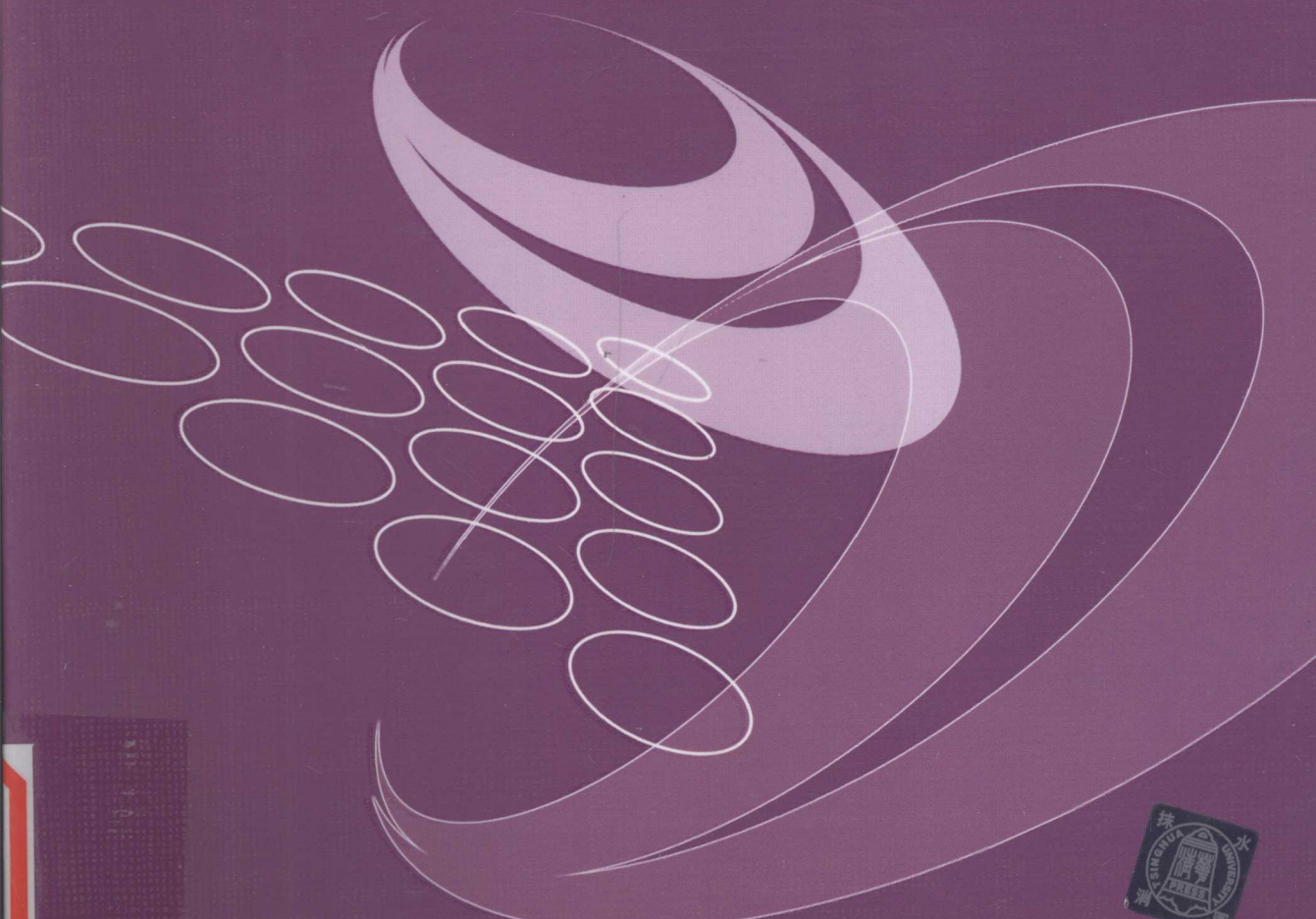
普通高等院校
计算机专业（本科）实用教程系列

数据结构实用教程

（Java语言描述）习题参考解答

徐孝凯　编著



清华大学出版社

普通高等院校计算机专业（本科）实用教程系列

数据结构实用教程（Java 语言描述）习题参考解答

徐孝凯　编著

福建师范大学
图书馆
藏书印记

1052127

T1052127

清华大学出版社
北京

内 容 简 介

本书是与作者编著的《数据结构实用教程（Java 语言描述）》一书相配套的辅助教材。全书共分为 11 章，包括绪论、集合、线性表、稀疏矩阵和广义表、栈和队列、树和二叉树、常用二叉树、图、图的应用、查找、排序。每章给出了相应内容的知识要点、练习题和参考解答。练习题包括选择题、填空题、运算题、算法分析题、算法设计题等题型，算法分析和设计题都是采用目前最实用的 Java 语言描述的，并且还给出了与算法相应的调试程序和上机运行过程，以及必要的解题思路。

本书题型丰富，涉及数据结构课程的全部内容，使用本书既能加深对数据结构基本概念的理解和认识，又能提高对各种数据结构进行运算的算法分析与设计能力。本书可以作为读者学习数据结构课程的辅助教材，也可以作为参加数据结构课程研究生考试的复习参考书。

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　13701121933

图书在版编目（CIP）数据

数据结构实用教程（Java 语言描述）习题参考解答 / 徐孝凯编著. --北京：清华大学出版社，2013.1
普通高等院校计算机专业（本科）实用教程系列
ISBN 978-7-302-30701-3

Ⅰ. ①数… Ⅱ. ①徐… Ⅲ. ①数据结构-高等学校-题解 ②JAVA 语言-程序设计-高等学校-题解
Ⅳ. ①TP311.12-44 ②TP312-44

中国版本图书馆 CIP 数据核字（2012）第 278589 号

责任编辑：郑寅堃　王冰飞
封面设计：张　昱
责任校对：时翠兰
责任印制：李红英

出版发行：清华大学出版社
　　网　　址：http://www.tup.com.cn，http://www.wqbook.com
　　地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
　　社 总 机：010-62770175　　**邮　　购**：010-62786544
　　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　　质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：8.25　　**字　　数**：201 千字
版　　次：2013 年 1 月第 1 版　　**印　　次**：2013 年 1 月第1 次印刷
印　　数：1～2000
定　　价：17.00 元

产品编号：047496-01

前　言

本书是与作者编著的《数据结构实用教程(Java 语言描述)》一书相配套的辅助教材。全书共分为 11 章，包括绪论、集合、线性表、稀疏矩阵和广义表、栈和队列、树和二叉树、常用二叉树、图、图的应用、查找、排序。每章给出了相应内容的知识要点、练习题和参考解答。练习题包括选择题、填空题、运算题、算法分析题、算法设计题等题型，算法分析和设计题都是采用目前最实用的 Java 语言描述的，并且还给出了与算法相应的调试程序和上机运行过程，以及必要的解题思路。

本书既给出了练习题，又给出了相应的参考解答，希望读者不要依赖于现成的答案，要凭借自己所学的知识，尽量独立地思考问题和解决问题，做出解题答案，然后再同书中的答案比较，相互借鉴。书中所给的答案不是唯一正确的，特别对于算法设计题，更是如此，有些可能存在着缺点和不足。所以，读者要充分相信自己，你编写出的算法可能会更好。对于很难独立做出的一部分练习题，可以从现成答案中得到启发。希望本书能够成为帮助读者学习和提高数据结构知识的良师益友。

数据结构是一门实践性很强的课程。对于自己编写的每一个算法，最好通过上机调试，验证其正确性和有效性。在上机调试的过程中，必须通过各种典型的数据输入使得算法中的每条语句都被执行过，或者说不存在没有被执行过的语句或语句块。若调试过程发现语法或逻辑错误，则要及时修改。所谓逻辑错误，是指算法设计上隐含的错误，虽然算法能够被正确地编辑和连接，但运行后得不到正确的结果。通过上机操作能够学习到书本上很难学到的实际知识和经验。

在本书习题和解答中采用的各种数据接口和存储类型，都在主教材《数据结构实用教程（Java 语言描述）》一书中有着详细的介绍和讨论。所以，希望读者能够与主教材一起配套使用，以取得良好的学习效果。

书中所有算法和程序都在 Java 开发和运行环境下编译和调试通过，确保了语法和逻辑设计的正确性。

本书题型丰富，涉及数据结构课程的全部内容，使用本书既能加深对数据结构基本概念的理解和认识，又能提高对各种数据结构进行运算的算法分析与设计能力。本书可以作为读者学习数据结构课程的辅助教材，也可以作为参加数据结构课程研究生考试的复习参考书。

徐孝凯

2012 年 9 月

目　录

试读结束：需要全本请在线购买：　www.ertongbook.com

第1章 绪 论

1.1 知识要点

1．数据结构包括逻辑结构和存储结构两个方面。数据的逻辑结构被分为集合结构、线性结构、树型结构和图型结构4种。数据的存储结构被分为顺序结构、链接结构、索引结构和散列结构4种。数据的一种逻辑结构可以选用一种或几种较合适的存储结构来实现，即被有效地存储到计算机的存储系统中。

2．在集合结构中，不考虑数据之间的任何次序，它们是各自独立的。在线性结构中，数据之间是一对一的关系。在树型结构中，数据之间是一对多的关系。在图型结构中，数据之间是多对多的关系。

3．一个数组占有一块连续的存储空间，前后相邻的元素其存储位置也相邻，或者说，元素之间的线性关系通过存储位置顺序相邻的关系反映出来。因此，对于线性结构的数据，通常是利用一维数组来实现其顺序存储结构，让线性结构中的第1个数据元素存储到数组中下标为0的元素中，第2个数据元素存储到数组中下标为1的元素中，依次类推。

4．抽象数据类型是数据和对其进行各种操作的定义体。这里所说的数据是广义的，是带有结构的数据，它可以具有任何逻辑结构和存储结构。在Java语言中，抽象数据类型利用接口或类的定义来实现。

5．算法的评价指标主要为正确性、健壮性、可读性和有效性 4 个方面。有效性又包括时间复杂度和空间复杂度两个方面。若一个算法的时间复杂度和空间复杂度越好，则说明它越节省运行时间和存储空间，也表明算法越有效。

6．算法的时间复杂度和空间复杂度通常用数量级的形式表示出来。数量级的形式可分为常量级、对数级、线性级、线性乘对数级、平方级、立方级、指数级、阶乘级等多个级别。当数据处理量较大时，处于前面级别的算法比处于后面级别的算法更有效。

1.2 练习题

1.2.1 单选题

1．若一个数据具有集合结构，则元素之间具有（　　）。

A．线性关系　　B．层次关系　　C．网状关系　　D．无任何关系

2．数据逻辑结构包括的4种结构类型为（　　）。

A．集合结构、线性结构、树型结构、图型结构

B．顺序结构、链接结构、索引结构、散列结构

C．数组、记录、字符串、文件

D．树型结构、图型结构、文件结构、表结构

3．下面二元组表示的数据结构B属于（　　）。

B=(*K*,*R*)，其中：

K={a,b,c,d,e,f,g,h}

R={r}

r={<a,b>,<b,c>,<c,d>,<d,e>,<e,f>,<f,g>,<g,h>}

A．线性结构　　B．集合结构　　C．树型结构　　D．图型结构

4．下面二元组表示的数据结构C属于（　　）。

C=(*K*,*R*)，其中：

K={a,b,c,d,e,f,g,h}

R={<d,b>,<d,g>,<b,a>,<b,c>,<g,e>,<g,h>,<e,f>}

A．线性结构　　B．集合结构　　C．树型结构　　D．图型结构

5．下面二元组表示的数据结构D属于（　　）。

D=(*K*,*R*)，其中：

K={1,2,3,4,5,6}

R={(1,2),(2,3),(2,4),(3,4),(3,5),(3,6),(4,5),(4,6)}

A．线性结构　　B．集合结构　　C．树型结构　　D．图型结构

6．下面二元组表示的数据结构E属于（　　）。

E=(*K*,*R*)，其中：

K={48,25,64,57,82,36,75,43}

R={<48,25>,<48,64>,<64,57>,<64,82>,<25,36>,<82,75>,<36,43>}

A．线性结构　　B．集合结构　　C．树型结构　　D．图型结构

7．下面二元组表示的数据结构F属于（　　）。

F=(*K*,*R*)，其中：

K={48,25,64,57,82,36,75,43}

R={<48,25>,<48,64>,<64,57>,<64,82>,<25,36>,<82,75>,<36,43>,<57,75>}

A．线性结构　　B．集合结构　　C．树型结构　　D．图型结构

8．设有一个二维数组a[*m*][*n*]，按行存放于一个连续的存储空间中，则a[*i*][*j*]元素（$0 \leqslant i \leqslant m-1, 0 \leqslant j \leqslant n-1$）的存储位置序号为（　　）。假定a[0][0]存储位置序号为1。

A．$i \times m+j$　　B．$i \times n+j+1$　　C．$i \times m+j-1$　　D．$i \times n+j$

9．设有一个int型二维数组A[10][20]，按行存放于一个连续的存储空间中，并已知A[0][0]的存储字节地址为200，每个元素占4个字节，则A[6][12]的存储字节地址为（　　）。

A．728　　B．332　　C．692　　D．400

10．一个算法的时间复杂度为$(3n^2+2n+7)/(5n)$，其数量级表示为（　　）。

A．$O(n)$　　B．$O(n^2)$　　C．$O(n^3)$　　D．$O(1)$

11．下面程序段的时间复杂度为（　　）。

```
for(int i=0; i<m; i++)
    for(int j=0; j<n; j++)
        a[i][j]=i*j;
```

A. $O(m^2)$　　B. $O(n^2)$　　C. $O(m*n)$　　D. $O(m+n)$

12. 执行下面程序段时，表示内循环体的S语句的执行次数为（　　）。

```
for(int i=1; i<=n; i++)
    for(int j=1; j<=i; j++) S;
```

A. n^2　　B. $n^2/2$　　C. $n(n+1)$　　D. $n(n+1)/2$

13. 执行下面程序段时，表示内循环体的S语句的执行次数为（　　）。

```
for(int i=1; i<n; i++)
    for(int j=1; j<=i; j++) S;
```

A. $n(n-1)$　　B. $n(n-1)/2$　　C. $n(n+1)$　　D. $n(n+1)/2$

14. 执行下面程序段时，表示内循环体S语句的执行次数为（　　）。

```
for(int i=1; i<n; i++)
    for(int j=i-1; j>=0; j--) S;
```

A. $n(n-1)/2$　　B. $n(n-1)$　　C. $n(n+1)/2$　　D. $n(n+1)$

15. 执行下面程序段时，表示内循环体S语句的执行次数为（　　）。

```
for(int i=0; i<n; i++)
    for(int j=i; j<n; j++) S;
```

A. $n(n-1)/2$　　B. $n(n-1)$　　C. $n(n+1)/2$　　D. $n(n+1)$

16. 执行下面程序段时，for循环体语句被执行的次数为（　　）。

```
int i,s=0;
for(i=1; s<20; i++) s+=i*i;
```

A. 3　　B. 4　　C. 5　　D. 6

17. 执行下面程序段时，if语句被执行的次数为（　　）。

```
int f(int n) {
    if(n==1) return 1;
    else return n*f(n-1);
}
```

A. n　　B. $n-1$　　C. $n+1$　　D. 1

18. 下面算法的时间复杂度为（　　）。

```
int f(int n) {
    if(n==0 || n==1) return 1;
    else return n*f(n-1);
```

试读结束：需要全本请在线购买： www.ertongbook.com

```
}
```

A. O(1)　　B. O(n)　　C. O(n^2)　　D. O(n!)

19. 使用 f(5)调用下面算法时，返回的函数值为（　　）。

```
int f(int n) {
    if(n==1) return 1;
    else return n*n+f(n-1);
}
```

A. 25　　B. 40　　C. 50　　D. 55

20. 使用 f(4)调用下面算法时，该算法被调用执行的总次数为（　　）。

```
int f(int n) {
    if(n==0 || n==1) return n;
    else return f(n-1)+f(n-2);
}
```

A. 7　　B. 8　　C. 9　　D. 10

21. 使用 f(4)调用下面算法时，该算法被执行后返回的函数值为（　　）。

```
int f(int n) {
    if(n==0 || n==1) return n;
    else return f(n-1)+f(n-2);
}
```

A. 2　　B. 3　　C. 4　　D. 5

22. 从一个数组 a[7]中顺序查找元素时，假定查找第一个元素 a[0]的概率为 1/3，查找第二个元素 a[1]的概率为 1/4，查找其余元素的概率均相同，则在查找成功时同元素的平均比较次数为（　　）。

A. 3　　B. 35/12　　C. 25/12　　D. 32/12

1.2.2 算法分析题

指出下列各算法的功能并求出其时间复杂度。

1.

```
boolean prime(int n)
{
    int i=2;
    while(i*i<=n)
        if(n%i==0) break; else i++;
    return i*i>n;
}
```

2.

```
int sum1(int n)
```

```
{
    int i, p=1, s=0;
    for(i=1; i<=n; i++){ p*=i; s+=p;}
    return s;
}
```

3.

```
int sum2(int n)
{
    int i, s=0;
    for(i=1; i<=n; i++){
        int j, p=1;
        for(j=1; j<=i; j++) p*=j;
        s+=p;
    }
    return s;
}
```

4.

```
int minimize(int n)
{
    int i=0,s=0;
    while(s<n) s+=++i;
    return i;
}
```

5.

```
void multiTable(int n)
{
    for(int i=1; i<=n; i++)
    {
        for(int j=i; j<=n; j++)
            System.out.print(i+"*"+j+"="+i*j+"  ");
        System.out.println();
    }
}
```

6.

```
int greatCount(int[][] a, int k)
{
    int c=0;
    for(int i=0; i<a.length; i++)
        for(int j=0; j<a[i].length; j++)
            if(a[i][j]>=k) c++;
    return c;
}
```

7.

```
void matriMult(int[][] a, int[][] b, int[][] c)
{
    if(a[0].length!=b.length) {
        System.out.println("a 和 b 不匹配，不能相乘，退出运行!");
        System.exit(1);
    }
    int i,j,k;
    for(i=0;i<a.length;i++)
        for(j=0;j<b[0].length;j++)
            c[i][j]=0;
    for(i=0;i<a.length;i++)
        for(j=0;j<b[0].length;j++)
            for(k=0;k<a[0].length;k++)
                c[i][j]+=a[i][k]*b[k][j];
}
```

1.2.3 算法设计题

设计二次多项式 ax^2+bx+c 的一种抽象数据类型，假定起名为 Quadratic，该类型的数据部分为双精度类型的 3 个系数项 a、b 和 c，操作部分为：

（1）初始化二次多项式中的 3 个数据成员 a、b 和 c。

Quadratic(double aa, double bb, double cc);

（2）做两个多项式加法，即它们对应的系数相加，返回相加结果。

Quadratic add(Quadratic q);

（3）根据给定 x 的值计算多项式的值并返回。

double value(double x);

（4）计算多项式等于 0 时的两个实数根，对于有实根、无实根和不是二次方程（即 a==0）这 3 种情况需要返回不同的整数值(1,0,–1)，以便调用函数能够做不同的处理。当有实数根时，分别用 r[1]和 r[2]保存所得到的两个实数根。

int seekRoot(double[] r);

（5）按照 $a*x^2+b*x+c$ 的格式（x^2 用 x^2 表示）输出二次多项式，在输出时要注意去掉系数为 0 的项，并且当 b 和 c 的值为负时，其前不能出现加号。

void print();

请写出上面的抽象数据类型所对应的 Java 类。

1.3 练习题参考解答

1.3.1 单选题

1. D　2. A　3. A　4. C　5. D

6．C　7．D　8．B　9．A　10．A
11．C　12．D　13．B　14．A　15．C
16．B　17．A　18．B　19．D　20．C
21．B　22．B

部分单选题解析：

19．f(5)=5*5+f(4)=25+4*4+f(3)=25+16+3*3+f(2)=25+16+9+2*2+f(1)
=25+16+9+4+1=55

20．f(4)->{f(3)->{f(2)->{f(1),f(0)},f(1)},f(2)->{f(1),f(0)}}，共有 9 次调用。

21．f(4)=f(3)+f(2)=[f(2)+f(1)]+[f(1)+f(0)]=[f(1)+f(0)+1]+[1+0]
=[1+0+1]+[1]=2+1=3

22．在含有 n 个元素的数据表中顺序查找任一元素的平均比较次数为 $\sum_{i=1}^{n} p_i c_i$，p_i 为查找第 i 个元素的概率，c_i 是查找第 i 个元素时需要比较的元素数，查找所有元素的概率之和为 1，若查找每个元素的概率相同，则平均查找长度的计算公式可简化为 $\frac{1}{n}\sum_{i=1}^{n} c_i$。

此题的计算式为

$$\frac{1}{3}\times 1+\frac{1}{4}\times 2+\frac{1}{12}(3+4+5+6+7)=35/12$$

1.3.2　算法分析题

1．判断 n 是否为一个素数，若是则返回逻辑值 true，否则返回逻辑值 false。该算法的时间复杂度为 $O(\sqrt{n})$。

分析：此函数算法的时间复杂度取决于 while 循环体被执行的次数，此次数不超过 $\sqrt{n}$ 次，因为当 i 从 2 循环到 $\sqrt{n}$ 时才执行循环体。

2．计算 $\sum_{i=1}^{n} i!$ 的值。时间复杂度为 $O(n)$。

3．计算 $\sum_{i=1}^{n} i!$ 的值。时间复杂度为 $O(n^2)$。

分析：此算法的时间复杂度取决于内层 for 循环体语句“p*=j”被执行的次数，此次数为 1+2+3+…+n，即等于 $n(n+1)/2$，所以其时间复杂度为 $O(n^2)$。

4．求出满足不等式 1+2+3+…+$i\geqslant n$ 的最小 i 值。时间复杂度为 $O(\sqrt{n})$。

分析：因为 s 的值为 1+2+3+…+i，即 $(1+i)i/2$，当 $(1+i)i/2\geqslant n$ 时循环结束，所以当 n 很大时，i 的平方与 n 成正比，也就是说 i 的值（while 循环的次数）与 n 的平方根成正比。

5．打印出一个具有 n 行的乘法表，第 i 行（$1\leqslant i\leqslant n$）中有 $n-i+1$ 个乘法项，每个乘法项为 i 与 j（$i\leqslant j\leqslant n$）的乘积。时间复杂度为 $O(n^2)$。

6．统计并返回二维数组 a 中大于等于 k 的元素的个数。时间复杂度为 $O(m\times n)$，假定 m 和 n 分别表示二维数组 a 的行数 a.length 和列数 a[0].length。

7．矩阵相乘，即 a[m][n]×b[n][p]→c[m][p]。时间复杂度为 $O(m\times n\times p)$。这里假定二维数组 a 的行列数为 m 和 n，二维数组 b 的行列数为 n 和 p，二维数组 c 的行列数为 m 和 p。

1.3.3 算法设计题

抽象数据类型如下：

```
ADT Quadratic  is
   Data:
      double a,b,c;                                  //二次项、一次项和常数项系数
   Operations:
      public Quadratic(double aa, double bb, double cc);    //构造函数
      public Quadratic add(Quadratic q);     //二次多项式相加
      public double value(double x);         //二次多项式求值
      public int seekRoot(double[] r);       //二次多项式方程求解
      public void print();                   //输出二次多项式
end Quadratic
```

Java 类参考答案如下：

```
public class Quadratic
{
    private double a,b,c;
    public Quadratic(double aa, double bb, double cc) {
       a=aa; b=bb; c=cc;
    }
    public Quadratic add(Quadratic q)  {
        Quadratic qq=new Quadratic(0,0,0);
        qq.a=a+q.a;
        qq.b=b+q.b;
        qq.c=c+q.c;
        return qq;
    }
    public double value(double x) {
        return a*x*x+b*x+c;
    }
    public int seekRoot(double[] r) {
        if(a==0) return -1;  //不是二次方程返回-1
        double x=b*b-4*a*c;
        if(x>=0){
            r[1]=(-b+Math.sqrt(x))/(2*a);
            r[2]=(-b-Math.sqrt(x))/(2*a);
            return 1;          //有实数根返回 1
        }
        else
            return 0;          //有虚数根返回 0
    }
     public void print() {
```

```
        if(a!=0.0)              //输出二次项
            System.out.print(a+"*x^2");
        if(b!=0.0)              //输出一次项
            if(b>0) System.out.print("+"+b+"*x");
            else if(b<0) System.out.print(b+"*x");
        if(c!=0.0)  //输出常数项
            if(c>0) System.out.print("+"+c);
            else if(c<0) System.out.print(c);
        System.out.println();
    }
}
```

用于调试的主函数类如下：

```
public class Lianxi1_2
{
    public static void main(String[] args)
    {
        double a1=1,b1=9,c1=18;
        double a2=3,b2=-8,c2=5;
        Quadratic q1=new Quadratic(a1,b1,c1);
        Quadratic q2=new Quadratic(a2,b2,c2);
        Quadratic q3;

        q3=q1.add(q2);
        double x=q1.value(2);
        double[] r=new double[3];
        int t1=q1.seekRoot(r);
        if(t1==-1) System.out.println("不是二次方程!");
        else if(t1==0) System.out.println("有虚数根!");
        else System.out.println("有实数根!");

        q1.print();
        q2.print();
        q3.print();
        System.out.println(x+" "+r[1]+" "+r[2]);
    }
}
```

运行结果如下：

```
D:\tsinghua>javac Quadratic.java
D:\tsinghua>javac Lianxi1_2.java
D:\tsinghua>java  Lianxi1_2
有实数根!
1.0*x^2+9.0*x+18.0
3.0*x^2-8.0*x+5.0
4.0*x^2+1.0*x+23.0
40.0 -3.0 -6.0
```

试读结束：需要全本请在线购买： www.ertongbook.com