59
1.4 Java 图形用户界面的程序设计.... 28	2.2 Java 面向对象编程..... 73
1.4.1 GUI 设计基本概念..... 28	2.3 Java 图形用户界面的程序设计.....82
1.4.2 常用的 AWT 和 Swing 标准 组件和容器..... 30	2.4 Java Applet 小程序设计..... 88
1.4.3 布局管理器..... 31	2.5 输入/输出流和文件处理.....89
1.4.4 菜单..... 33	2.6 Java 多线程.....93
1.4.5 对话框..... 34	2.7 基础工具类与集合框架.....97
1.4.6 事件处理..... 37	2.8 自测习题.....99
1.4.7 图形用户界面的用户自定义 成分..... 39	第 3 章 Java 程序填空题典型考题 精解..... 110
	3.1 程序填空题常见考点分析.....110

3.1.1	Java 程序的基本结构定义	110	4.4.3	结构化设计方法	216
3.1.2	变量的定义与赋值、数据类型转换	114	4.4.4	软件测试	217
3.1.3	成员方法的定义与调用	117	4.4.5	程序的调试	218
3.1.4	字符串及数组相关的属性与方法	118	4.5	数据库设计基础	218
3.1.5	三类对话框	120	4.5.1	数据库系统的基本概念	218
3.1.6	图形用户界面设计中的组件定义、初始化、布局	122	4.5.2	数据模型	220
3.1.7	事件处理	123	4.5.3	关系代数	222
3.1.8	多线程的使用	125	4.5.4	数据库设计与管理	227
3.1.9	数据流与文件的相关方法	126	第 5 章	二级公共基础知识典型例题精解	229
3.2	程序填空题解答指导	127	5.1	基本数据结构与算法	229
3.2.1	解题思路	127	5.2	程序设计基础	233
3.2.2	解题技巧	127	5.3	软件工程基础	236
3.3	基本操作题	131	5.4	数据库设计基础	238
3.4	简单应用题	140	5.5	自测习题	244
3.5	综合应用题	152	第 6 章	二级 Java 考试真题汇编	247
3.6	自测习题	165	6.1	Java 单选题真题集	247
第 4 章	二级公共基础知识综述	191	6.2	Java 程序填空题真题集	268
4.1	考点提要	191	6.3	公共基础知识真题集	292
4.2	数据结构与算法	192	第 7 章	Java 学习与考试方法指导	297
4.2.1	算法	192	7.1	Java 程序设计学习之道	297
4.2.2	数据结构的基本概念	193	7.1.1	日常学习	297
4.2.3	线性表	194	7.1.2	期末复习	299
4.2.4	栈和队列	197	7.2	二级 Java 考试简介	299
4.2.5	二叉树	200	7.2.1	考试大纲	299
4.2.6	查找技术	204	7.2.2	考试形式	300
4.2.7	排序技术	205	7.3	二级 Java 考试应试策略	308
4.3	程序设计基础	210	7.3.1	复习思路	308
4.3.1	程序设计方法与风格	210	7.3.2	答题技巧	309
4.3.2	结构化程序设计	211	参考答案	312	
4.3.3	面向对象的程序设计方法	212	参考文献	318	
4.4	软件工程基础	213			
4.4.1	软件工程基本概念	213			
4.4.2	结构化分析方法	215			

1.1 Java 概述

Java 是 Sun 公司于 1995 年正式推出的新一代面向对象程序设计语言。Java 语言具有简单易学、面向对象、平台无关、解释执行、多线程、分布式、安全、动态等特点。

Java 概述导学图如图 1-1 所示。



图 1-1 Java 概述导学图

1.1.1 Java 开发平台

要用 Java 语言开发应用软件, 首先需要安装 Java 开发环境, 包括 Java 平台及 Java 编程环境。Java 发展到现在, Java 平台主要分为三个版本: Java SE、Java EE、Java ME。Java 编程环境很多,

可以是简单的文本编辑器类软件，比如 TextPad，也可以是集成开发环境，比如 NetBeans IDE、Eclipse 等。

如果要开发 Java 程序，则必须获得 JDK（Java 开发工具包），JDK 包括了 JRE（Java 运行环境）以及开发过程中所需要的一些工具程序，像 javac、java、appletviewer、javadoc、javap 等工具程序。JVM（Java 虚拟机）包括在 JRE 中，所以若要运行 Java 程序，必须获得 JRE 并进行安装。

常用 JDK 工具程序的功能见表 1-1。

表 1-1 常用 JDK 工具程序一览表

程序名称	功能描述	命令格式
javac.exe	这是 Java 编译器程序，负责检查 Java 源程序是否有语法错误并生成相应的字节码文件，字节码文件的基本名与源文件中类名同名、但扩展名为.class	javac 文件名.java
java.exe	这是 Java 解释器程序，负责解释执行 Java Application 字节码文件	java 主类名
appletviewer.exe	这是 Java 小程序查看器，可以模拟 WWW 浏览器运行 Applet 小程序，使用它调试程序，不需要反复调用庞大的浏览器	appletviewer 文件名.html
javap.exe	这是 Java 反汇编器，显示编译类文件中的可访问功能和数据，同时显示字节代码含义	javap 选项参数 类名
javadoc.exe	这是 Java 文档生成器，可以根据 Java 源代码中的说明语句生成 HTML 格式的 API 说明文档	javadoc -d 文档存放目录 -author -version 源文件名.java
jar.exe	这是 Java 打包工具，可将多个相关的类文件打包成单个 JAR 文件，用来发布 Java 应用程序，双击该 jar 文件即可运行应用程序	具体操作方式详见： http://java.hhit.edu.cn/News/Detail_Layout1.aspx?id=19009

1.1.2 Java 程序文件的结构

Java 采用面向对象的编程方式，源程序都是由若干书写形式互相独立的类组成的。根据程序运行环境的不同，Java 程序可以划分为两大类：

- Java Application，即 Java 桌面应用程序，可以独立运行。
- Java Applet，即 Java 小程序，需要嵌入 HTML 网页中运行。

Java 源程序是扩展名为.java 的简单文本文件，Java 源程序经由 Java 编译器 javac.exe 生成字节码文件；Java 字节码是扩展名为.class 的可解释执行的二进制文件，Java 字节码文件由 Java 解释器 java.exe 解释执行。

1. Java Application 程序的结构

1) Java Application 字符界面

字符界面的程序基本结构如下：

```
import java.包.*;           //根据需要加载要用的包中的类，如果是 lang 包中的类则省略
public class 类名           //类声明
{
    public static void main(String args[]) //必需的 main 方法，程序运行入口
    {
        ... //语句系列 }方法体
    }
}
```

}类体

一个 Java 应用程序必须有一个类包含 main()方法, main()方法所在的类是应用程序的主类,也是应用程序运行时的入口。一个程序中有多个类时,主类可以是非 public 类,但运行时要调用主类名。

2) Java Application 图形界面

图形用户界面 (Graphical User Interface, GUI, 又称图形用户接口)是指采用图形方式显示的计算机操作用户界面。图形界面的 Java 应用程序可以通过各种组件响应用户操作,实现复杂的交互功能,是目前广泛采用的桌面应用程序界面。

图形界面的程序基本结构如下:

```
import java.awt.*;           //加载图形界面设计要用的 awt 包中的所有类
import java.awt.event.*;    //加载事件响应要用到的 event 包中的类
public class 类名 extends Frame implements 事件接口 //声明该类是继承于 Frame 类的子
//类,并实现了对事件接口的响应
{
    组件类 xx;               //组件声明,这里 xx 泛指组件类对象的名称引用
    public 构造方法名()     //与类同名
    {
        setTitle("标题栏显示文字"); //设置窗体的标题
        setLayout(new FlowLayout()); //设置窗体上各控件的布局为流式布局
        xx=new 组件构造方法名(实参); //xx 组件初始化
        add(xx);             //将 xx 组件加入窗体
        xx.add 事件接口(this); //为组件注册事件监听器
        setSize(w,h);       //设置窗体宽度 w、高度 h
        setVisible(true);   //让窗体可见
    }
    public void 接口中的抽象方法(接口事件 e)
    {
        //处理事件响应的代码块
    }
    public static void main(String args[]) //main()方法
    {
        new 构造方法名();     //构造一个当前类的新窗体对象
    }
}
```

如果要用 Swing 组件设计图形界面,则需要在上面的框架中增加一行代码:

```
import javax.swing.*;      //加载 javax.swing 包中的类
```

同时将类声明改为:

```
public class 类名 extends JFrame //声明该类是继承于 JFrame 类的子类
```

其余结构相同。

2. Java Applet 小程序的结构

Java Applet 是用 Java 语言编写的一种不能单独运行但可嵌入在其他应用程序中的小程序。一般嵌入在 HTML 编写的 Web 页面中,可以通过<applet>标记把 XXX.class 嵌入页面中,由 Web 浏览器内部包含的 Java 解释器解释运行。

Java Applet 程序需要在 Web 浏览器中运行,浏览器运行时自带窗口,所以 Java Applet 适合以

图形化界面展示,其初始化代码一般写在 Applet 类的 init()方法中,运行时自动执行。Applet 程序不以 main()方法为主入口,所以无须包含 main()方法。

Java Applet 的程序结构如下:

```
import java.applet.*;           //必须加载 applet 包中的类
import java.awt.*;             //加载图形界面设计用到的 awt 包中的类
public class 类名 extends Applet //声明该类是继承于 Applet 类的子类
{
    ...                          //其他语句
}
```

嵌入相应 HTML 网页文件中的代码如下:

```
<html>
  <body>
    <applet code="类名.class" height=400 width=500>
    </applet>
  </body>
</html>
```

注:代码中的 height=400, width=500 用于设置 Applet 在网页上的显示区域大小,此数值仅供参考,可根据需要适当修改。

1.2 Java 程序设计基础

1.2.1 Java 标识符和关键字

1. 标识符

标识符是 Java 程序代码中用来标记类名、方法名、变量名、常量名、数组名、包名等名称的字符系列。Java 是英文字母大小写敏感的语言,标识符也要区分字母大小写。

1) 标识符需遵循的命名规则

- (1) 标识符可以由字母、数字和两个特殊字符(下画线_、美元符号\$)组合而成。
- (2) 标识符必须以字母、下画线_或美元符号\$开头,不能以数字开头。
- (3) 标识符是区分大小写的,长度不限,但实际命名不宜过长。
- (4) 应该使标识符能一定程度上反映它所表示的变量、常量、对象或类的意义。
- (5) Java 的关键字不可以用作标识符。

2) Java 中常用的命名约定

- (1) 包名:包名是全小写的名词。
- (2) 类名、接口名:一般由一个或多个单词组成,且尽量使用名词,每个单词的首字母大写而其他字母均为小写,如 MobileTelephone;如果类名称中包含单词缩写,则这个缩写词的每个字母均应大写,如 XMLExample。
- (3) 方法名:第一个单词首字母小写,其他单词的首字母大写,如 getAge(),一般采用动词或动词+名词组合。

(4) 常量名:基本数据类型的常量名为全部大写,如果是由多个单词构成,可以用下画线隔开,如 MAX_CARD_NUMBER。

(5) 变量名：全小写，一般为名词，如 length。

注意：实际使用中，习惯约定变量命名使用 3 个字符前缀来表示数据类型，即基本结构为 typeVariableName。例如，定义一个整型变量：intDocCount，其中 int 表明数据类型，后面为表意的英文名，每个单词首字母大写。

2. 关键字

Java 中事先规定了 50 个关键字（见表 1-2），这些关键字不允许当作标识符来用。

表 1-2 Java 关键字列表

abstract	assert	boolean	break	byte
case	catch	char	class	const
continue	default	do	double	else
enum	extends	final	finally	float
for	goto	if	implements	import
instanceof	int	interface	long	native
new	package	private	protected	public
return	strictfp	short	static	super
switch	synchronized	this	throw	throws
transient	try	void	volatile	while

注意：true、false、null 虽然不是 Java 关键字，但也不能用作标识符。

1.2.2 Java 数据类型

Java 数据类型可分为两大类：基本数据类型和引用数据类型。基本数据类型是原始类型，可直接用于变量和常量声明，基本数据类型分为 8 种：char、boolean、byte、short、int、long、float、double。引用数据类型包括类、数组、接口。下面主要介绍基本数据类型。

1. 基本数据类型

各种基本数据类型占用字节数和取值范围见表 1-3。

表 1-3 Java 基本数据类型

数据类型	关键字	对应的造型类	占用字节数（位数）	取值范围
整型	int	Integer	4 (32)	$-2^{31} \sim 2^{31} - 1$
字节型	byte	Byte	1 (8)	$-2^7 \sim 2^7 - 1$ (-128~127)
短整型	short	Short	2 (16)	$-2^{15} \sim 2^{15} - 1$ (-32 768~32 767)
长整型	long	Long	8 (64)	$-2^{63} \sim 2^{63} - 1$
单精度浮点型	float	Float	4 (32)	$-3.402\ 823\ 47E+38 \sim 3.402\ 823\ 47E+38$
双精度浮点型	double	Double	8 (64)	$-1.797\ 693\ 134\ 862\ 315\ 7E+308 \sim 1.797\ 693\ 134\ 862\ 315\ 7E+308$
字符型	char	Character	2 (16)	'\u0000'~'\uffff'
逻辑型	boolean	Boolean	1 (8)	true、false

几点说明:

(1) Java 中的整数类型是不含小数的数值类型, 包括 int (是默认的整数类型)、byte、short、long。例如, 123 (十进制, 无特殊标识), 077 (八进制, 以 0 为前缀), 0x3ABC (十六进制, 以 0x 或 0X 为前缀)、2005L (长整型整数, 以大写 L 或小写 l 作为后缀)。

(2) 浮点类型是带小数的数值类型, 包括 float (以 f 为后缀且不可省略)、double (以 d 为后缀、可省略, 是默认的小数类型), 浮点类型数据只采用十进制数表示, 如 1.23f、2345.349d、1.56 (省略了后缀 d)。

(3) Java 中采用的字符集为 Unicode 编码, 1 个字符占 16 位。

(4) 字符类型 (char) 是用单引号括起来的单个字符, 如 'A'。

(5) 字符型常量也可以用整数表示, 例如: 'A'、'7' 也可以用整数 65、55 表示。

(6) Java 中使用转义字符 “\” 表明一个特殊的字符, 如 \r 表示一个回车符、\n 表示一个换行符、\t 表示一个制表符。

(7) 由多个字符组成的字符序列称为字符串 (String), 字符串用双引号括起来, 如 "green"。

(8) 逻辑类型 (boolean) 又称布尔型, 只有两个取值 true 和 false, 分别表示逻辑值 “真” 和 “假”。

2. 基本数据类型的转换

Java 语言的多种基本类型数据在运算时, 存在相互之间的转换问题。当一个数据从某个类型被转换为取值范围更大的类型时, 不会损失任何信息, 此时的类型转换可以自动进行。例如, 可以将 byte 类型数值转换为 short 类型、将 short 类型数据转换为 int 类型。反之, 则需进行强制类型转换, 此时该类型转换很可能存在精度损失。

Java 允许下面的基本数据类型转换自动进行:

byte → short → int → long → float → double; 字符型: char → int。

强制数据类型转换方式为:

(目标类型名)要转换的表达式或值;

例如:

```
int x=(int)23.56;
int y=(int)1999L;
long z=(long)56.36F;
float a=0; long b=0; double c=0; int d=(int)a+(int)b;
```

几点说明:

(1) 逻辑型数据不能将数据类型转换为数值型或字符型。

(2) 字符类型可以转换为数值型, 例如:

```
int x='A' ; //x=65, 对应其在 Unicode 中的位置
```

(3) 字符串可以转换为数值型, 例如:

```
double y=Double.parseDouble ("12.5"); //y=12.5
```

1.2.3 常量、变量、运算符与表达式

1. Java 常量

常量是在程序运行过程中其值保持不变的数据。常量名应符合标识符命名规则, 且一般用全

部字母大写表示，多个单词间以下画线（_）间隔。Java 中可以用关键字 `final` 表示一个常数，定义常量标识符的格式为：

```
final 数据类型 常量名=常量值;
```

例如：

```
final double CM_PER_INCH=2.54;
```

还可以用 `static final` 定义适用于某个类中多个方法的类常数。

2. Java 变量

变量是内存区域在程序中的表示，它是内存中的一个存储单元，存储单元中的内容在程序运行时可以变化。在 Java 中，定义了一个变量就会为其分配一块内存空间，此空间中可以存储一个某类型的数据。Java 中的变量应先定义后使用，在定义时需要指明变量的名称和数据类型，变量名首字母小写。

变量的定义格式为：

```
数据类型 变量名 [=初值];
```

例如：

```
int a=5;
int b;
```

变量有一定的生命周期和作用域，变量的作用域是指允许访问变量的范围。

3. Java 运算符

运算符用来对数据进行加工处理，Java 中的运算符主要分为赋值运算符、算术运算符、位运算符、逻辑运算符、关系运算符等。Java 运算符导学图如图 1-2 所示。

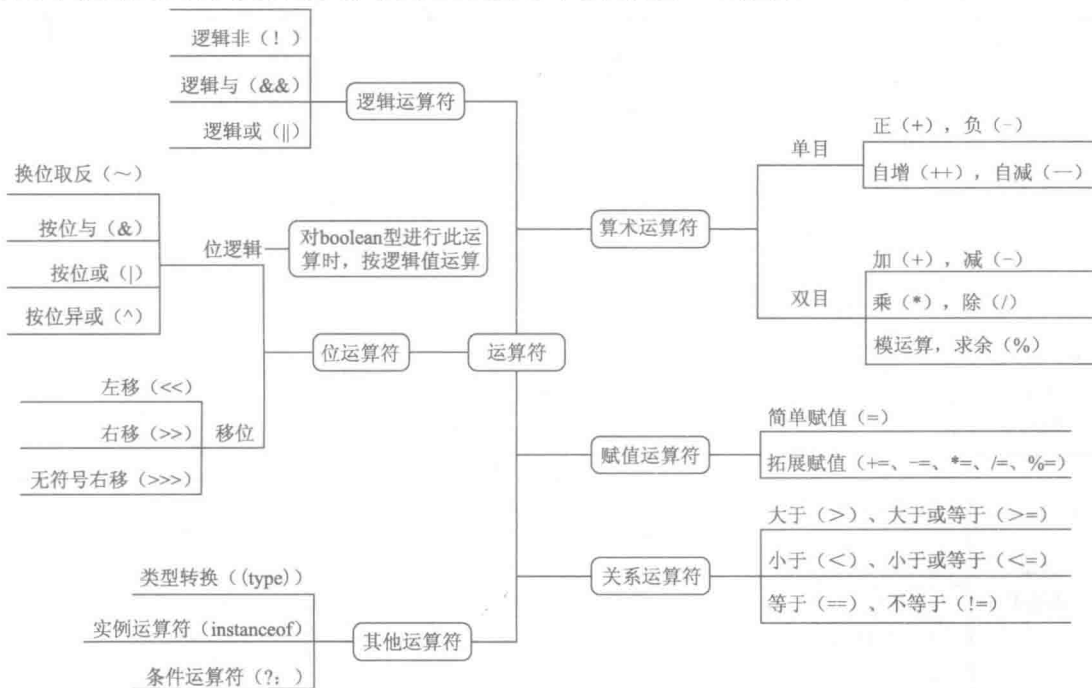


图 1-2 Java 运算符导学图

不同的运算符有不同的优先级，运算符的优先级决定了表达式中运算执行的先后顺序。Java 运算符的优先级见表 1-4。

表 1-4 Java 运算符的优先级

运算符	描述	优先级
. [] ()	域、数组、括号	1
++ -- - ! ~	一元运算符	2
* / %	乘、除、取余	3
+ -	加、减	4
<< >> >>>	移位运算	5
< <= > >=	关系运算	6
== !=	关系运算	7
&	位与	8
^	位异或、异或	9
	位或	10
&&	逻辑与（短路与）	11
	逻辑或（短路或）	12
?:	条件运算	13
= *= /= %= += -= ^=	赋值运算	14
&= = <<= >>= >>>=	赋值运算	15

4. 表达式

表达式是由常量、变量、运算符组成的有一定含义的式子，运算结果即为表达式的值，表达式是程序代码的重要组成部分。典型表达式示例如表 1-5 所示。

表 1-5 典型表达式示例

分类	表达式示例	运算结果	分析说明	
算术表达式	double x=(1+2+3)/4.0;	x=1.5	除数用 4.0 代表是 double 型的，保留小数	
	double y=(1+2+3)/4;	y=1.0	参与运算的都是 int 型值，忽略了小数	
	double z=(double)(1+2+3)/4;	z=1.5	将分母进行了强制类型转换，保留小数	
	int j=5, i=2, k, m; k=i+j++; //自增符++在后 m=--i+j; //自减符--在前	k=7, j=6 m=6, i=1		++、--符号在变量前或后，运算顺序不同： ++在变量 j 之后，j 先参加运算，再自增； --在变量 i 之前，i 先自减，再参加运算
	int i=5, j, k; j=i%3; //模运算，求余数 k=i/3; //除运算，求商	j=2 k=1		% 是模运算，结果是取余数； / 是除法，结果是商，两个整数相除时会忽略小数
关系表达式	char i='A', j='a'; System.out.println(i==j); System.out.println(i<j);	false true	关系表达式的操作结果是严格的逻辑类型，即只能是 true 或者是 false。 对于字符，==、<是按编码大小来比较	
逻辑表达式	int age=25; (age>20) && (age<30)	true	判断 age 变量的值是否满足条件“20<age<30”，满足，则表达式结果为 true，否则结果为 false。	
	(age>30) && (age<40)	false	在 a&&b 表达式中，当 a 的值是 false 时，不再计算 b，直接得出结果是 false	
	char ch='A'; (ch='B') (ch='b') (ch='A') (ch='a')	false true	判断 char 型变量 ch 是否等于字符'B'或'b'，是则表达式结果为 true，否则结果为 false。 在 a b 表达式中，当 a 的值是 true 时，不再计算 b，直接得出结果是 true	

续表

分 类	表达式示例	运 算 结 果	分 析 说 明
位 表达式	short a=15, b=2, x, y; x=~a; y=a^b;	x=-16 y=13	a: 00000000 00001111 ~a:11111111 11110000 (-16的补码) a:00000000 00001111 b:00000000 00000010 a^b:00000000 00001101
条件 表达式	int a=10, b=20, x; x= a > b ? 1 : 2 ;	x=2	若条件表达式的结果为 true, 则返回冒号前的值; 若为 false, 则返回冒号后的值
位移 表达式	short a=15, b=2, x, y, z; x=a<<b; y=a>>b; z= a>>>b;	x=60 y=3 z=3	a<<b: a 向左移动 b 位, 空位补 0 a>>b : a 向右移动 b 位, 空位补符号位, 正数补 0, 负数补 1 a>>>b: a 向右移动 b 位, 移后空位均用 0 填充

1.2.4 流程控制语句

Java 程序中的基本控制结构有三种: 顺序结构、分支结构(又称选择结构)、循环结构。下面主要介绍后两种。

1. 分支结构

分支结构是指可以从多个执行路径中选择其中之一去执行的流程控制结构。选择不同分支路径的依据是条件表达式的取值, 当条件表达式的值为 true, 执行其中一个分支; 当条件表达式的值为 false 时, 执行另一个分支。分支结构分类及语句格式见表 1-6。

表 1-6 分支结构

分 类	语 句 格 式	说 明
if 单分支结构	if(条件表达式) { 语句块; // if 分支 }	条件表达式为真时执行; 当只有一条语句时可以不加大括号
if-else 双分支结构	if(条件表达式) { 语句块; // if 分支 } else { 语句块; //else 分支 }	条件表达式为真时执行 if 分支; 条件表达式为假时执行 else 分支
if-else if-else 多分支结构	if(条件表达式 1) { 语句块 1; } else if(条件表达式 2) { 语句块 2;	条件表达式 1 为真时, 执行语句块 1; 条件表达式 2 为真时, 执行语句块 2; ... 条件表达式 n 为真时, 执行语句块 n; 其他条件, 执行 else 分支

续表

分 类	语 句 格 式	说 明
if-else if-else 多分支结构	<pre> } ... else if(条件表达式 n) { 语句块 n; } else { 语句块; //else 分支 } </pre>	
switch-case 多分支结构	<pre> switch(表达式) { case 判断值 1: 语句块 1; case 判断值 2: 语句块 2; ... case 判断值 n: 语句块 n; default: 语句块 n+1; } </pre> <p>注意：一般情况下，各语句块中应使用 break 语句让流程跳转出来，否则可能出现意外结果</p>	<p>(1) 表达式只能返回 int、short、byte 和 char 型的值；case 子句中的判断值必须是常量；</p> <p>(2) 多分支语句把表达式返回的值依次与每个 case 子句中的值相比较，如果遇到匹配的值，则执行该 case 语句块；当表达式的值与任一 case 子句中的值都不匹配时，程序执行 default 后面的语句，default 子句可选；</p> <p>(3) case 分支中包含多条语句时，可以不用大括号 {} 括起来；</p> <p>(4) 如果没有 break 语句，当程序执行完匹配的 case 语句序列后，还会继续执行后面的 case 语句序列</p>

2. 循环结构

循环结构用于重复执行某些操作，它是程序代码中非常重要、使用频率极高的语句。循环语句有三种：while 语句、do-while 语句和 for 语句。循环结构分类及语句格式见表 1-7。

表 1-7 循环结构

分 类	语 句 格 式	说 明
while 循环	<pre> while(条件表达式) { 循环体; } </pre>	<p>实现条件循环，先判断表达式，当条件为真时，重复循环，否则退出循环。</p> <p>若循环体有多行，则必须使用花括号将其括起来。这种循环可能一次都不执行</p>
do-while 循环	<pre> do { 循环体; }while(条件表达式); </pre>	<p>先执行一次循环体，再进行判断，当条件为真时，重复循环，否则退出循环。</p> <p>这种循环不管条件是否成立，至少会执行一次循环体</p>
for 循环	<pre> for(表达式 1; 表达式 2; 表达式 3) { 循环体; } </pre>	<p>表达式 1：给循环变量赋初值，表明循环的起始位置；</p> <p>表达式 2：用于判断循环停止时的条件，若条件满足则继续执行循环体，否则跳出；</p> <p>表达式 3：循环步长，控制循环变量每次循环时按什么方式变化；</p> <p>三个语句之间必须使用英文分号分隔。</p> <p>若执行的语句有多行，则必须用花括号将循环体括起来</p>

1.2.5 Java 的数组

数组是有序数据的集合，是一种引用数据类型。数组中的每个元素具有相同的数据类型，它们既可以是基本数据类型，也可以是引用数据类型。

1. 一维数组的创建和使用

1) 数组声明

创建数组时要先声明数组的名称和数组所包含元素的数据类型。声明的语法：

格式 1： 数组元素类型 数组名 []；

格式 2： 数组元素类型 [] 数组名；

方括号[]是数组的标志，推荐采用格式 2。

例如，一维数组：`int[] myArray;`

```
Car[] myCar;
```

2) 创建数组空间

定义好的数组必须创建数组空间后才可以使用。Java 不支持变长数组，所以在创建数组空间时必须指明数组的长度，以确定所开辟内存空间的大小。数组长度属性 `length` 取第一维的长度值。语法格式如下：

```
数组名=new 数组元素类型[数组元素的个数]
```

例如：`myArray=new int[10];`

```
myCar=new Car[15];
```

数组声明可以和创建数组空间同时完成，例如：

```
int[] myArray=new int[10];
```

```
Car[] myCar=new Car[15];
```

如果数组元素类型是基本数据类型，可以在创建数组空间的同时赋初值，例如：

```
int[] myArray={13,21,3,40,25,16,71,48,59,10 }
```

3) 创建数组元素并初始化

对于以某一类对象为数组元素的数组，则必须创建并利用构造函数初始化每个数组元素，一般采用循环结构来完成。若数组元素类型为基本数据类型，则采用直接赋值方式。例如：

```
for(int i=0;i<mycar.length;i++)
```

```
{
```

```
    mycar[i]=new Car("奥迪",1000+i);           //出厂流水号 10001~10001
```

```
}
```

2. 二维数组的创建和使用

二维数组的声明方式与一维数组的声明类似，使用两个[]标志即可。例如，创建一个 2 行 3 列的 `int` 型二维数组，可以采用以下三种方法之一：

方法 1：`int a[][];`

```
a=new int[2][3];
```

方法 2：`int [][] a=new int[2][3];`

方法 3：`int a[][]={{1,2,3},{4,5,6}};`

Java 是基于一维数组的原理来实现二维数组的。二维数组 `a` 可以看作一个一维数组，拥有数