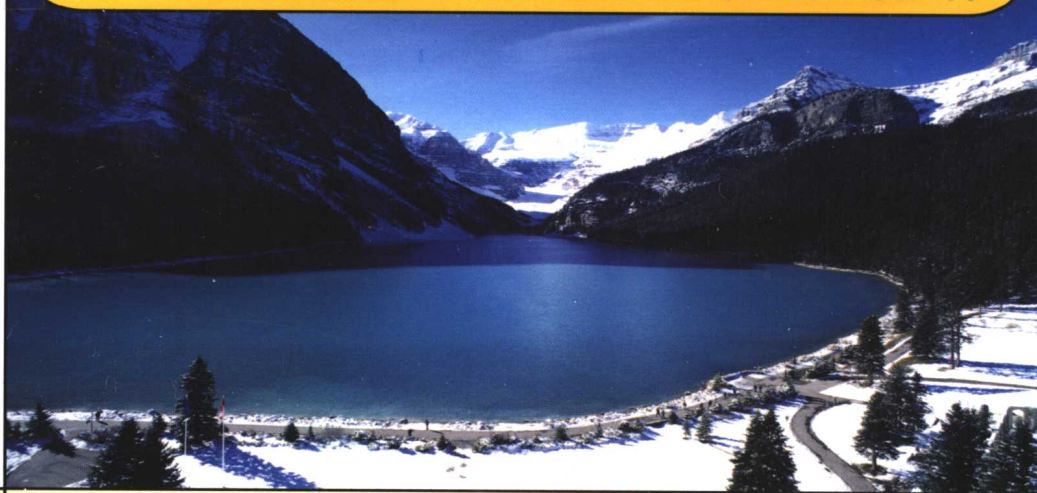


高等院校信息管理与信息系统专业系列教材



Java程序开发 例题与习题

张基温 陶利民 编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

高等院校信息管理与信息系统专业系列教材

Java 程序开发 例题与习题

张基温 陶利民 编著

清华大学出版社
北 京

内 容 简 介

本书是一本通过编程实践引导学生提高 Java 程序开发能力的教材。全书共分 7 章:第 1 章主要培养学生解题时首先考虑如何定义类的习惯;第 2 章、第 3 章分别训练如何设计方法函数和数据结构;第 4 章培养学生通过类的派生、多态性以及类之间的消息传递解决复杂问题的能力;第 5 章培养学生如何利用系统资源;第 6 章培养学生如何设计“用户友好”的程序界面;第 7 章是关于 Applet 程序设计的基本训练。

本书既有编程知识介绍,也有精选的例题,还有大量自测习题及其参考答案。可以作为教授 Java 程序设计课程的习题课教材,也可供读者自学。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Java 程序开发例题与习题/张基温,陶利民编著. 北京:清华大学出版社,2003
(高等院校信息管理与信息系统专业系列教材)

ISBN 7-302-06487-3

I. J… II. ①张… ②陶… III. JAVA 语言—程序设计—高等学校—习题
IV. TP312-44

中国版本图书馆 CIP 数据核字(2003)第 022944 号

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

[http:// www.tup.tsinghua.edu.cn](http://www.tup.tsinghua.edu.cn)

责任编辑:范素珍 付宇光

印 刷 者:清华大学印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:18.75 字数:426 千字

版 次:2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

书 号:ISBN 7-302-06487-3/TP·4874

印 数:0001~5000

定 价:24.00 元

出版说明

20 世纪三四十年代,长期摸索前进的古老的计算技术与刚走向成熟的电子技术相结合。这一结合,不仅孕育了新一代计算工具——电子计算机,还产生了当时谁也没有料到的巨大效应。电子计算机——这种当初为计算而开发出来的工具,很快就超出计算的范畴,成为“信息处理机”的代名词;人类开始能够高效率地开发并利用信息;信息对人类社会的作用得以有效地发挥,并逐步超过材料和能源成为人类社会的重要支柱;信息产业急剧增长,信息经济高速发展,社会生产力达到了新的高度;人们的信息化意识不断加强,人类在信息资源方面开始更加激烈的竞争,社会发展走上信息化轨道。

科学技术是第一生产力,教育是基础。为了加速社会信息化的过程,以培养信息资源开发人才为目标的信息管理与信息系统专业应运而生。

对于与信息有关的学科从纵向来看,信息管理与信息系统处于信息学、信息技术、信息管理、信息经济、信息社会学这个层次的中间。它以信息学和信息技术为基础,并与信息经济和信息社会学相关联。从横向来看,它处在管理学、信息科学与技术、系统科学等有关学科领域的交叉点上。它对技术有极高的要求,又要求对组织的深刻理解和对行为的合理组织,反映了科学与人本融合的特点。这种交叉和融合正是信息管理与信息系统的最重要特征,是别的学科或专业难以取代和涵盖的。

我国的信息管理与信息系统专业创建于 20 世纪 70 年代末。在不到 20 年的时间里,已发展到 150 多个点,成为培养信息化人才的主要摇篮。其发展速度之快,影响之深远,已令世人和学术界刮目相看。

然而,作为一个新的学科,这个专业的课程体系、教学内容以及教学方法都需要经历一个逐步完善、逐步成熟的过程。特别是教材的建设更需要经过长期的实践和探索。没有这样一个过程,具有专业特点,符合中国实际的教材是不可能产生的。近 20 年来,大家一直在课程体系的完善和建设并具有自己专业特点的教材方面不断进行探讨。1991 年全国 10 所财经类院校的经济信息管理专业负责人汇聚在太原,召开第一次教学研讨会。以后,1993 年在大连,1995 年在武汉,1997 年在烟台,又有更多的院校参加了这一研讨。在讨论中,各校同仁一致认为,教材建设是当务之急,它不仅直接体现和落实培养目标,同时也是学科建设的根本所在。目前一些课程缺乏专业特点,简单搬用其他专业教材的状况亟待改变。在武汉会议上,这一共识得到了与会的国家教委有关部门负责同志的赞许,清华大学出版社也对此给予了热情的支持。会议确定了首批计划编写八九本教材,由张基温教授主编,由清华大学出版社出版,并聘请了魏晴宇、陈禹两位教授作为顾问。

经过两年多的工作,在全国许多高等院校同仁的共同努力下,其中 7 本已完成初稿。我们希望这批教材的问世,能够起到抛砖引玉的作用,对各校信息管理与信息系统专业的建设和发展有所裨益。

近 20 年来的实践使我们对信息管理与信息系统专业的重要性和困难有了切身的体会。一方面,席卷全球的信息化大潮把信息管理推到了时代发展的前沿,信息、信息管理、信息系统已经成为全社会关注的热点,这为信息管理与信息系统专业的建设创造了良好的外部条件,提供了难得的机遇。另一方面,信息技术的迅速发展与普及,多种社会经济因素的互相渗透和影响,前所未有的许多新问题、新情况的出现,又给这个专业的发展带来了很大的困难。我们深感责任之重大和任务之艰巨。在这套教材问世之时,我们再次表示这样一个心愿:希望与全国的同行共勉,为祖国信息化建设的宏伟事业多添一块砖,多加一块瓦,多出一份力,培养出更多的优秀人才。

由于上述种种原因,这套教材当然不会是完整的,也不会是完美的。它必然要不断补充、不断修改、不断完善。因此,对于它的任何修改意见,都是我们非常盼望的。希望能够在这套教材出版后,收到更多的意见和建议,使之逐步走向成熟。

全国高等院校计算机基础教育研究会
财经信息管理专业委员会
信息管理与信息系统专业教材编委会
1997 年 9 月

前 言

计算机应用总是与程序联系在一起的。但是,随着信息时代的到来,程序设计不仅是培养学生计算机应用开发能力的技术课程,更重要的是它应当作为训练信息时代基本素养——计算思维能力的重要公共基础课程。

程序设计是一门实践性极强的课程。要学好程序设计课程,提高程序设计的能力,主要要进行两方面的训练:一是熟练地掌握一种计算机语言,正确地运用语言机制,充分地利用语言提供的环境和资源;二是掌握基本的程序设计方法,不断积累程序设计的经验,使自己的程序设计能力不断提高。简单地说,就是在正确的程序设计方法的指导下多做练习。本书就是为有这种需求的读者群编写的一本教材,既可以作为教授 Java 程序开发课程的习题课的教材,也可以作为读者自学的教材。

为了便于读者自学,本书每一章的内容都分为 4 部分:

- x. 1 编程知识
- x. 2 例题示范
- x. 3. 1 自测习题
- x. 3. 2 参考答案

本书编写的线索与我主编的《Java 程序开发教程》(清华大学出版社 2002 年 10 月第 1 版)相同:

第 1 章主要培养学生解题时首先考虑如何定义类的习惯;

第 2 章、第 3 章分别训练如何设计方法函数和数据结构;

第 4 章培养学生如何通过类的派生、多态性以及类之间的消息传递来解决复杂问题;

第 5 章培养学生如何利用系统资源;

第 6 章培养学生如何设计“用户友好”的程序界面;

第 7 章是关于 Applet 程序设计的基本训练。

本书中的例题和习题参考答案都是我的研究生陶利民调试的。朱嘉钢副教授参加了部分工作。

张基温

2002 年 11 月

目 录

第 1 章 Java 入门	1
1.1 编程知识	1
1.1.1 Java 平台	1
1.1.2 Java 程序及其结构	2
1.1.3 Java 方法的结构	4
1.1.4 Java 运算符初步	4
1.1.5 注释	4
1.2 例题示范	5
1.3 自我测试	8
1.3.1 自测习题	8
1.3.2 参考答案	9
第 2 章 方法设计	19
2.1 编程知识	19
2.1.1 运算符与表达式	19
2.1.2 语句流控制	21
2.1.3 算法介绍	26
2.2 例题示范	27
2.3 自我测试	37
2.3.1 自测习题	37
2.3.2 参考答案	45
第 3 章 数据成员设计	85
3.1 编程知识	85
3.1.1 Java 数据类型	85
3.1.2 Java 程序中的数据形式	85
3.1.3 Java 标识符与关键字	87
3.1.4 变量的作用域	87
3.1.5 数组	88
3.2 例题示范	91
3.3 自我测试	98
3.3.1 自测习题	98

3.3.2 参考答案	102
第4章 Java 程序结构	113
4.1 编程知识	113
4.1.1 类的继承	113
4.1.2 对象的使用与对象间的通信	114
4.1.3 多态性	115
4.1.4 接口	116
4.1.5 包	117
4.1.6 访问控制与修饰符	118
4.2 例题示范	119
4.3 自我测试	128
4.3.1 自测习题	128
4.3.2 参考答案	142
第5章 Java 资源及其利用	165
5.1 编程知识	165
5.1.1 java.lang 包	166
5.1.2 java.io 包	177
5.1.3 java.net 包和 java.util 包	186
5.2 例题示范	190
5.3 自我测试	204
5.3.1 自测习题	204
5.3.2 参考答案	210
第6章 图形用户界面 GUI	223
6.1 编程知识	223
6.1.1 建立图形用户界面	223
6.1.2 几个重要的图形组件	226
6.1.3 GUI 组件包与 JComponent	229
6.2 例题示范	234
6.3 自我测试	241
6.3.1 自测习题	241
6.3.2 参考答案	242
第7章 Applet 程序基础	261
7.1 编程知识	261
7.1.1 Applet 类	261

7.1.2	向 Applet 传递参数	263
7.1.3	在 Applet 中使用图形、字体和颜色	263
7.1.4	在 Applet 中使用图像和声音	267
7.1.5	JAR 文件——Java 存档文件	269
7.2	例题示范	270
7.3	自我测试	273
7.3.1	自测习题	273
7.3.2	参考答案	274

Java 入门

1.1 编程知识

1.1.1 Java 平台

1. Java 程序的半编译、半解释特点

一种计算机语言就是一种程序设计工具,并用于将程序员编写的程序源代码翻译成机器可以执行的目标代码。尽管高级语言已经不像低级语言那样完全依赖于计算机的硬件系统,但是硬件和操作系统平台的差异还是依稀可见,使得在一种平台上开发的程序运行于另一个平台上时,多少还要进行一些修改。随着计算机网络的广泛应用,各种异构系统都溶入了 Internet 的海洋。在这种情形下,平台之间的不兼容性成为计算机应用的一个障碍。就像咖啡可以被不同种族的人饮用一样,Java 的目标是成为 Internet 上的“世界语”,用它编写的程序具有平台无关性。

如图 1.1 所示表明 Java 程序从编写到运行的一般过程。Java 实现其平台无关性的

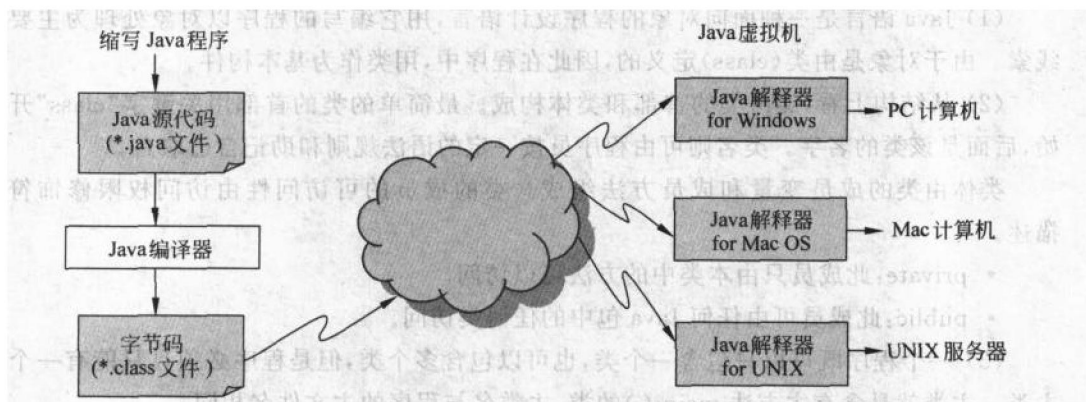


图 1.1 Java 程序的半编译和半解释

奥秘在于它的语言平台由分布在不同地点的两部分组成。

- 编译器:对源代码进行半编译,生成与平台无关的字节码文件。
 - 解释器:分布在网络中不同的操作系统平台上,用于对字节码文件半解释执行。
- 这些分布在不同平台上的解释器,也称为 Java 虚拟机。

所以,通常把 Java 程序的翻译过程称为半编译、半解释。而其他语言程序则要么是全编译,要么是全解释。正因为如此,Java 程序才具有“一次编写,到处运行”的特点。

2. Java 程序开发环境——SDK

SDK 集成有如下一些功能:

- (1) Java 编译器
- (2) Java 解释器
 - 内附 Appletviewer——Applet Bytecode 解释器
 - 内附 Java——Application 解释器
- (3) 简单的除错工具

目前 SDK 所支持的操作平台包括 Solaris、FreeBSD、Linux、Windows 等。

SDK 可以从站点 <http://java.sun.com/products/jdk/> 下载得到。

1.1.2 Java 程序及其结构

Java 是一种广泛使用的网络编程语言,用它编写的程序都在 Java 虚拟机中执行。但是激活机制有两种方式,据此也将使用 Java 语言编写的程序分为相应的两大类:Java 独立应用程序(Java Application)和 Java 小应用程序(Java Applet)。Java Application 在本机上由 Java 解释程序来激活 Java 虚拟机,而 Java Applet 则通过浏览器来激活 Java 虚拟机。此外,它们的程序结构也不相同。

1. Java 应用程序(Java Application)

(1) Java 语言是一种面向对象的程序设计语言,用它编写的程序以对象处理为主要线索。由于对象是由类(class)定义的,因此在程序中,用类作为基本构件。

(2) 从结构上看,类由类的首部和类体构成。最简单的类的首部用关键字“class”开始,后面是该类的名字。类名则可由程序员按一定的语法规则和助记自己取名。

类体由类的成员变量和成员方法组成。类的成员的可访问性由访问权限修饰符描述。

- private:此成员只由本类中的方法可以访问。
- public:此成员可由任何 Java 包中的任何类访问。

(3) 一个程序既可以只包含一个类,也可以包含多个类,但是程序必须且只能有一个主类。主类就是含有主方法 main()的类,主类名与程序的主文件名相同。

(4) main()方法是 Java Application 程序的入口,有固定的书写格式:

• 2 •

```

public static void main (String[] args)
{
    //...
}

```

main()方法对一个程序起总控作用。Java 解释器总是从 main()方法开始解释和执行应用程序,因此一个可执行的 Application 必须而且只能有一个 main()方法。

2. Java Applet(Java 小应用程序)

Java Applet 也是由若干个类组成的,每个类也用关键字 class 声明。但是,Java Applet 不像 Java Application 那样作为独立程序能够被直接执行,它必须由支持 Java 的 WWW 浏览器或 AppletViewer 加载执行。在执行 Java Applet 之前,必须先准备一个供其栖身的 HTML 文件,使用特殊的 HTML 标记,把 Applet 嵌入其中。一般将 Applet 嵌入到 HTML 文件中使用的 HTML 标记形式为:

```

<applet code="..." width=... height=...>
</applet>

```

其中,<applet>和</applet>说明在此插入一个 Applet,字节码文件的名字放在“code =”后(扩展名.class 必不可少!),而 width 和 height 参数说明了此 Applet 占用的宽度和高度。

从某种意义上讲,可以将 Applet 视为在 WWW 浏览器中执行的程序。如图 1.2 所示,Applet 运行的完整过程是:首先由 Java 编译器将其源代码编译生成通用的字节码文件,然后还得另外编写一个 HTML 文件将该字节码嵌入其中;再将此字节码文件和 HTML 文件保存在服务器的特定路径下。当 WWW 浏览器下载此 HTML 文件并显示时,它会自动下载其中指定的 Applet 字节码,然后调用内置在浏览器中的 Java 解释器来解释执行下载到本机的字节码程序。

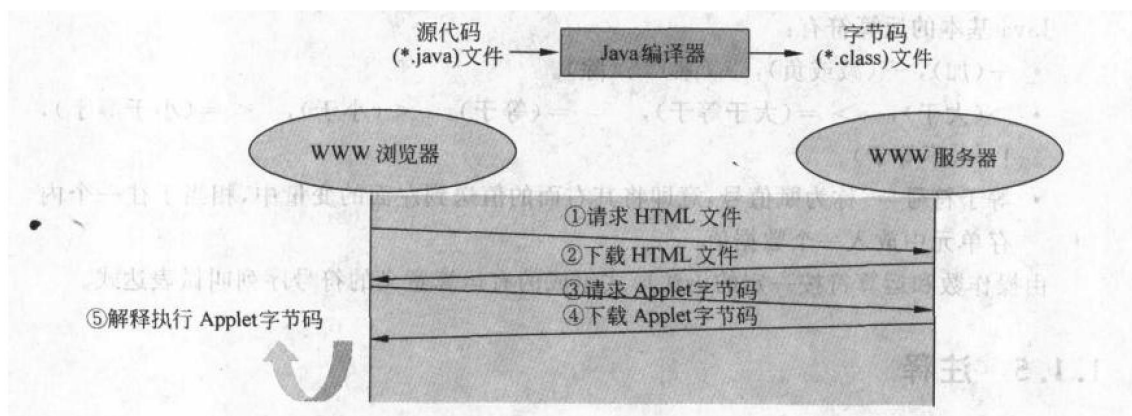


图 1.2 Applet 的执行流程示意图

1.1.3 Java 方法的结构

一个完整的方法定义如图 1.3 所示的结构。它由两部分组成：方法首部和方法体。

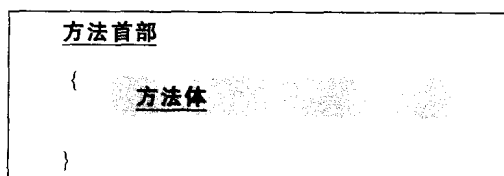


图 1.3 方法的结构

方法首部要描述下列内容：

- 方法名。
- 方法的返回值类型。
- 参数及其类型。
- 其他修饰符。

方法体是括在一对花括号中的一些语句序列，它们描述了方法要执行的操作。方法体的执行是在某种事件的发生时被激发的，方法被激发的过程称为方法的调用。方法被调用执行后，往往要向调用者返回一个数据。在方法中，要使用语句 `return` 返回结果。`return` 语句返回的数据的类型称为方法的类型。有些方法也可以不返回任何值。无返回值的方法，其类型为“`void`”型。

方法执行过程中，要对一些数据进行加工。被加工的数据中，有一些是由调用者传递来的（称为方法的参数），有一些则是方法自己的。方法也可以没有参数。

1.1.4 Java 运算符初步

Java 基本的运算符有：

- `+`（加），`-`（减或负），`*`（乘），`/`（除）。
- `>`（大于），`>=`（大于等于），`=`（等于），`<`（小于），`<=`（小于等于），`!=`（不等于）。
- 等于符号 `=` 称为赋值号，意即将其右面的值送到左面的变量中，相当于往一个内存单元中放入一个数据值。

由操作数和运算符按一定的语法形式组成的有运算意义的符号序列叫做表达式。

1.1.5 注释

注释是对程序、程序的局部或一个语句的说明，用于增加程序的可读性。注释不影响程序的逻辑功能。为了不使注释与程序的代码相混淆，需要按一定的格式用注释符将注

释部分与程序的代码部分区分开。Java 允许使用以下 3 种注释格式。

(1) 在“//”后面写注释,只能写在一行中,并且后面不能再写任何程序代码,称单行注释。

(2) 在“/*”和“*/”之间写的注释,可以占用几行,并且后面可以再写程序代码,如

```
/* This is a Application.  
The name of this program  
is SimpleProgram. */
```

(3) 在“/**”和“*/”之间写注释是 Java 语言特有的注释方式。这种注释主要是为支持 JDK 工具 Javadoc 而设置的,Javadoc 工具生成软件文档时会用到其中的注释内容,如

```
/** This is a Application WelcomeApp.  
Program's name is WelcomeApp.java */
```

1.2 例题示范

例 1.1 设计一个电子日历类 Date 的界面。

解:设计一个类时,一般先分析对象的应用环境,分析对象应具有的静态属性和动态属性,由此抽象出类数据成员和方法成员,从而得到类的界面。

1. 电子日历类 Date 的静态属性

- Year(表示年)
- Month(表示月)
- Day(表示日)

2. 电子日历类 Date 的动态属性

使用一个电子日历时,进行的操作与对应的方法有以下几种。

- (1) 创建一个电子日历对象——构造方法,它与类同名。
- (2) 设定电子日历的日期——方法 getDate()。
- (3) 电子日历给出日期——方法 setDate()。
- (4) 让电子日历进行相关计算:
 - 计算与当前日期相距多少天——方法 getDays(Date date1)。
 - 求出与当前日期相距多少天的日期——方法 getOtherDate(int d)。

3. 电子日历类 Date 中各成员的访问权限的确定

正像对电子日历不能让用户拆开外壳直接去拨动显示装置中的元件,而只能让用户使用电子日历上的有关按键进行操作一样,数据成员(Year、Month 和 Day)是不允许外部操作的,因此访问权限应为 private;而方法好像是电子日历上的有关按键,专门供用户

使用的,因此访问权限应为 public。Year、Month 和 Day 只能由方法成员进行操作。

4. 方法的类型的确定

单个的 Year、Month 和 Day 都是些整数,所以它们都是 int 类型。

在将要设计的类 Date 中,方法给出的结果有如下 4 种情形。

(1) 给出天数,应为 int 类型,如计算与当前日期相距多少天的方法 getDays(Date date1)。

(2) 给出日期,应为 Data 类型,如:

- 计算与当前日期相距多少天的日期的方法 getOtherDate(int d)。
- 给出日期的方法 getDate()。

(3) 只进行一些操作,没有输出的方法,应为 void 类型,如设定电子日历日期的方法 getDate()。

(4) 不能有任何修饰的方法——构造方法。

5. 方法参数的确定

一个方法需要外界(调用者)向它传递的数据,称为参数。对于类 Date 来说,它的方法按照参数可以分为以下 4 种情形。

(1) 只要传递一个整数——天数的方法:求出与当前日期相距多少天的日期方法 getOtherDate(int d)。

(2) 只要传递一个日期的方法:计算与当前日期相距多少天的方法 getDays(Date date1)。

(3) 不需要传递数据,好像只要按键就可以的方法:

- 设定电子日历日期的方法 setDate()。
- 给出日期的方法 getDate()。

(4) 要具体给出 Year、Month 和 Day 的值的方法——构造方法,好像制造好一个电子日历时要先给它一个具体的日期值一样。

根据上述分析,可以得到 Date 类的界面如下:

```
class Date                                // 定义 Date 类
{
    //以下是 Date 类的数据成员
    private int Year;                      // 年
    private int Month;                    // 月
    private int Day;                      // 日

    // 以下是 Date 类的方法成员
    Date(int year,int month,int day);     // 带参数的构造方法
    public void setDate();                 // 设定日期的方法
    public Date getDate();                 // 给出日期的方法
    public int getDays(Date date1);       // 计算与当前日期相距多少天
    public Date getOtherDate(int d);       // 计算与当前日期相距多少天的日期
}
```

当然,上面定义的仅仅是类 Date 的界面,还不是一个完整的类。因为每一个方法如何实现还没有给出进一步的定义。

例 1.2 设计并测试一个表示图书的 Book 类,它包含图书的书名、作者、月销售量等属性,另有两个构造方法(一个不带参数,另一个带参数)和两个成员方法 setBook()(用于设置书名、作者、月销售量等属性)及 printBook()(用于输出书名、作者、月销售量等数据)。

解: Book 类包含的属性有 title(书名)、author(作者)及 numsold(月销售量),另有两个构造方法 Book()和 Book(String str1,String str2,int num)、setBook()及 printBook()。本题参考程序如下:

```
class Book
{
    String title;           // 书名
    String author;          // 作者
    int numsold;            // 月销售量

    Book(){ }              // 不带参数的构造方法

    Book(String str1,String str2,int num)    // 带参数的构造方法
    {
        title=str1;        // 将 title 的初值置为 str1
        author=str2;       // 将 author 的初值置为 str2
        numsold=num;       // 将 numsold 的初值置为 num
    }

    void setBook(String str1,String str2,int num) // 设置属性值的方法
    {
        title=str1;        //将 title 的初值置为 str1
        author=str2;       //将 author 的初值置为 str2
        numsold=num;       //将 numsold 的初值置为 num
    }

    void printBook()
    {
        System.out.println("书名:"+title);    // 输出书名
        System.out.println("作者:"+author);    // 输出作者
        System.out.println("月销售量:"+numsold); // 输出月销售量
    }
}

// 下面是测试上述类的主类
public class TestBook{
    public static void main(String[] args){
        Book book1=new Book();                // 创建 Book 类的对象 book1
        book1.setBook("C# 实用教程","吴军",700); // 调用方法 setBook()设置图书信息
    }
}
```



```

        book1.printBook(); // 输出图书的信息
        Book book2=new Book("C# 语言基础教程","张威",600); // 创建 Book 类的对象 book2
        book2.printBook(); // 输出图书的信息
    }
}

```

本程序运行结果如下：

```

书名：C# 实用教程
作者：吴军
月销售量：700
书名：C# 语言基础教程
作者：张威
月销售量：600

```

例 1.3 编写并运行一个 Java 程序，由命令行向应用程序传递一个字符串“Hello, world!”,并在显示器上输出该字符串。

解：命令行参数保存在 main()方法的参数字符串数组 args 中。本题程序如下：

```

// 文件名：PrintString.java
public class PrintString{
    public static void main(String[] args){
        if(args.length<1)
        {
            System.out.println("请由命令行传入一个字符串。");
            System.exit(-1); // 系统方法 exit(),用于退出应用程序
        }
        System.out.println(args[0]);
        // 字符串"Hello,world!"存放在 main()的参数 args 的第 0 个元素中
    }
}

```

本程序运行结果如下：

```

java printstring Hello, world! <Enter>
Hello, world!

```

1.3 自我测试

1.3.1 自测习题

题 1.1 考察一些熟悉的对象，如学生、课程、教师、教室等。假如某个教学管理软件要用到这些对象，请设想一下这些对象的使用环境，据此抽取这些对象的属性和行为，并设计出相应的类的界面。

题 1.2 设计并测试一个求长方体的体积和表面积的 Java 类。

题 1.3 设计并测试一个表示用户的 User 类，它能获取用户的用户名和口令。