



新世纪应用型高等教育
软件专业系列规划教材

Java 核心编程技术 实验指导教程

新世纪应用型高等教育教材编审委员会 组编

主 编 张 屹 蔡木生



大连理工大学出版社



新世纪

新世纪应用型高等教育
软件专业系列规划教材

Java 核心编程技术 实验指导教程

新世纪应用型高等教育教材编审委员会 组编

主编 张 屹 蔡木生

副主编 聂常红 黄玲玲 吴向荣

邹立杰 林若钦



10101010

10001101010

大连理工大学出版社

图书在版编目(CIP)数据

Java 核心编程技术实验指导教程 / 张屹, 蔡木生主编. — 大连: 大连理工大学出版社, 2010. 10
新世纪应用型高等教育软件专业系列规划教材
ISBN 978-7-5611-5839-5

I. ①J… II. ①张… ②蔡… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2010)第 197263 号

大连理工大学出版社出版

地址: 大连市软件园路 80 号 邮政编码: 116023

发行: 0411-84708842 邮购: 0411 84703636 传真: 0411-84701466

E mail: dutp@dutp. cn URL: <http://www.dutp.cn>

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸: 185mm×260mm 印张: 12.5 字数: 289 千字
印数: 1~2000

2010 年 10 月第 1 版 2010 年 10 月第 1 次印刷

责任编辑: 潘弘喆

责任校对: 林宜竹

封面设计: 张莹

ISBN 978-7-5611-5839-5

定 价: 25.00 元

前 言

本书是与《Java 核心编程技术》配套的实验指导教程,共有 15 个实验,分别对应着各章内容。每个实验包含:实验目的、主要知识点、实验内容与步骤、实验总结四部分内容,其中:“实验目的”说明了实验的主要内容和要求,通常有掌握、熟悉、了解等不同层次要求;“主要知识点”列举了本次实验相关的主要知识点,这是实验的基础和灵魂;“实验内容与步骤”给出实验的具体内容和操作步骤,题型有程序填空、分析、运行、改错、编程等,为加深读者对相关知识点的理解,特别提供了一系列小问题让读者思考、探究;“实验总结”则是对实验的重要性、重点、难点进行说明。

怎样使用这一实验教程呢?这里,我们给出几点建议:(1)关于实验的课时数:如果 Java 课程是开设 1 学期,每个实验可安排在 1 周完成,当然也可以根据各自要求进行增加或删减;若是开设 2 学期,则可以将一些实验扩展在 2~3 周完成;(2)关于实验环境:请按第 1 章要求下载、安装 JDK,并进行适当配置,强烈推荐安装 Eclipse,这是当今最为流行的 Java 开发软件,有助于提高编程效率;(3)关于课前预习:读者在上机实践前,应对“主要知识点”的内容熟悉、掌握,这是实验的前提。如果达不到这一要求,应结合教材及时复习相关知识点;(4)关于上机实践时出现的问题:上机实践时出现问题和麻烦,这是很正常的现象,即使是专家、教授也不可避免。面对问题的正确态度是正视它、解决它,调试程序也是一种能力,这种能力需要在上机实践中去提升。Java 中涉及的类、接口很多,我们没必要也不可能全部记住它们,但应该能通过 Java API 文档和 Eclipse 提示信息快速找到自己所要的内容,这是程序员的一项基本功,应予以重视;(5)关于编程能力的培养:常听一些学生说,我能看懂别人的程序,但自己不会写,这是编程能力不强的表现。要提高编程能力,最重要的是理清编程思路,并进行一定题量的训练。阅读、分析他人代码,理解其编程思路是一种非常有效的途径,在

此基础上模仿、改进、创新,日积月累才能提高编程水准。全书共有 120 多个题目,能保证一定的编程训练量。

本书由广州大学华软软件学院的几位老师共同编写,其中:实验 1、实验 2、实验 4、实验 9、实验 13 由蔡木生编写,实验 8、实验 10、实验 11、实验 14、实验 15 由黄玲玲编写,实验 3、实验 5、实验 6 由聂常红编写,实验 7、实验 12 由林若钦编写。全书由张屹、蔡木生统稿。

由于编者水平有限,编写时间仓促,书中疏漏在所难免,恳请读者批评、指正。

所有意见和建议请发往:gzjckfb@163.com

欢迎访问我们的网站:<http://www.dutpgz.cn>

联系电话:0411—84707492 84706104

编 者

2010 年 7 月



实验 1	Java 入门	1
1.1	实验目的	1
1.2	相关知识点	1
1.3	实验内容与步骤	3
1.4	实验总结	6
实验 2	Java 编程基础	8
2.1	实验目的	8
2.2	相关知识点	8
2.3	实验内容与步骤	13
2.4	实验总结	20
实验 3	类与对象	21
3.1	实验目的	21
3.2	相关知识点	21
3.3	实验内容与步骤	25
3.4	实验总结	38
实验 4	常用类	39
4.1	实验目的	39
4.2	相关知识点	39
4.3	实验内容与步骤	40
4.4	实验总结	44
实验 5	继承和多态性	46
5.1	实验目的	46
5.2	相关知识点	46
5.3	实验内容与步骤	48
5.4	实验总结	64
实验 6	抽象类与接口	65
6.1	实验目的	65
6.2	相关知识点	65
6.3	实验内容与步骤	66
6.4	实验总结	76
实验 7	异常处理	77
7.1	实验目的	77
7.2	相关知识点	77
7.3	实验内容与步骤	78
7.4	实验总结	84

实验 8 Java 泛型与 Java 集合	85
8.1 实验目的	85
8.2 相关知识点	85
8.3 实验内容与步骤	86
8.4 实验总结	94
实验 9 文件与输入输出流	95
9.1 实验目的	95
9.2 相关知识点	95
9.3 实验内容与步骤	98
9.4 实验总结	105
实验 10 图形用户界面设计	106
10.1 实验目的	106
10.2 相关知识点	106
10.3 实验内容与步骤	110
10.4 实验总结	126
实验 11 事件处理	127
11.1 实验目的	127
11.2 相关知识点	127
11.3 实验内容与步骤	129
11.4 实验总结	140
实验 12 小应用程序—Applet	141
12.1 实验目的	141
12.2 相关知识点	141
12.3 实验内容与步骤	142
12.4 实验总结	149
实验 13 数据库编程	150
13.1 实验目的	150
13.2 相关知识点	150
13.3 实验内容与步骤	154
13.4 实验总结	159
实验 14 多线程	160
14.1 实验目的	160
14.2 相关知识点	160
14.3 实验内容与步骤	162
14.4 实验总结	172
实验 15 网络编程	173
15.1 实验目的	173
15.2 相关知识点	173
15.3 实验内容与步骤	176
15.4 实验总结	193

实验 1 Java 入门

1.1 实验目的

1. 了解 Java 开发环境建立的必要性;
2. 学会 JDK 的下载与安装;
3. 熟悉环境变量的配置与测试;
4. 掌握命令行环境下 Java 程序的编译及运行方法;
5. 熟悉 Eclipse 集成环境下 Java 项目的创建、Java 程序的编辑、编译与运行方法;
6. 了解 Java 程序的两种类型:Java 应用程序和 Java 小应用程序;
7. 能够模仿书中例子,编写简单的 Java 应用程序,并运行;
8. 能够模仿书中例子,编写简单的 Java 小应用程序,并运行。

1.2 相关知识点

1. Java 程序可分为 Application 和 Applet(或 JApplet,下同)两种类型,尽管两者在程序格式、运行方式等方面存在差异,但它们也有共同点:先用文本编辑器得到扩展名为 java 的源文件,再编译成扩展名为 class 的类文件,如图 1-1 所示。

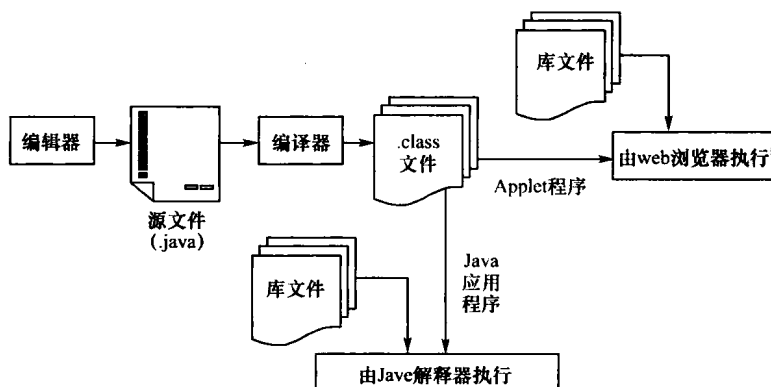


图 1-1 两类 Java 程序的执行流程

从上可知,创建 Java 开发环境是非常必要的,在此基础上才能进行程序的编写、编译和运行。

2. Java 源程序可以用文本编辑器(如记事本、写字板等)生成,专业的集成开发环境

(如 Eclipse)使用方便、效率更高。Java 程序的编译、运行通常需要安装 JDK。JDK 的下载、安装比较简单,只要从 SUN 公司网站下载相应文件,并在向导指引下一步一步完成即可,不过有两项工作很重要,请务必重视:

(1)熟悉 JDK 目录的内容:包括 bin、lib、db、demo、jre、include、sample、src 等子目录,特别是 bin 目录中包括的工具程序,如:javac. exe、java. exe、javadoc. exe、appletviewer. exe 等;

(2)环境变量的配置:现假设 JDK 的安装目录为 C:\jdk1.6.0,主要有三个环境变量需要配置:

①变量名:JAVA_HOME,变量值:C:\jdk1.6.0。这一变量指明了 JDK 的安装位置,利用该变量,会使其他环境变量值的书写更简洁;当 JDK 的安装目录发生变化时,只要修改它的值,其他设置不需要改变;

②变量名:PATH,变量值:…;%JAVA_HOME%\bin,…表示已设置的搜索路径,相互间用“;”隔开,当然也可以不使用 JAVA_HOME 环境变量,而直接写 bin 的绝对路径名。该变量的设置主要是为了让系统找到 Java SE 所提供的工具程序,而不需要每次使用时,都指定完整的路径名称;

③变量名:CLASSPATH,变量值:.(点号,表示当前目录)。这一变量的设置是为 Java 程序编译、运行时所用,依据该变量,工具程序、JVM 等能找所需要的类或接口。现在,位于 jre/lib 目录下的 rt.jar 和 i18n.jar 及 jre/lib/ext 目录中的 jar 包都不需要在 CLASSPATH 中设置,应用程序就能找到它们;如果是读者自己的 jar 包或使用第三方的 jar 包,则需要设置。

在 Windows 系统中可右击“我的电脑”,点击“属性/高级/环境变量”来配置这三个环境变量。环境变量的配置是否正确,可通过运行 bin 目录的工具程序予以检验。

3. Java 程序的两种类型

(1)Java 应用程序,是一种能在支持 Java 的平台上独立运行的程序,通过 JVM(Java 虚拟机)解释执行。Java 应用程序中用 public 修饰的类称为主类,主类一定包含 main()方法,它是应用程序的主要入口。包含主类的源程序命名有特殊规定,主文件名与主类名相同,扩展名为 java,严格区分大小写。Java 应用程序的格式如下:

```
public class 类名 {
    //属性的声明
    public static void main(String [] args){
        //程序代码
    }
    //其他方法的定义
}
```

(2)Java 小应用程序,是一种通过<applet>…</applet>标记将字节码文件(即 class 文件)内嵌在 html 文档中,由支持 Java 的浏览器来运行的文件。小应用程序没有 main()方法,其格式如下:

```
import java. applet. * (或 javax. swing. * );
import java. awt. * ;
```

```
public class 类名 extends Applet(或 JApplet) {  
    //属性的声明  
    public void init() { //初始化方法  
        //程序代码  
    }  
    public void paint(Graphics g) { //显示结果方法  
        //程序代码  
    }  
    //其他方法  
}
```

嵌入小程序的 html 文档格式如下:

```
<html>  
    <head><title>...</title></head>  
    <body>  
        <applet code=类名.class width=宽度 height=高度>  
        </applet>  
    </body>  
</html>
```

4. Java 程序的编译、运行

前面说过,任何一个文本编辑器都可以编辑 Java 文件,不过要注意主文件名应与主类名相同,扩展名为 java。编译、运行有以下两种主要方式:

(1) 命令行方式:例如在 Windows 系统中,点击主菜单中的“运行”,输入 cmd,再按回车键。

- ①编译: javac 源程序.java;
- ②Java 应用程序的运行: java 主类名;
- ③Java 小应用程序的运行: appletviewer html 文档。

(2) 集成开发环境:提供强大的功能,集编辑、编译、运行于一体,现以流行的 Eclipse 为例来说明:

- ①创建 Java 项目:打开菜单【File】|【New】|【Java Project】,输入项目名、存放目录等内容;
- ②创建 Java 类:打开菜单【File】|【New】|【class】,输入类名、包名等,编辑文件;
- ③编译、运行:先选定源程序,再点击菜单【Run】(或向右三角形或快捷菜单【Run As ...】等),再选择 Java Application(或 Java Applet),若还需要输入命令行参数,则进一步选择【Open Run Dialog...】,提供一个或多个参数。

1.3 实验内容与步骤

1. JDK 的下载与安装

- (1) 从 Sun 公司的站点 <http://java.sun.com> 选择适合 Windows 的 JDK 最新版本

(如 JDK 6)进行下载,并将其保存在某一目录中(如 D:\java);

(2)运行下载所得到的文件 jdk-6u18-windows-i586-p.exe,更改 JDK 的安装目录(如 C:\jdk1.6.0);

(3)进入 JDK 的安装目录,查看 bin、lib、jre、demo 等子目录内容。

2. 环境变量的配置与测试(假设 JDK 的安装目录为 C:\jdk1.6.0)

(1)从“我的电脑/属性/高级/环境变量”(或“开始/设置/控制面板/系统/高级/环境变量”)打开环境变量配置窗口,如图 1-2 所示。

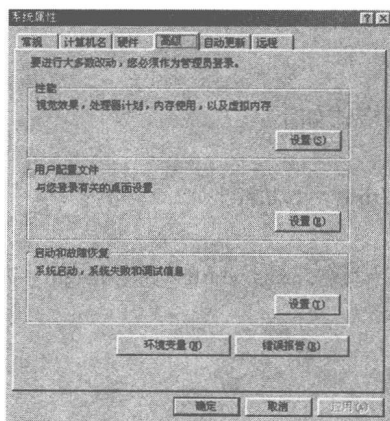


图 1-2 环境变量的配置

(2)点击“新建”按钮,创建系统变量 JAVA_HOME,如图 1-3 所示。

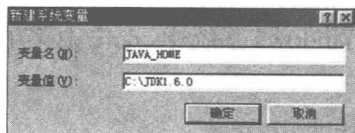


图 1-3 新建系统变量 JAVA_HOME

(3)如果系统变量 PATH 已存在,则点击“编辑”按钮,修改 PATH 的变量值,在变量值最前面添加%JAVA_HOME%\bin,如图 1-4 所示。

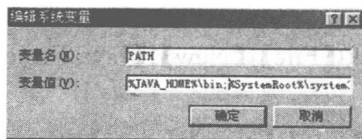


图 1-4 编辑系统变量 PATH

假设,系统变量 PATH 原先不存在,就新建一个,方法如(2)。

(4)点击“新建”按钮,创建系统变量 CLASSPATH,一定要加上“.”(点号),表示当前目录,如图 1-5 所示。

(5)结果如图 1-6 所示,点击“确定”按钮保存。

(6)测试环境变量

①点击“开始”菜单/运行...,输入 cmd,并按回车键,进入命令行环境;

②输入并执行下列命令:

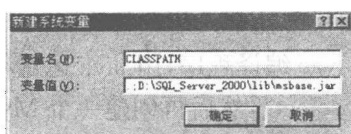


图 1-5 新建环境变量 CLASSPATH



图 1-6 环境变量配置结果

```
java - version
```

```
javac
```

若出现如图 1-7 所示的画面,说明环境变量配置正确;否则,可能存在问题。

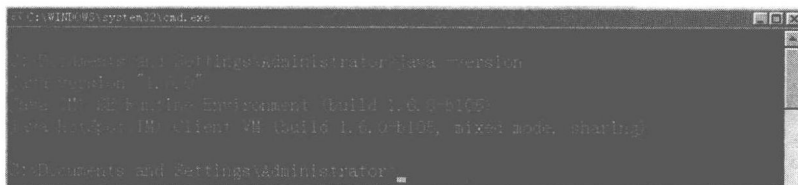


图 1-7 运行 java - version 命令后的画面

问题:

- ①在命令行环境下,如何改变当前盘符、当前目录,显示当前目录的文件和子目录,清除屏幕,快速调用前面使用过的命令?
- ②在命令行环境下,如何使用剪贴板进行“剪切”、“复制”、“粘贴”操作?
- ③如何退出命令行环境?

3. 请用“记事本”(或“写字板”)等文本编辑器,根据自己的情况输入下列内容,并以 MySelf. java 命名保存到 D:\myjava 目录中:

```
public class MySelf {  
    public static void main(String [] args){  
        System.out.println("姓名:"+"XXXXX");  
        System.out.println("学号:"+"XXXXX");  
        System.out.println("专业:"+"XXXXX");  
    }  
}
```

说明:程序中的 XXXXX 用自己的个人信息替代。

然后在命令行环境下,用 javac 编译程序,用 java 运行程序。

4. 在 Eclipse 中创建项目 project1,然后再创建一个 MySelf 类,程序内容与上一题相同,再编译、运行程序。比较一下,集成开发环境与命令行环境的区别。

5. 参考教材例题,编写一个应用程序:输入长方形的长度、宽度,计算长方形的周长、面积。

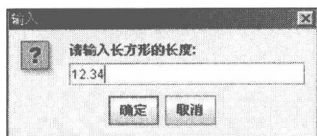


图 1-8 输入长方形的长度

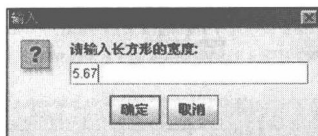


图 1-9 输入长方形的宽度

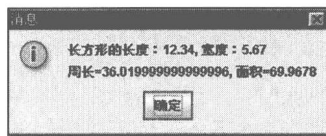


图 1-10 输出计算结果

6. 参考教材例题,编写一个小应用程序,输出一张图片或自己的照片,类似于图1-11。



图 1-11 小应用程序输出的图片

7. 运行指定目录下的 Java 程序

(1) 在 JDK 安装目录下有 demo\jfc 子目录,其中有多多个应用程序,例如:Notepad、Java2D、SwingSet2 等,请用鼠标双击扩展名为 jar 的文件,了解一下 Java 应用程序的功能,有兴趣的读者还可以了解一下位于 src 目录下的源代码;

(2) 运行 JDK 安装目录下的 demo\applets 子目录下的 Clock 程序。

8. Java 学习的一些网站

(1) Java 技术网 <http://java.sun.com>

(2) SUN 中国技术社区 <http://gceclub.sun.com.cn>

(3) IBM Java 技术新手入门 <http://www.ibm.com/developerworks/cn/java/newto>

(4) Java 中文站 <http://www.java-cn.com>

(5) 中国最大的程序员网 <http://www.csdn.net>

1.4 实验总结

本实验的内容是 Java 初学者应该掌握的,包括:Java 开发环境的搭建与测试,Java 程

程序的编辑、编译、运行,Java 程序的分类。要搭建 Java 开发环境,首先要下载 JDK,然后配置有关环境变量,例如:JAVA_HOME、PATH、CLASSPATH 等,可通过运行 java 等有关命令来测试;使用任何一个文本编辑器都可以编写 Java 源程序,不过要注意其命名要求;可在命令行环境中使用 javac、java 等编译、运行 Java 程序,Eclipse 等集成环境集编辑、编译、运行于一体,功能强大、使用方便、效率更高,应逐渐熟悉;Java 程序可分为 Application 和 Applet 两种类型,平时使用最多的还是 Application,Application 和 Applet 既有相同点,也有不同点,要注意区分。尽管有关 Java 程序设计的许多知识有待于后续章节来学习,但是模仿例题,也应能编写、运行一些简单的 Java 程序。我们常说:良好的开端是成功的一半,只要把基础打牢,循序渐进,掌握 Java 并不是一件难事。

实验 2 Java 编程基础

2.1 实验目的

1. 熟悉基本的数据类型,包括其所占字节数、数值范围及常数的后缀形式;
2. 掌握变量的声明、初始化方法,能够正确区分全局变量(即成员变量)和局部变量(即本地变量);
4. 掌握算术运算符(+、-、*、/、%) and 自增(++)、自减(--)运算符的使用;
5. 掌握关系运算符(>、>=、<、<=、==、!=)和逻辑运算符(!、&&、||)的使用;
6. 掌握 if...else...语句的单分支、双分支、多分支结构的使用;
7. 掌握 switch 语句的使用;
8. 熟悉循环结构的初始化部分、条件判断部分、修改条件部分、循环体部分的功能;
9. 掌握 for 语句的使用,熟悉 for 语句的简单应用(累加、连乘积、分类统计);
10. 掌握 while 语句的使用;
11. 掌握 do...while 语句的使用,能够正确区分 do...while 与 while 的不同之处;
12. 熟悉 Java 程序的常用输入输出格式;
13. 掌握一维数组的概念:一维数组的声明,创建,初始化,元素的访问;
14. 掌握多维数组的概念:二维数组的声明,创建,初始化,元素的访问。

2.2 相关知识

1. Java 中的数据类型可分为基本数据类型和引用数据类型两大类,其中:基本数据类型包括 byte、short、int、long、char、float、double、boolean 8 种,引用数据类型包括类、接口、数组等。熟悉基本数据类型的关键字、所占字节、数据范围、使用场合及数据类型的相互转化十分重要,这是编程的基础。

2. Java 中的关键字有 50 多个,它们是一些有特殊含义的单词,只能按系统规定的方式来使用,不能单独用作标识符。包名、类名、接口名、方法名、对象名、常量名、变量名等统称为标识符,对于其命名 Java 有严格规定:必须以字母(严格区分大小写)、下划线(_)、美元符号(\$)开头,后续字符除了这三类之外,还可以是数字、中文字符、日文字符、韩文字符、阿拉伯字符,命名尽可能规范,做到“见名知义”。顾名思义,变量是指其值在程序运行中可改变的量,Java 是强类型语言,变量要“先声明、后使用”。在一条声明语句中,可以声明多个变量,也可以赋初值(称为初始化)。变量在运算前应初始化或赋值,例如: `int count; count=count+1;` 如果 count 是全局变量(即类的成员变量)且未初始化,则取默

认值(例如:int 型为0);如果 Count 是局部变量(即本地变量)且没有初始化,则编译时出错。Java 中的常量是一类特殊的变量,其值在程序运行期间不会改变,符号常量用 final 关键字来定义,常量名通常用大写,且多个单词之间用下划线连接,例如:Math 类的 PI 表示圆周率的近似值,Byte 类的 MIN_VALUE 与 MAX_VALUE 分别表示 Byte 类型的最小值和最大值。

3. 运算符指明了操作数将要进行的运算。根据操作数的数目,运算符可分为单目运算符、双目运算符、三目运算符,从功能上可分为:算术运算符、自增自减运算符、关系运算符、逻辑运算符、位运算符、移位运算符、赋值运算符、条件运算符等:

(1)算术运算符:包括+、-、*、/和%,请注意“/”运算符在整数运算与浮点数运算的差异,char 类型有整数类型的特征,能进行算术运算;

(2)自增自减运算符:包括++、--,根据是“先运算后赋值”,还是“先赋值后运算”可分为前缀、后缀两种不同形式;

(3)关系运算符:包括>、>=、<、<=、==、!=,运算结果为 true 或 false;它们可分成三组互为相反的运算,只要知道其中一个运算式的值,与之相反的式子的结果也自然确定;

(4)逻辑运算符:包括!(非)、&&(与)、||(或),它们是在逻辑运算的基础上进行的连接运算,运算结果仍为 true 或 false,不过要注意它们运算的优先级不同;在运算结果能够确定的情况下,一些式子可能不再进行运算,即出现“短路”现象;

(5)位运算符:包括~(非)、&(与)、|(或)、^(异或),这是在二进制“位”的基础上进行的运算,运算结果仍为整数,^(异或)有特殊用途;

(6)移位运算符:包括<<(左移)、>>(带符号右移)、>>>(无符号右移),这也是在二进制基础上进行的运算,左移后在右边一律加0,带符号数右移后则根据其正负来决定补什么内容,正数左边补0,负数左边补1,无符号右移左边补0;

(7)赋值运算符:包括=、+=、-=、*=、/=、%=、&=、|=、!=、<<=、>>=等复合赋值运算符,它们的运算顺序为从右到左;

(8)条件运算符:运算符是?,这是唯一的一个三目运算符。

当一个式子中出现多个不同的运算符时,应按照运算符的优先级进行运算。

4. 运算符将操作数连接起来构成表达式,在表达式的尾部加上分号(;)就形成了一条语句,语句是构成程序的基本单位,程序就是由一条一条的语句组合而成的。为了增加程序的可读性,可使用注释语句。Java 程序的注释语句有三种格式:单行注释(以//开始,到行尾结束)、多行注释(以/*开始,到*/结束,可以跨越多行文本内容)、文档注释(以/**开始,中间行以*开头,到*/结束)。编写源程序时添加必要的注释,这是程序员应该具备的基本素养。

5. Java 程序的三种基本结构:顺序结构、选择结构和循环结构

(1)顺序结构:按照书写顺序从上到下、逐条执行的,这是最简单的结构。

(2)分支结构:根据条件是否满足来选择某一语句的执行,可分为以下几种类型:

①if 单分支结构

②if...else...双分支结构

```

if (条件表达式) {
    语句
}

if (条件表达式){
    语句 1
} else {
    语句 2
}

```

③if...else...多分支结构

```

if (条件 1) {
    语句 1
} else if (条件 2) {
    语句 2
} else if (条件 3) {
    语句 3
} else {
    语句 3
}

if (条件 1) {
    语句 1
} else {
    if (条件 2) {
        语句 2
    } else {
        语句 3
    }
}

if (条件 1) {
    语句 1
} else if (条件 2){
    语句 2
} else if (条件 3) {
    语句 3
} else if (...)
    ...
} else { //条件 1、条件 2、...均不满足
    ...
}

```

④switch 多分支结构

```

switch (表达式){
    case 常量 1:
        语句 1;
        break;
    case 常量 2:
        语句 2;
        break;
    ... ...
    default:
        语句 n+1;
        break;
}

```

(3)循环结构

在高级语言中,可以将反复执行的语句用循环语句来实现。与 C++ 类似,Java 中的循环有三种基本形式:

①for 语句

```

for (表达式 1 ; 表达式 2 ; 表达式 3 ){
    循环体;
}

```

说明:该语句包含了“循环控制部分”和“循环体”两部分,其中,“循环控制部分”是由初始表达式(即表达式 1)、条件判断表达式(即表达式 2)、循环控制变量修改表达式(即表达式 3)三部分组成,通过循环控制变量来控制循环的执行次数;“循环体”是每次循环所要执行的语句或语句块,它可以访问或修改循环控制变量的值。