



21世纪高等学校电子信息类专业规划教材

Java语言 学习指导与习题解答

于万波等 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

21 世纪高等学校电子信息类专业规划教材

Java 语言学习指导 与习题解答

于万波等 编著

清华大学出版社
北京交通大学出版社

· 北京 ·

内 容 简 介

本书是《Java 语言实用教程》的配套辅导书。

本书概括了 Java 语言的一些基本内容与要点,对一些典型问题进行了分析。收集了《Java 语言实用教程》中的习题,并分章做了讲解。在每章后面给出了一些测试题,这些测试题多数是为了提高程序设计能力而给出的。

第 8 章给出了近几年全国计算机等级考试二级 Java 语言考试笔试真题及答案,第 9 章给出了一套二级 Java 语言模拟试题及答案。

本书既可以作为教材辅助指导,也可以独立使用。亦可作为计算机等级考试(二级 Java 语言)的辅导教材适合于 Java 语言的初学者,也适合于巩固提高 Java 语言学习成果的读者。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

Java 语言学习指导与习题解答/于万波等编著. —北京:清华大学出版社;北京交通大学出版社,2008.12

(21 世纪高等学校电子信息类专业规划教材)

ISBN 978-7-81123-481-7

I. J… II. 于… 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 188603 号

责任编辑:郭东青

出版发行:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京交大印刷厂

经 销:全国新华书店

开 本:185×260 印张:14.75 字数:368 千字

版 次:2009 年 1 月第 1 版、2009 年 1 月第 1 次印刷

书 号:ISBN 978-7-81123-481-7/TP·453

印 数:1~4 000 册 定价:23.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

前 言

本书是清华大学出版社与北京交通大学出版社 2008 年 10 月出版的《Java 语言实用教程》的学习辅导书。

事实上，这本书也是一本习题集或者说是习题解答。第 1 章至第 7 章收集了许多程序设计习题，适合 Java 语言的初学者，也适合想提高自己 Java 语言编程能力的读者。这些习题是经过精心设计的，每个习题都会让读者掌握一个知识点或者学会一门编程技术。

第 8 章与第 9 章收集了二级 Java 语言考试试题，这些试题训练读者掌握 Java 语言的一些基本知识，这些内容是任何一个学习 Java 语言的读者都应该掌握的。

做习题是一个重要的学习过程，不过太难的习题不利于初学者。习题应该是读者经过努力能够解决的问题，应该尽量使用模拟修改的方式学习编程。本书的习题就是根据这一思想设计的。

黄艳峰编写了前 3 章，其他章节由于万波编写，迟春见、王凯搜集整理了全国计算机等级考试（二级 Java 语言）的考试试题，在此一并表示感谢。

编 者

2008 年 12 月 1 日

目 录

第 1 章 Java 程序介绍	1
1.1 内容要点与典型题分析	2
1.1.1 Java 应用程序	2
1.1.2 类的基本知识	2
1.1.3 程序设计	4
1.1.4 文件操作	6
1.1.5 高层网络编程	8
1.2 教材课后习题提示与解答	8
1.3 测试题与参考答案	21
第 2 章 图形用户界面设计	31
2.1 内容要点与典型题分析	32
2.1.1 常用的 GUI 组件	32
2.1.2 事件	36
2.1.3 容器	37
2.1.4 菜单	40
2.1.5 swing 组件	42
2.2 教材课后习题提示与解答	42
2.3 测试题与参考答案	59
第 3 章 图形图像编程与动画制作	72
3.1 内容要点与典型题分析	73
3.1.1 图形绘制	73
3.1.2 图像操作	77
3.1.3 动画制作	77
3.2 教材课后习题提示与解答	78
3.3 测试题与参考答案	87

第4章 程序设计规范	93
4.1 内容要点与典型题分析	94
4.1.1 数据类型	94
4.1.2 分支与循环结构	96
4.1.3 方法的重载	96
4.1.4 接口	97
4.1.5 线程	98
4.1.6 异常	100
4.2 教材课后习题提示与解答	101
4.3 测试题与参考答案	117
第5章 类的构造	122
5.1 内容要点与典型题分析	122
5.1.1 类的组织	122
5.1.2 AWT 包	124
5.1.3 数学类	124
5.1.4 Applet 类	124
5.1.5 输入输出类	125
5.2 教材课后习题提示与解答	125
5.3 测试题与参考答案	129
第6章 输入输出与网络程序设计	133
6.1 内容要点	134
6.1.1 Socket 聊天程序设计	134
6.1.2 数据报发送与接收	134
6.1.3 RMI 程序设计	134
6.1.4 输入输出	134
6.2 教材课后习题提示与解答	134
6.3 测试题与参考答案	140
第7章 数据库管理程序	148
7.1 内容要点	148
7.1.1 数据库表	148
7.1.2 简单的数据库表操作	148
7.2 教材课后习题提示与解答	149
7.3 测试题与参考答案	154
第8章 二级 Java 语言笔试试题选编	158
8.1 2005 年 4 月二级 Java 语言笔试试题及解答	158

8.2	2005 年 9 月二级 Java 语言笔试题题及解答	163
8.3	2006 年 4 月二级 Java 语言笔试题题及解答	167
8.4	2006 年 9 月二级 Java 语言笔试题题及解答	172
8.5	2007 年 4 月二级 Java 语言笔试题题及解答	179
8.6	2007 年 9 月二级 Java 语言笔试题题及解答	185
8.7	2008 年 4 月二级 Java 语言笔试题题及解答	191
8.8	2008 年 9 月二级 Java 语言笔试题题及解答	198
第 9 章	二级 Java 语言模拟试题	205
9.1	笔试模拟试题	205
9.2	上机模拟试题	211
9.3	模拟试题答案	215
附录 A	Java 的体系结构	218
附录 B	Java 语法补充	220
附录 C	实用功能拾遗	224
附录 D	全国计算机等级考试二级 Java 语言考试大纲	226

第 1 章

Java 程序介绍

在这一章里通过一些简单的实例让读者（包括没有学习过计算机语言的初学者）大致了解什么是 Java 程序设计。有了大致的了解以后，逐渐深入学习后续的具体内容。

本章的基本要求

1. 学会安装使用 JDK；
2. 学会设计简单的程序，如在窗体上安装组件等；
3. 了解类与方法的概念；
4. Applet 程序的编写与运行；
5. 继承（使用）Applet 绘图；
6. 创建与删除文件或目录；
7. 了解使用简短的 Java 程序浏览网页。

本章重点

1. 通过自然数求和了解循环语句的意义；
2. 通过比较大小及计算分段函数的值了解分支语句的作用；
3. 在窗口上安装组件；
4. 绘图语句（方法）的使用。

本章难点

1. 使用 `InputStreamReader` 类与 `BufferedReader` 类完成读取操作，主要是对 `InputStreamReader` 类与 `BufferedReader` 类的理解要花费很多时间；
2. 求阶乘方法的设计与调用；
3. IP 地址的读取与转换；
4. 浏览网页程序中的一些语句，例如，

```
int v=ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;  
Container cp=getContentPane();  
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)等。
```




1.1 内容要点与典型题分析

1.1.1 Java 应用程序

1. Java 与 JDK

Java 是语言规范，而 JDK 可以看做是该语言规范的编译执行环境。依照 Java 语言规范编写程序，然后在 JDK 上得到结果。Java 的编译执行环境很多，在这里使用 JDK 作为上机环境。关于 JDK 与 SDK 的有关介绍可以参考附录 A。

虽然这一节题目是 Java 与 JDK，不过对二者都没有做过多的解释，只是讲述了如何安装使用 JDK。关于 JDK 的使用，在书中前几章中都使用了简单的方法，就是把 Java\bin 中的文件 java 与 javac 复制到当前工作目录下。之所以使用这种方法，考虑两点：一个是简单，一个是有意识地引导读者把 Java 已有的东西与自己的工作结合在一起。

关于使用 JDK，在这一节中使用了 javac 与 java 这两个命令，编译运行程序。事实上这两个命令使用频率是最高的。

注意 命令窗口不是 JDK 的组成部分，另外，编辑 Java 程序与编译 Java 程序不是一个概念。

2. 应用程序示例

在这一节中通过 16 个例题（例 1-4 至例 1-19）给出了 Java 程序的最小的缩影。具体内容有：

- 算数计算引出（for）循环语句；
- 比较排序引出分支语句；
- 利用求阶乘方法计算无理数 e 的值，含有的知识点有类内函数的构造，调用其他类的方法等；
- 窗体上安装组件是很有趣的内容，特别是使用循环语句安装组件很值得实际操作一下，在这一节中只是安装，并没有对组件进行更多的设置及编写程序；
- 下拉选择框选项的安装；
- 布局方式的设置（初次出现 `setLayout (new FlowLayout ())`）。

通过这些内容认识 Java 程序。

1.1.2 类的基本知识

1. 类的结构

Java 是一个纯粹面向对象的语言，每个程序都涉及类，所以在这里先对类做最基本的介绍。此处的介绍不是用更多的语言文字进行描述，而是使用多个小例子说明一些重要的概念，如类的组成、方法，方法的调用、继承等。

考虑到 Frame 类的重要性，把继承 Frame 类的几个例子放在这一节里。例子比较简单，学习了 `setTitle ()`、`setState ()`、`setSize ()`、`setVisible ()` 方法等。

在继承 Frame 类的程序中，定义了内部类，如 class MouseE extends 继承（也称扩展）了系统类 MouseAdapter。在这个内部类中填写 MousePressed（MouseEvent e）方法，设计实现了鼠标按下事件。也是有意把鼠标事件放在这个位置让读者初识。

接下来介绍了 TextField 类、Button 类与 Canvas 类，与其说是介绍了这几个类，还不如说是重新加深对这几类组件的认识。通过学习可以了解它们的一些常用方法。

在讲解 Canvas 类的时候，使用了一些绘图语句。

Applet 类是一个特殊的类，说它特殊主要是基于两点，一是继承这个类可以有界面、可以绘图、可以像继承窗体一样在上面安装组件；二是它的运行比较特殊，需要借助于浏览器或者使用命令 appletviewer *.html 来看结果。

有些类可以直接在 JDK 中运行（如有主方法的类），有些类是专门为其他类调用的（如例 5-3 中的类 V），有些类需要嵌入到 HTML 文件中使用（如继承 Applet 的小应用程序）。

2. 类的继承与重用

“类的继承”这个词组在前面出现过，也有一个很小的例子（例 1-22）。现在是介绍得更深入一些，通过例 1-39 讲述了方法的继承，也介绍了变量的继承。

这里提到的重用指的是方法的重用，通过 Math.sin(x) 讲解了方法重用的概念与优点。

另外也介绍了类与对象的关系，一般情况下，都是利用类名称定义对象，一个类名可以定义多个对象，每个对象都可以使用该类中的方法与变量。

【例 1-1】 使用 static 修饰类内的方法。

下面给出了两个程序。

程序一：

```
class UseFuction
{
    static int y=1,i,yn;
    public static void main(String args[])
    {
        yn=jiecheng(17);
        System.out.println(yn);
    }
    public static int jiecheng(int n)
    {
        for(i=1;i<n;i++)
        { y=y* i;}
        return(y);
    }
}
```

程序二：

```
class QiuEdeZhi
{
    static int i;
```



```
static double yn;  
public static void main(String args[])  
{  
    yn=1+1/UseFuction.jiecheng(1)+1/UseFuction.jiecheng(2)+  
    1/UseFuction.jiecheng(3);  
    System.out.println(yn);  
}  
}
```

上面两个程序都可以正常编译运行。

如果把程序一中的定义方法语句 `public static int jiecheng (int n)` 中的 `static` 去掉, 重新编译, 那么程序一旦出现错误, 只要把主方法中调用的语句 `yn = jiecheng (17);` 与 `System.out.println (yn);` 注释掉, 程序一就可编译运行。

此时编译程序二出现错误, 错误提示如下:

```
E:\Java\jdk1.6.0\example>javac QiuEdeZhi.java  
QiuEdeZhi.java:7: 无法从静态上下文中引用非静态 方法 jiecheng(int)  
yn=1+1/UseFunction.jiecheng(1)+ 1/UseFunction.jiecheng(2)+ 1/UseFunction.jiecheng(3);  
^  
QiuEdeZhi.java:7: 无法从静态上下文中引用非静态 方法 jiecheng(int)  
yn=1+1/UseFunction.jiecheng(1)+ 1/UseFunction.jiecheng(2)+ 1/UseFunction.jiecheng(3);  
^  
QiuEdeZhi.java:7: 无法从静态上下文中引用非静态 方法 jiecheng(int)  
yn=1+1/UseFunction.jiecheng(1)+ 1/UseFunction.jiecheng(2)+ 1/UseFunction.jiecheng(3);  
^  
3 错误
```

事实上有下面这样的语法规则:

类内的方法分实例方法与类方法, 用 `static` 修饰的方法是类方法。调用类方法的时候需要使用类名称作为前缀。

调用实例方法, 需要先把类实例化为对象后, 再使用对象名调用。例如, 程序一修改后保留不变。此时如果想让程序二正常编译运行, 可以把语句

`yn=1+1/UseFuction.jiecheng(1)+1/UseFuction.jiecheng(2)+1/UseFuction.jiecheng(3);` 修改为

`UseFunction a=new UseFunction();` 与

`yn=1+1/a.jiecheng(1)+1/a.jiecheng(2)+1/a.jiecheng(3);` 两个语句。

修改后, 程序二就可以正常编译运行了。

1.1.3 程序设计

有前面两节的内容作为基础, 在这里讲解程序设计就容易一些了。

1. 变量与表达式

运算符除了算术运算符外, 还有关系运算符与逻辑运算符等。

`>`、`<`、`>=`、`<=`、`==`、`!=` 都是运算符, 叫做关系运算符。

逻辑运算符有 `!`、`&&`、`||` 等, 分别表示非、与、或。

关系运算与逻辑运算的最后结果都是真或假, 即 `True` 或 `False`。

实际上在程序设计课程中, 喜欢把很多符号都叫做运算符, 例如, 赋值符号 “=” 也

被叫做赋值运算符。

常量、变量、运算符及表达式的语法规则也有很多，如转义字符的使用、运算符的优先级等（参考附录 B）。

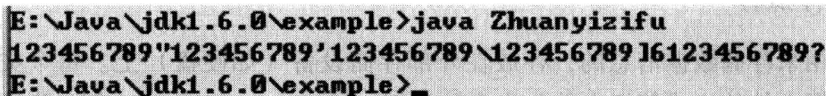
最常用的转义字符就是“\n”，在输出（等）语句中使用“\n”就是要进行换行。诸如这样的转义字符还有：“\b”表示退格，“\t”表示水平跳格，“\f”表示换页，“\r”表示回车，“\"”表示双引号，“\'”表示单引号，“\\”表示反斜杠等。

退格表示向后退一个字符的位置（相当于键盘上的〈-键〉），水平跳格一般跳 8 个字符（相当于键盘上的 Tab 键），回车是指把光标移到本行的开始处，然后执行新的输出或其他任务。

【例 1-2】 转义字符的使用。

```
public class Zhuanyizifu
{
    public static void main(String args[])
    {
        System.out.print("123456789\b");
        System.out.print("123456789\t");
        System.out.print("123456789\f");
        System.out.print("123456789\r");
        System.out.print("123456789\"");
        System.out.print("123456789\'");
        System.out.print("123456789\\");
        System.out.print("123456789\1356");
        System.out.print("123456789\u1356");
    }
}
```

该程序运行后的结果如图 1-1 所示。



```
E:\Java\jdk1.6.0\example>java Zhuanyizifu
123456789'123456789'123456789\123456789 l6123456789?
E:\Java\jdk1.6.0\example>_
```

图 1-1 转义字符的使用

可以进一步修改程序，分析各个转义字符的意义。

程序中，“\xxx”表示编码是 xxx（八进制）的 Latin-1 字符，“\uxxxx”表示编码是 xxxx（十六进制）的 Unicode 字符。

【例 1-3】 运算符的优先级与结合性。

```
public class yusuanfuyuxianji
{
    public static void main(String args[])
    {
        int a=12,b=7,c=9;
```



```
        int x = a > b + c ? a * b : a % c / 2;
        System.out.println(x);
    }
}
```

该程序的运行结果是 1。

语句 `int x = a > b + c ? a * b : a % c / 2;` 中, ? 的优先级最高, ? 结合在一起的语句也叫做问号表达式, 有时也叫做三目运算表达式。

在计算三目运算表达式的时候, 首先看 ? 前面的表达式是真还是假, 如果是真, 则计算 `a * b`; 如果是假, 则计算 `a % c / 2`。实际上三目运算表达式相当于分支语句。

在这个例题中, 因为 `a > b + c` 不成立, 所以先计算 `a % c / 2`。因为 % 与 / 优先级相同, 结合性是从左向右, 所以先计算 `a % c` 得到 3, `3 / 2` 得到 1。

最后把 1 赋给变量 `x`。

初学时, 如果运算符的优先级与结合性掌握不好的话, 可以尽量使用小括号, 把要先计算的数或者变量括起来。

2. 分支

分支是一种重要的程序结构。前面已经接触了分支语句, 教材通过一个具体的问题如求一元二次方程的根及求分支语句的执行次数这两个例题, 进一步加深对分支语句的理解。

3. 循环

循环是计算机语言最重要的功能。

通过计算循环语句的执行次数加深对循环语句的理解; 利用循环语句绘图给出了循环的直观解释。

实际上在这本书中, 语法内容的安排是从“小节”(1.1.2 节)到“中节”(1.3 节)再到“章”(第 4 章), 这样逐渐深入展开。

1.1.4 文件操作

文件是用来存储数据信息的, 存放在外部存储设备(如硬盘)上。每个文件都有自己的名字, 根据文件的格式等把文件分为多种类型, 如 `.txt`、`.exe`、`.doc` 等。为了有效地组织文件, 可以在硬盘中设置文件夹。

文件及文件夹的操作可以由操作系统来完成, 读写文件也可以用特有的软件来完成, 如记事本、Word 等。数据库文件可以用数据库软件系统建立与管理。

可是有很多场合需要使用语言来建立或者删除目录, 需要读文件、写文件。例如, 开发一个应用程序, 需要在程序运行过程中建立一个文件, 然后向这个文件中写入数据, 之后再从该文件中读取数据, 当程序运行后自动删除这个文件, 这时就需要使用语言操作文件。

1. 文件与目录

学习这一节时, 尽量通过最简单的程序了解如何使用 Java 语言创建目录、删除目录、创建文件、删除文件。



2. 读文件

读文件指的是从硬盘文件中读取数据或信息。给出了一个例题（例 1-57）是把文本文件中的所有（字节）内容都读取出来。这是读取文件的一个最基本、最常用的功能。

读写文件一般都使用“流”来处理，其实文件对象本身也提供了很多方法，例如，使用文件类定义了对象后，文件对象可以调用文件类的方法 `public long length()`，该方法返回文件的长度。

【例 1-4】 方法 `public long length()` 能够返回以字节为单位的文件长度，调用的时候使用 `FILE` 对象调用。编写程序返回某个文件的长度，然后利用该长度把文件内容全部读取出来。

设计下面程序：

```
import java.io.*;

public class duwenjian11
{
    public static void main(String args[])
    {
        int chi;
        File MyFile1 = new File("C:", "e1.txt");
        try{
            FileInputStream fin = new FileInputStream(MyFile1);
            for(int i = 0; i < MyFile1.length(); i++)
            {
                chi = fin.read();
                System.out.print((char)chi);
            }
            fin.close();
        }
        catch(IOException e) {
            System.err.println(e);
        }
    }
}
```

该程序编译运行后，能够把文件 `e1.txt`（存储在当前工作目录下）的内容全部读取出来，显示在命令窗口中。

3. 写文件

教材中给出了一个写文件程序作为例子，该例子是运行程序后，在命令窗口中输入字符，直到输入“#”时，按 `Enter` 键退出程序。该程序存在的最大不足是每次写文件的时候，文件中原先的内容就被清空。如果想保留原先内容的话，可以把语句

```
FileOutputStream fout = new FileOutputStream(MyFile1);
```

修改为

```
FileOutputStream fout = new FileOutputStream(MyFile1, true);
```



这样就可以实现文件添加功能了，如例 1-5 所示。

【例 1-5】 向文件中添加内容。

设计下面程序：

```
import java.io.*;
public class xiewenjian11
{
    public static void main(String args[])
    {
        char ch;
        File MyFile1=new File("C:", "e1.txt");
        try
        {
            FileOutputStream fout=new FileOutputStream(MyFile1,true);
            while((ch=(char)System.in.read())!='#')
                fout.write(ch);
            fout.close();
        }
        catch(IOException e)
        {
            System.err.println(e);
        }
    }
}
```

该程序运行后，在命令窗口输入要添加的信息，最后输入#，然后按 Enter 键，输入的内容便被添加到 C 盘根目录下文件 e1.txt 中。用记事本打开该文件发现原先内容还在，新写的内容被添加进去了。

1.1.5 高层网络编程

把这一节的名称叫做“高层网络编程”是有意识地让读者了解网络程序设计是分层的，并且一般来说越高层越简单。

这一节的内容有：取本地主机名，取 IP 地址，利用主机名返回 IP 地址，设计程序浏览网页，完善浏览网页程序等。

重点内容是设计与完善浏览网页程序。

一个非常简短的程序就可以浏览网页，也许这正是 Java 魅力的第一次展示。

1.2 教材课后习题提示与解答

一、实验分析题

1. 下面程序能够把输入的三个数从小到大排序，请分析该程序是如何实现这一功能的。



```
public class C1
{
    public static void main (String args[])
    {
        int a,b,c,t=0;
        a=Integer.parseInt (args[0]);
        b=Integer.parseInt (args[1]);
        c=Integer.parseInt (args[2]);
        if (a>b)
        {
            t=a;a=b;b=t;
        }
        if (a>c)
        {
            t=a;a=c;c=t;
        }
        if (b>c)
        {
            t=b;b=c;c=t;
        }
        System.out.println("顺序为"+a+" "+b+" "+c);
    }
}
```

提示: $a > b$, $a > c$, $b > c$ 三个条件都不成立的时候, 自然排好序了; $a > b$, $a > c$ 两个条件语句都不成立的时候, a 就是最小的, 所以只要互换 b 与 c 就可以了。

2. 下面程序也是绘制图形, 编译运行该程序, 观察分析运行结果。

```
import java.awt.*;
class qipan extends Frame
{
    //源程序中应该加上下面的 main 方法
    public static void main (String[] args) {
        new qipan();
    }
    qipan()
    {
        FlowLayout f=new FlowLayout();
        setLayout(f);
        for(int i=1;i<=16;i++)
        {
            if(i<10)
            {
```




```

        Button b=new Button("第" + " " + i + "个");
        add(b);
    }
    else
    {
        Button b=new Button("第" + i + "个");
        add(b);
    }
}
setBounds(100,100,240,160);
setVisible(true);
}
}

```

3. 比较两个数大小，然后输出较大的数，可以使用下面程序语句：

```

if (a>b)
    b=a;
print  b

```

分析这个程序与例 1-10 的异同。

4. 在例 1-38 中，把 class UseClass 中的语句 UseFuctionNew u2 = new UseFuctionNew (); 去掉，把语句 y1 = u1.jiecheng (3) + u2.jiecheng (5); 修改为 y1 = u1.jiecheng (3) + u1.jiecheng (5); 程序是否一样能够计算 3 的阶乘与 5 的阶乘的和？

做实验调试运行，观察分析得到的结果。

5. 把例 1-42 中定义变量的 float 改为 double，其他都不改变，观察实验结果。

6. 例 1-43 中，如果把语句 k = Math.sqrt (e * e + g * g) 与 System.out.println (k) 合成一句 System.out.println (Math.sqrt (e * e + g * g))，调试程序，看是否可以正常运行。

7. 调试下面程序，分析运行后得到的结果或者错误信息。

```

import java.io.*;
public class Testmkdir1
{
    public static void main(String args[])
    {
        String sdir = "C:/Program Files/Java/jdk1.6.0/example/e1";
        File Fdir1 = new File(sdir);
        Fdir1.mkdir();
    }
}

```

8. 调试下面程序，分析运行后得到的结果或者错误信息。

```

import java.io.*;
public class Testmkdir1

```