

21

21 世纪全国高校应用人才培养信息技术类规划教材



# Java语言程序设计 复习指导与习题解答

周少琦 沈泽刚 贾福龙 主 编



北京大学出版社  
PEKING UNIVERSITY PRESS

TP312/2888C

2008

21 世纪全国高校应用人才培养信息技术类规划教材

# Java 语言程序设计复习指导 与习题解答

周少琦 沈泽刚 贾福龙 主编



北京大学出版社  
PEKING UNIVERSITY PRESS

## 内 容 简 介

本书是《Java 语言程序设计》一书的配套复习指导与习题解答,书中包括9章的内容提要、疑难解析、129道问答思考题及解答和253道程序设计题及解答。

内容提要中提出各章的要点,便于读者从总体上把握全章内容,层次脉络一清二楚、应知应会心中有数。疑难解析中抽出各章重要而又有一定难度的问题从理论上深入分析、图形上生动描绘、示例上另辟蹊径,便于读者各个击破而后融会贯通。问答思考题及解答涵盖了教材中所有有一定思考性的理论问题和综合性的实践问题,每道题都做了细致的分析和简要的解答。全书以程序设计题为主,将教材中的所有程序设计题全部作了解答,并进行了细致的程序优化和界面美化。这些程序设计题可分为三个层次:让初学者尽快入门,让入门者尽快提高,让提高者施展才华、大显身手。

本书既可以作为大专院校有关专业Java语言程序设计课程教师布置作业、学生课下练习的辅助用书,也可以作为该课程自学人员的指导用书,还可以作为软件开发人员的参考用书。

### 图书在版编目(CIP)数据

Java语言程序设计复习指导与习题解答/周少琦,沈泽刚,贾福龙主编. —北京:北京大学出版社,2008.5

(21世纪全国高校应用人才培养信息技术类规划教材)

ISBN 978-7-301-13631-7

I. J… II. ①周… ②沈… ③贾… III. JAVA语言—程序设计—高等学校—教学参考资料  
IV. TP312

中国版本图书馆CIP数据核字(2008)第051242号

书 名: Java语言程序设计复习指导与习题解答

著作责任者: 周少琦 沈泽刚 贾福龙 主编

责任编辑: 温丹丹

标准书号: ISBN 978-7-301-13631-7/TP·0949

出版者: 北京大学出版社

地 址: 北京市海淀区成府路205号 100871

电 话: 邮购部 62752015 发行部 62750672 编辑部 62765126 出版部 62754962

网 址: <http://www.pup.cn>

电子信箱: [xxjs@pup.pku.edu.cn](mailto:xxjs@pup.pku.edu.cn)

印 刷 者: 河北滦县鑫华书刊印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787毫米×1092毫米 16开本 17.25印张 409千字

2008年5月第1版 2008年5月第1次印刷

定 价: 30.00元

---

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究

举报电话: 010-62752024; 电子信箱: [fd@pup.pku.edu.cn](mailto:fd@pup.pku.edu.cn)

# 前 言

计算机程序设计语言也是一种语言，如同学外语需要大量会话一样，学习计算机程序设计语言也必须有大量的与计算机会话的实践。与计算机会话的实践就是编写程序、上机调试程序、运行程序，并最终获得预期的结果，所以学习计算机程序设计语言需要大量的程序设计题目以供实践练习。Java 语言更是这样，类与对象的概念、封装与继承的机制、异常的处理机制、事件的处理模型等都需要有足够的程序设计题目的练习来认识、体会、总结，最终形成一个完整的知识体系。

本书是《Java 语言程序设计》一书的配套复习指导与习题解答，书中包括 9 章的内容提要、疑难解析、129 道问答思考题及解答和 253 道程序设计题及解答。

内容提要中提出各章的要点，简明扼要、一目了然。便于读者从总体上把握全章内容，层次脉络一清二楚、应知应会心中有数。

疑难解析中抽出各章重要而又有一定难度的问题从细剖析。限于篇幅，教材中不少问题难于展开说明、形象描述，此处则从理论上深入分析、图形上生动描绘、示例上另辟蹊径，便于读者各个击破而后融会贯通。

问答思考题及解答涵盖了教材中所有有一定思考性的理论问题和综合性的实践问题。每道题都做了细致的分析和简要的解答，所有解答都做到既全面周到又简短利落。

本书以程序设计题为主，将教材中的所有程序设计题全部作了解答，并进行了细致的程序优化和界面美化。初稿中程序设计题的参考答案按一行一句、逐层缩进 4 个字符的格式，为了减少篇幅，终稿中采用一行多句、逐层缩进两个字符的格式。这些程序设计题目与教材中的教学内容结合紧密，第 1 章后面原有的几道题与教学内容结合得不够密切被删去，又增加了几道与教学内容结合密切的题目，并按一般设计与方法设计分别列出，结果使程序设计题目的总数由 249 道增加到 253 道。

这 253 道程序设计题可以划分为三个层次，不同层次分别体现如下三个特点。第一，针对性，各章程序设计题的多数题目与教材中的各节内容一一对应或稍作补充，补充的内容都是前面各节讲过的内容，目的在于使读者融会贯通。第二，思考性，对应各章偏后的题目都略有一定的难度，要求读者深入透彻地掌握相关的内容，在小范围的实际应用中提高灵活运用所学知识的能力。第三，实用性，这方面主要指的是指定为综合性实验的程序设计题，它们在更大的范围内综合应用本章及前面各章所学过的知识，比教材中的示例要丰富和完善得多，都可以直接应用于相应的软件开发，有广泛的实际应用价值。

本书努力使初学者尽快掌握基本的 Java 语言程序设计的方法和技巧；让勤于思考的读者提高自己综合所学知识、解决实际问题的编程能力；更给兴趣浓厚、肯于钻研的读者提供了施展才华、大显身手的实践环境，让他们在紧张的劳作之后尽享成功的喜悦和胜利者的骄傲。这是笔者的初衷，但是由于我们的水平有限，书中疏漏和错误之处在所难免，欢迎广大读者批评指正，联系邮件可寄到 [zh\\_shqi1116@163.com](mailto:zh_shqi1116@163.com)。

周少琦

2008 年元月 20 日于渤海大学

# 目 录

第 1 章	Java 语言语法基础.....	1
1.1	复习指导.....	1
1.1.1	内容提要.....	1
1.1.2	疑难解析.....	5
1.2	复习题.....	6
1.2.1	问答思考题.....	6
1.2.2	程序设计题.....	7
1.3	复习题解答.....	9
1.3.1	问答思考题解答.....	9
1.3.2	程序设计题解答.....	12
第 2 章	面向对象理论基础.....	24
2.1	复习指导.....	24
2.1.1	内容提要.....	24
2.1.2	疑难解析.....	26
2.2	复习题.....	28
2.2.1	问答思考题.....	28
2.2.2	程序设计题.....	29
2.3	复习题解答.....	31
2.3.1	问答思考题解答.....	31
2.3.2	程序设计题解答.....	33
第 3 章	Java 类库语言包.....	52
3.1	复习指导.....	52
3.1.1	内容提要.....	52
3.1.2	疑难解析.....	54
3.2	复习题.....	55
3.2.1	问答思考题.....	55
3.2.2	程序设计题.....	55
3.3	复习题解答.....	57
3.3.1	问答思考题解答.....	57
3.3.2	程序设计题解答.....	61
第 4 章	Java 类库实用包.....	78
4.1	复习指导.....	78
4.1.1	内容提要.....	78
4.1.2	疑难解析.....	80

4.2	复习题 .....	82
4.2.1	问答思考题 .....	82
4.2.2	程序设计题 .....	82
4.3	复习题解答 .....	83
4.3.1	问答思考题解答 .....	83
4.3.2	程序设计题解答 .....	84
<b>第 5 章</b>	<b>Java 类库输入输出包 .....</b>	<b>105</b>
5.1	复习指导 .....	105
5.1.1	内容提要 .....	105
5.1.2	疑难解析 .....	107
5.2	复习题 .....	108
5.2.1	问答思考题 .....	108
5.2.2	程序设计题 .....	108
5.3	复习题解答 .....	110
5.3.1	问答思考题解答 .....	110
5.3.2	程序设计题解答 .....	112
<b>第 6 章</b>	<b>图形用户界面类库包 .....</b>	<b>136</b>
6.1	复习指导 .....	136
6.1.1	内容提要 .....	136
6.1.2	疑难解析 .....	139
6.2	复习题 .....	143
6.2.1	问答思考题 .....	143
6.2.2	程序设计题 .....	143
6.3	复习题解答 .....	145
6.3.1	问答思考题解答 .....	145
6.3.2	程序设计题解答 .....	148
<b>第 7 章</b>	<b>数据库程序设计 .....</b>	<b>185</b>
7.1	复习指导 .....	185
7.1.1	内容提要 .....	185
7.1.2	疑难解析 .....	187
7.2	复习题 .....	188
7.2.1	问答思考题 .....	188
7.2.2	程序设计题 .....	188
7.3	复习题解答 .....	189
7.3.1	问答思考题解答 .....	189
7.3.2	程序设计题解答 .....	191
<b>第 8 章</b>	<b>网络程序设计 .....</b>	<b>215</b>
8.1	复习指导 .....	215
8.1.1	内容提要 .....	215
8.1.2	疑难解析 .....	217

---

8.2	复习题	218
8.2.1	问答思考题	218
8.2.2	程序设计题	218
8.3	复习题解答	219
8.3.1	问答思考题解答	219
8.3.2	程序设计题解答	222
第9章	多媒体程序设计	236
9.1	复习指导	236
9.1.1	内容提要	236
9.1.2	疑难解析	239
9.2	复习题	240
9.2.1	问答思考题	240
9.2.2	程序设计题	241
9.3	复习题解答	242
9.3.1	问答思考题解答	242
9.3.2	程序设计题解答	247
	参考文献	266

# 第 1 章 Java 语言语法基础

## 1.1 复习指导

### 1.1.1 内容提要

#### 1. Java 语言概述

Java 语言是一种新型的面向对象的程序设计语言，所谓对象指的是人类行动或思考时作为目标的人或事物，所谓面向对象就是把客观存在的人或事物作为计算机程序设计的目标。“类”就是许多相同或相似的事物的综合，Java 语言中的类则是一种封装了变量和方法的复合数据类型。

Java 语言是一种准解释型的高级编程语言，所以 Java 语言程序的开发需要编写源程序、编译生成字节码文件和解释运行 3 个过程。编写源程序可以使用任何非排版的文本编辑软件，在编写 Java 语言源程序的时候，需要注意如下要求。

(1) Java 语言是区分大小写的语言，关键字的大小写不能搞错。

(2) Java 语言源程序中的每一个语句都要用分号结束，一行可以书写多条语句，每条语句都要用分号结束。

(3) 源程序编写好之后，要以文件的形式保存在硬盘或软盘上，称为源程序文件，或称为源文件。这个源文件的名字必须与程序的主类名相一致，所谓主类是指包含有 main 方法的类。这个源文件还要以 java 为其扩展名。

(4) Java 语言有 3 种注释形式：

① // 单行注释。注释从// 开始向后，直到行尾。

② /\*.....\*/ 多行注释。在/\*和\*/ 之间可以有多个行。

③ /\*\*.....\*/ 文档注释。所有在 /\*\*和\*/ 之间的内容可以用 Java 文档管理工具 javadoc 解析来生成应用文档。

(5) 源程序中要恰当地使用分隔符，空格、逗号、分号及行结束符都称为分隔符，Java 语言规定，任意两个相邻标识符、数字、保留字或两个语句之间必须至少有一个分隔符以便编译程序能识别。

执行编译的操作就是运行 Java 的编译器系统软件，Java 的编译器系统软件位于 JDK 中的文件夹 bin 内，名称为 javac.exe。运行的格式如下：

源程序所在目录>javac.exe 所在目录javac 源程序名.java

运行一个编译好的 Java 字节码程序，需要调用 Java 的解释器系统软件，Java 的解释器系统软件也位于 JDK 中的文件夹 bin 内，名称为 java.exe。运行的格式如下：

字节码文件所在目录>java.exe 所在目录java 源程序名（也是字节码文件名）



## 2. 基本数据类型

Java 语言中的基本数据类型共有 8 种, 分为 4 小类, 分别是整数类型、浮点类型、字符类型和布尔类型, 它们大多数是用标识符来标识的变量或常量。

在 Java 语言中书写标识符的规则如下。

- (1) 标识符必须由字母、下划线 (-)、美元符 (\$) 开始。
- (2) 其余字符可以是字母、下划线、美元符和数字。
- (3) 标识符是区分大小写的。
- (4) 标识符没有最大长度限制。

变量是程序中存储单元的标识符表示, 用来存放指定类型的数据, 其值在程序运行过程中是可变的。常量是在程序运行期间保持不变的量, Java 语言中的常量分为普通常量和标识符常量。

整数类型数据简称整型数据, 是指那些没有小数部分的数字。整型数据又分为 byte、short、int、long 4 种类型, 分别存储着由小到大的整数集合。浮点类型数据就是实数类型数据, 是指那些有小数部分的数字, 浮点型数据有单精度类型 float 和双精度类型 double 两种。字符类型数据就是单一的字符, Java 语言不采用通常计算机系统采用的 ASCII 代码集, 而是采用 Unicode 国际标准字符集。布尔类型数据用于表达两个对立的逻辑状态, 它只有两个值 true (真) 和 false (假)。

Java 语言中整型、实型、字符型数据可以混合运算。运算中, 不同类型的数据先转化为同一类型, 然后进行运算。按照优先关系, 转换分为两种: 自动类型转换和强制类型转换。按照优先关系, 低级类型数据要转换成高级类型数据时, 进行自动类型转换; 而要高级类型数据转换成低级类型数据, 则需用到强制类型转换。

## 3. 运算符及表达式

运算符就是表示各种运算的符号, 不同的运算符适用于不同的数据类型, 参与运算的数据称为操作数。Java 语言运算符按其要求的操作数个数分为: 单目运算符、双目运算符、三目运算符; 按其功能分为: 赋值运算符、算术运算符、关系运算符、逻辑运算符、位运算符、条件运算符等。

表达式是用运算符把操作数连接起来表达某种或某些运算过程的式子, 每个表达式经过运算后都会产生一个确定的值。

赋值运算符“=”用来把右边表达式的值赋给左边的变量, 即将右边表达式的值存放在左边变量名所表示的存储单元中。

算术运算符用于对整型数和实型数的运算, 按其要求的操作数的个数分为单目运算符和双目运算符两类。单目运算符只有一个操作数, 如 + 使操作数取正值、- 使操作数取负值、++ 使操作数加 1、-- 使操作数减 1。双目运算符要求两个操作数, 如 + 求两数和、- 求两数差、\* 求两数积、/ 求两数商、% 求两数相除的余数。算术表达式由操作数和算术运算符组成, 在算术表达式中操作数只能是整型或浮点型数据, 算术运算符作用于操作数, 完成算术运算。

关系运算符有 == (等于)、!= (不等于)、< (小于)、<= (小于等于)、> (大于)、>= (大于等于) 6 种, 在一个关系运算符两边的数据类型应一致。由两个操作数和关系运算符构成一个关系表达式, 一个关系表达式的结果类型为布尔型, 即关系式成立为 true, 不成立为 false。

逻辑运算符有 6 个，它们是：！（非）、&（与）、|（或）、^（异或）、&&（简洁与）和||（简洁或）。简洁运算（&&、||）与非简洁运算（&、|）的区别在于：非简洁运算在计算左右两个表达式之后再运算；简洁运算可能只计算左边的表达式而不计算右边的表达式即可取值。这些运算符要求的操作数和结果值都是布尔型。

位运算符用来对二进制位进行运算，位运算符的操作数只能为整型和字符型数据。Java 语言中定义了 7 种类型的位运算符：~（按位取反）、&（按位与）、|（按位或）、^（按位异或）、>>（有符号二进制右移一定位）、<<（有符号二进制左移一定位）、>>>（无符号二进制右移一定位）。移位运算符前是被移位操作数，移位运算符后是指定移位次数的操作数。

条件运算符是“?:”，它是 Java 中唯一的三目运算符。它要求三个操作数，其格式如下：

<布尔表达式>? <表达式 1>: <表达式 2>

第一个操作数必须是布尔表达式，其他两个操作数可以是数值型或布尔型表达式。条件运算符的含义是：当<布尔表达式>为真时，整个表达式值取为<表达式 1>的值，否则取值为<表达式 2>的值。

在赋值运算符“=”前加上其他运算符，即构成扩展赋值运算符。

表达式的运算次序取决于表达式中各种运算符的优先级。优先级高的先运算，优先级低的后运算。另外还可用括号“（）”改变表达式的运算次序，括号中的表达式首先被运算。

#### 4. 程序流程控制语句

程序流程控制语句是用来控制程序中各语句执行顺序的语句，Java 语言提供了以下几种流程控制语句：选择控制、循环控制和转向控制。

选择控制语句提供了这样一种控制机制，它根据某个特定表达式的值选择执行某一特定的语句序列，其他与特定表达式值不匹配的语句序列则被跳过不执行。选择控制语句分为条件选择语句和开关选择语句，条件选择语句又分为简单条件选择语句和多条件选择语句。

简单条件选择语句，即 if - else 语句可以根据判定条件的真假来选择两种操作中的一种。if - else 语句的语法格式为：

```
if(布尔表达式)    语句 1;
    [else        语句 2;    ]
```

if - else 语句中内嵌的语句 1 或语句 2 又是 if - else 语句的情况称为 if - else 语句的嵌套，这就是多条件选择语句。它的语法格式如下：

```
if(布尔表达式 1)    语句 1;
else if(布尔表达式 2) 语句 2;
.....
else if(布尔表达式 m) 语句 m;
[else 语句 n;]
```

开关选择语句就是 switch 语句，它根据表达式的结果来执行多个可能操作中的一个，它的语法格式如下：

```
switch(表达式){
    case 常量 1: 语句 1;    [break;]
    case 常量 2: 语句 2;    [break;]
    .....
```

```

case 常量 n: 语句 n;    [break;]
[default: 默认处理语句]    }

```

循环控制语句的作用是控制反复执行同一段代码直到满足结束条件为止，许多实际问题的解决需要用到循环控制结构。循环控制结构一般应包含以下四部分内容。

(1) 初始化部分，用来设置循环控制的一些初始条件，如设置计数器等。

(2) 循环体部分，这是反复执行的一段代码，可以是单一的一条语句，也可以是复合语句。

(3) 迭代控制部分，用来修改循环控制条件。常常在本次循环结束，下一次循环开始之前执行。例如，使计数器递增或递减等。

(4) 终止判断部分，一般是一个关系表达式或逻辑表达式，其值用来判断是否满足循环终止条件。每执行一次循环都要对该表达式求值并进行判断，以保证及时终止循环。

Java 中提供的循环控制语句分为不确定次数循环控制语句和确定次数循环控制语句，不确定次数循环控制语句又分为 while 语句和 do-while 语句，确定次数循环控制语句就只有 for 语句了。

while 语句格式如下：

```
while(布尔表达式) 语句;
```

do-while 语句的语法格式如下：

```
do 语句;
while(布尔表达式);
```

for 语句的语法格式为：

```
for(表达式 1;表达式 2;表达式 3)
    语句;
```

Java 语言中的转向控制语句有两条：break 语句和 continue 语句。

## 5. 数组

Java 语言中的数组就是一组具有固定顺序、相同类型的数据集合，可以用数组名和下标唯一确定数组中的每一个元素。根据组织数组的结构不同，又将其分为一维数组、二维数组……，以此类推。

一维数组是由同一数据类型数据构成的线性表，其声明的语法格式有两种。

```
数组元素类型 数组名 []; 或 数组元素类型 [] 数组名;
```

在 Java 语言中，多维数组可看作是数组的数组。例如，二维数组可看作是一个特殊的一维数组，其每一个元素又是一个一维数组。

二维数组的一般声明格式如下：

```
数组元素类型 数组名[[]]; 或数组元素类型[[]] 数组名;
```

## 6. 方法

Java 语言中的方法同其他高级语言中的函数、过程或子程序类似，为了便于将复杂的任务分解为易于解决的简单任务，为了使程序中的部分代码可以根据需要能够多次重复地使用，而将完成某个特定任务逻辑上相关的语句组合在一起并赋予一个特定的名字。与其他高级语言不同的是，Java 语言中的方法一定要在类中定义，一定要通过类或对象来调用。

Java 语言方法的定义由方法头声明和方法体定义两部分组成，其一般定义格式如下：

[修饰符] 返回值类型 方法名(参数列表)

{方法体:

  局部变量定义;

  语句序列;        }

## 1.1.2 疑难解析

### 1. 多重循环

一个循环体内又包含另一个完整的循环结构，称为循环的嵌套。这种循环之中又嵌套循环的现象也称为多重循环。设计多重循环的程序可以采用“渐扩法”，所谓“渐扩法”就是先设计内循环，然后逐层向外扩展，直至满足问题的要求。这里面一个关键问题是参数的转换，一般来说，内循环的某些常量将要转换成外循环的变量，下面以判断 20 以内的各数是否是素数为例加以说明。

要判断 20 以内的各数是否是素数，首先要编写判断一个数是否是素数的程序，这个程序相对简单，编写如下：

```
public class Prime1 {
    public static void main(String args[]){
        int n=10;        boolean bl=true;
        for(int j=2;j<=(int)n/2;j++){
            if(n%j==0) { bl=false;    break; } }
        if(bl)    System.out.println(n+"是素数! ");
        else    System.out.println(n+"不是素数! ");    } } }
```

运行这个程序，结果显示为：10 不是素数！

将  $n=10$  改为  $n=13$ ，运行这个程序，结果显示为：13 是素数！

接着进行程序的扩展，程序的扩展可以看成两步进行。

第一步将上面的所有语句都用大括号括起来作为一个循环体，前面加上：

```
for(int i=2;i<20;i++)
```

第二步将常量  $n=10$  改为变量  $n=i$ ，就完成了两重循环的设计，满足了本题的要求。复杂的问题可以仿此逐层扩展，实现三重或更多重的循环程序设计。

### 2. 方法设计

掌握方法的设计主要要明确三点：第一，方法设计的目的是为了简化程序设计，是为了使完成某个特定任务、逻辑上相关的语句序列一次编写多处调用；第二，定义方法的目的是为了调用，而且是多处调用，所以它需要有自己的标志和标识符；第三，方法要在类中定义、要在主方法中调用，当然也可以在其他方法中调用，但决不能在其他方法中定义，方法之间是平等的。

明确了这三点可以把第 1 章中的任何一个示例程序都改写为方法，不仅可以改写为无参的方法，而且还可以改写为有参的方法。例如第一个 Java 语言源程序，可以将那唯一一条可执行语句用大括号括起来，前面加上 `static`、`void`、名字（例如，`greeting` 这是标识符）和小括

号 (这是标志), 再移到主方法的前面来, 而在主方法中原来可执行语句所在的位置上写上 `greeting()`; 就可以了。

上面改写的是一个无参的方法, 下面给出的则是一个有参的示例:

```
public class MyGreeting2 {
    static void greeting(String str) {
        System.out.println(str); }
    public static void main(String args[]) {
        String abc="Good morning class!";
        greeting(abc); } }
```

简单的程序如此, 复杂的程序也是如此, 下面给出的是将教材中例 1.25 打印输出 10 行的杨晖三角形改写为有参方法的示例:

```
class Yanghui2 {
    static void yanghui(int n) {
        int i,j;          int yhlevel=n;          int yanghui[][];
        System.out.println("杨晖三角形: ");  yanghui=new int[yhlevel][];
        for(i=0;i<yanghui.length;i++)  yanghui[i]=new int[i+1];
        yanghui[0][0]=1;          for(i=1;i<yanghui.length;i++)
        {  yanghui[i][0]=1;          for(j=1;j<yanghui[i].length-1;j++)
            yanghui[i][j]=yanghui[i-1][j-1]+yanghui[i-1][j];
            yanghui[i][yanghui[i].length-1]=1;          }
        for(i=0;i<yanghui.length;i++)
        {  for(j=0;j<n-i;j++)          System.out.print(" ");
            for(j=0;j<yanghui[i].length;j++)  System.out.print(yanghui[i][j]+" ");
            System.out.println();          }          }
    public static void main(String args[]) {          yanghui(15);          } }
```

本例中在输出的各行中加上了空格, 使输出一个近似的等腰三角形。

## 1.2 复习题

### 1.2.1 问答思考题

- (1) 什么是对象? 什么是类? 请举例说明。
- (2) Java 语言源程序编写时都有哪些要求?
- (3) Java 语言程序上机操作有哪些步骤? 涉及哪些命令文件?
- (4) Java 语言标识符书写有哪些规则? 请指出在下面的标识符中, 哪些是不正确的, 并说明原因: `here`, `_there`, `this`, `that`, `it`, `2to1`, `_it`。
- (5) 从下列字符串中选出正确的 Java 关键字:

`abstract`、`bit`、`boolean`、`case`、`character`、`comment`、`double`、`else`、`end`、`endif`、`extend`、`false`、`final`、`finally`、`float`、`for`、`generic`、`goto`、`if`、`implements`、`import`、`inner`、`instanceof`、`interface`、

line、long、loop、native、new、null、old、oper、outer、package、print、private、rest、return、short、static、super、switch、synchronized、this、throw、throws、transient、var、void、volatile、where、write。

(6) Java 中共有哪些基本数据类型，它们分别用什么符号来表示，各自的取值范围是多大？

(7) Java 语言中有哪些运算符和表达式？请举例说明。

(8) Java 语言中有哪些流程控制类型？各有哪些语句？

(9) Java 语言中的数组是如何定义和引用的？请举例说明。

(10) Java 语言中的方法是如何定义和调用的？什么是方法的重载？请举例说明。

(11) 什么是方法的重载？请举例说明。

### 1.2.2 程序设计题

(1) 编写输出师生早上相互问好的程序。

(2) 对每种数据类型定义一个变量，给它赋一个值，并打印输出。

(3) 用单目运算符计算正整数 5 两种自减后再减 1 的结果。

(4) 计算下列算术表达式的值：

int a=20,b=3,c=3,d=3,e=3; double f=20,g=3;

- ① a+b;      ② a-b;      ③ a\*b;      ④ a/b;      ⑤ a%b;      ⑥ a+b++;  
⑦ a++c;      ⑧ a+d--;      ⑨ a+-e;      ⑩ f\*g;      ⑪ f/g;      ⑫ f%g;

分析⑦式没有空格的情况。

(5) 计算下列关系表达式的值：

int a=20,b=3;

- ① a=b;      ② a!=b;      ③ a>b;      ④ a<b;      ⑤ a>=b;      ⑥ a<=b;

(6) 计算下列布尔表达式的值：

int x=6,y=7,z=8;

- ① (1) x<y      ② !(y<z)      ③ x<6||x>6  
④ x>y&&x<y      ⑤ x>y||x<y      ⑥ (x<y+2)&&(x+8>y)  
⑦ y==z&&x<y      ⑧ y==z||x<y      ⑨ (!(x==y))&&(x!=y)&&(x<y||x>y)

(7) 计算下列位表达式的值：

int x=24,y=-24,z=3;

- ① x&z      ② x|z      ③ x^z      ④ ~x      ⑤ ~y      ⑥ x<<z  
⑦ y<<z      ⑧ x>>z      ⑨ y>>z      ⑩ x>>>z      ⑪ y>>>z

分析>>和>>>的不同。

(8) 用条件运算符取两数之中的最小值。

(9) 阶跃函数是当 x 大于 0 时为 1，否则为 0，任意给出一个 x 值，编程求出函数值。

(10) 有一函数：

$$y \begin{cases} x & (x < 1) \\ 2x - 1 & (1 \leq x < 10) \\ 3x - 11 & (x \geq 10) \end{cases}$$

写一程序，给定  $x$  值，输出  $y$  值。

(11) 编写一程序：将给定的 3 个数，按从大到小顺序输出。

(12) 用 `switch` 语句判断一个月份属于一年中的哪一个季节。

(13) 编写程序，求 1~10 的平方和。

(14) 已知  $\pi/4=1-1/3+1/5-1/7+\dots+1/2*n-1$ ，试编写程序求解  $\pi$ ，当  $\pi>3.14159$  且  $\pi<3.14160$  时停止计算。

(15) 编程计算《孙子算经》中的一题：今有物不知其数，三三数之剩二，五五数之剩三，七七数之剩二，问物几何？

(16) 计算 1~10 的阶乘，并逐个输出。

(17) 计算 2 的正整数次幂前 10 项之和。

(18) 编写程序找出所有的水仙花数并输出。水仙花数是三位数，它的各位数字的立方和等于这个三位数本身，例如  $371=3^3+7^3+1^3$ ，371 就是水仙花数（请分别用 `while` 循环和 `for` 循环实现）。

(19) 计算 5、6、7、8、9 的 10000 以内的各次幂值。

(20) 用 `break` 语句判断某数是否为素数。

(21) 用 `continue` 语句计算 100 以内 5 的整数倍数的和。

(22) 计算 100 以内的素数，7 个一行左对齐排列。

(23) 求前 20 个斐波那契数，两列输出，每列左对齐。

(24) 编写程序，打印输出一个浮点数组中的最大值和最小值。

(25) 编写程序，在给出的一维数组中查找一个数，找到时输出已找到某数，未找到时输出未找到某数。

(26) 编写程序，在给出的一维数组中统计某一区间数值的个数。

(27) 编写程序，将给出的一维数组用冒泡法按由大到小顺序排列。

(28) 编写程序，将给出的一维数组用选择法按由大到小顺序排列。

(29) 编写程序，将一个二维数组的每一行按升序排列。

(30) 编写一个通用方法，打印输出一个浮点数组中的最大值和最小值。

(31) 编写一个通用方法，在给出的一维数组中查找一个数，找到时输出已找到某数，未找到时输出未找到某数。

(32) 编写一个通用方法，在给出的一维数组中统计某一区间数值的个数。

(33) 编写一个通用方法，将给出的一维数组用冒泡法按由大到小顺序排列。

(34) 编写一个通用方法，将给出的一维数组用选择法按由大到小顺序排列。

(35) 编写一个通用方法，将一个二维数组的每一行按升序排列。

(36) 编写一个方法求出 1000 以内的完全数。完全数是其所有因子（包括 1 但不包括该数自身）的和等于该数自身的数，例如  $28=1+2+4+7+14$ ，28 就是一个完全数。

(37) 编写一方法验证简洁与或者简洁或。

(38) 自拟题目、自编程序验证方法的重载。

(39) 用一维字符串数组保存教师姓名，用二维浮点型数组保存教师工资的各项收入，编写多个方法分别对工资进行输出、排序、统计、计算、查找等操作。

(40) 用一维字符串数组保存学生姓名，用二维整型数组保存学生成绩，编写多个方法分别对成绩进行显示、排序、统计、计算、查找等操作。

## 1.3 复习题解答

### 1.3.1 问答思考题解答

(1) 对象指的是人类行动或思考时作为目标的人或事物，在人们正常的逻辑思维中对每一个客观存在的目标所考虑的不外是他具有的特征以及可能的行为，所以 Java 语言将问题中涉及到的客观事物抽象为特征和行为的逻辑统一体称之为对象。类是许多相同或相似的事物的综合。这里所说的“相同或相似”指的是不同事物具有相同或相似的特征和行为，这里的综合则是一种逻辑上的抽象。例如当老师进行课堂教学时，就把学生作为对象，考虑到他们现有的知识、接受的能力和可能遇到的困难。此外，我们只能看到一个具体的学生，名字可能为“赵一”、“王二”等，但我们看不到一个没有具体姓名、年龄等的学生。然而我们却可以从成千上万不同层次的学生中抽出他们共同的特征和行为加以综合，从而形成学生这个概念，这就是学生“类”。

(2) 在编写 Java 语言源程序的时候有如下要求。

① Java 语言是区分大小写的语言，关键字的大小写不能搞错，如果把 `class` 写成 `Class` 或者 `CLASS` 都会导致错误。

② Java 语言源程序每一语句都要用分号结束，一行可以书写多条语句，每条语句都要用分号结束。

③ 源程序编写好之后，要以文件的形式保存在硬盘或软盘上，称为源程序文件，或称为源文件。这里需要注意，这个源文件的名称不能随便取，它必须与程序的主类名相一致，所谓主类是指包含有 `main` 方法的类。一个 Java 应用程序可能有多个类，但是由于在一个 Java 应用程序中必需定义且只能定义一个 `main` 方法，所以一个 Java 应用程序中的主类也就唯一地确定了。这个源文件还要以 `java` 为其扩展名。所以例 1.1 中的源文件应命名为 `MyGreeting.java`。

④ 源程序中“`//`”后面的内容是注释，注释用来对程序中的代码做出解释。注释的内容在程序编译时不产生目标码，因此，注释部分的有无对程序的执行不产生任何影响。但不要认为注释毫无用处，在一个复杂的程序中，添上注释可增加程序的可读性，也有利于程序的修改、调试和交流。

⑤ 源程序中要恰当地使用分隔符，空格、逗号、分号及行结束符都称为分隔符，Java 语言规定任意两个相邻标识符、数字、保留字或两个语句之间必须至少有一个分隔符以便编译程序能识别。

(3) Java 语言程序上机操作除了编辑源文件之外还有两个步骤：第一个步骤是编译源文件，将源文件编译成目标码文件，扩展名为 `class`。第二个步骤是运行目标码文件，执行主方法中语句所规定的操作，完成程序预定的任务。这两个步骤要用到 `jdk` 中的文件夹 `bin` 内的两个命令文件，一个是 `javac.exe`，一个是 `java.exe`。

(4) 标识符的书写规则是程序设计中必须遵守的重要规则，在 Java 语言中书写标识符的规则如下。

① 标识符必须由字母、下划线 (`-`)、美元符 (`$`) 开始。

② 其余字符可以是字母、下划线、美元符和数字。

③ 标识符是大小写敏感的。



#### ④ 标识符没有最大长度限制。

Java 语言不采用通常计算机系统采用的 ASCII 代码集,而是采用 Unicode 国际标准字符集。所以,Java 语言中可以对标识符命名的字母包括:英文字母 A~Z、a~z 以及 Unicode 字符集中序号大于 0xC0 的所有符号。

在 Java 语言中,有一部分标识符是被赋予了特殊的意义,并留作专门的用途,不能当作一般的标识符使用,这些标识符称为保留字或关键字。

在所给的标识符中,不正确的有: `this`, `2tol`。其中 `this` 是 Java 中的关键字,不能用作标识符; `2tol` 是以数字开头的字符串,也不符合 Java 对标识符的规定。

(5) 正确的 Java 关键字包括: `abstract`、`boolean`、`case`、`double`、`else`、`false`、`final`、`finally`、`float`、`for`、`generic`、`goto`、`if`、`implements`、`import`、`inner`、`instanceof`、`interface`、`line`、`long`、`native`、`new`、`null`、`outer`、`package`、`private`、`rest`、`return`、`short`、`static`、`super`、`switch`、`synchronized`、`this`、`throw`、`throws`、`transient`、`var`、`void`、`volatile`。

不是 Java 关键字的有: `bit`、`character`、`comment`、`end`、`endif`、`extend`、`line`、`loop`、`old`、`oper`、`print`、`where`、`write`。

(6) Java 的数据类型共分为两大类,一类是基本类型,一类是复合数据类型。基本类型共有 8 种,分为 4 小类,分别是逻辑类型、字符类型、整数类型和浮点类型。

逻辑类型或称布尔类型,使用 `boolean` 表示,它有两个常量值: `true` 和 `false`。

字符类型使用 `char` 表示,一个 `char` 表示一个 Unicode 字符。每一个 Unicode 字符可用“\uxxxx”表示,其中 xxxx 是任意的一个 16 位无符号整数,范围为 0~65 535。`char` 类型的常量值必须用一对单引号(‘’)括起来。例如,‘B’的 Unicode 值为‘\u0042’。

整数类型共有 4 种,分别是: `byte` (1 字节)、`short` (2 字节)、`int` (4 字节)及 `long` (8 字节)。`byte` 表示的数的范围为 -128~127, `short` 表示的数的范围为 -32 768~32 767, `int` 表示的数的范围为 -2 147 483 648~2 147 483 647, `long` 表示的数的范围为 -9 223 372 036 854 775 808~9 223 372 036 854 775 807, 整型常量可用十进制、八进制或十六进制形式表示。以 1~9 开头的数为十进制数,以 0 开头的数为八进制数,以 0x 开头的数为十六进制数。Java 中 4 种整型量都是有符号的。

浮点类型数据共有两种类型,分别是:单精度浮点数 `float`(4 字节)及双精度浮点数 `double` (8 字节)。`float` 表示的数的范围约为  $1.4e-45 \sim 3.4e+38$ , `double` 表示的数的范围约为  $4.9e-324 \sim 1.8e+308$ 。一般为了区分两种浮点数常量,通常在常量后加一个小写字符,加 `f` 表示单精度型浮点数,加 `d` 表示双精度型浮点数。

(7) Java 语言中有赋值运算符和赋值表达式,如: `a=5`;

算术运算符,按其要求的操作数的个数分为单目运算符和双目运算符两类。单目运算符只有一个操作数如 `+` 使操作数取正值、`-` 使操作数取负值、`++` 使操作数加 1、`--` 使操作数减 1。双目运算符要求两个操作数如 `+` 求两数和、`-` 求两数差、`*` 求两数积、`/` 求两数商、`%` 求两数相除的余数。算术表达式如: `int c=a+b`;

关系运算符有 `==` 等于、`!=` 不等于、`<` 小于、`<=` 小于等于、`>` 大于、`>=` 大于等于 6 种,在一个关系运算符两边的数据类型应一致。由两个操作数和关系运算符构成一个关系表达式,一个关系表达式的结果类型为布尔型,即关系式成立为 `true`,不成立为 `false`,例如: `7>5`; 结果为 `true`。

逻辑运算符有 6 个,它们是: `!` (非)、`&` (与)、`|` (或)、`^` (异或)、`&&` (简洁与)和