

Java语言基础 Day04

1. [水仙花数问题](#)
2. [猜字母游戏](#)
3. [质数问题（选做）](#)
4. [随机加法运算器（选做）](#)
5. [成绩统计程序](#)
6. [费氏数列问题](#)

1 水仙花数问题

1.1 问题

计算1000以内的所有水仙花数。用户输入0到1000以内的某个数，系统输出这个范围内的所有水仙花数，水仙花数是指一个 n 位数 ($n \geq 3$)，它的每个位上的数字的 n 次幂之和等于它本身。(例如： $1^3 + 5^3 + 3^3 = 153$)。系统交互情况如图-36所示：

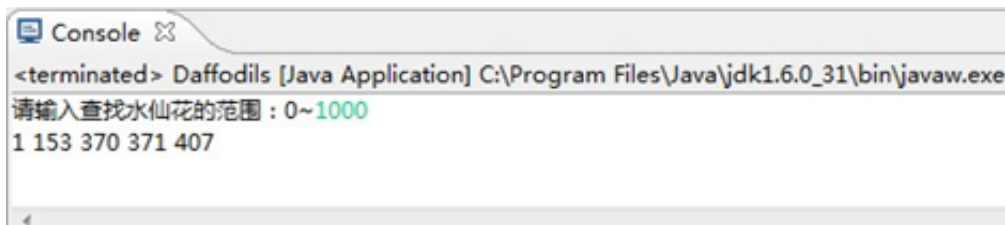


图-36

1.2 方案

系统使用双层for循环判断某个数是否为水仙花数，外层循环确保每个用户输入的范围内的数都判断到，里层循环判断某个数是否为水仙花数， $m \% 10$ 取到个位数，例如： $\text{int } m = 153$ ，经过 $\text{int } v = m \% 10$ 计算后 $v = 3$ ； $m /= 10$ 去除各位数，例如： $\text{int } m = 153$ ，经过 $m /= 10$ 计算后 $m = 15$ 。代码如下所示：

```
01.     for (int n = 3; n <= max; n++) {  
02.         // 判断n是否为水仙花数  
03.         int sum = 0;  
04.         for (int m = n; m > 0; m /= 10) {  
05.             int v = m % 10;  
06.             sum += v * v * v;  
07.         }  
08.  
09.     }
```

如果 n 和 sum 相等，那么说明 n 是水仙花数。代码如下所示：

```
01.     if (n == sum) {
02.         System.out.print(n + " ");
03.     }
```

1.3 实现

系统代码实现如下：

```
01.     import java.util.Scanner;
02.     public class Daffodils {
03.         public static void main(String[] args) {
04.             Scanner scanner = new Scanner(System.in);
05.             System.out.print("请输入查找水仙花的范围: 0~");
06.             int max = scanner.nextInt();
07.             scanner.close();
08.             for (int n = 3; n <= max; n++) {
09.                 // 判断n是否为水仙花数
10.                 int sum = 0;
11.                 for (int m = n; m > 0; m /= 10) {
12.                     int v = m % 10;
13.                     sum += v * v * v;
14.                 }
15.                 if (n == sum) {
16.                     System.out.print(n + " ");
17.                 }
18.             }
19.         }
20.     }
```

[隐藏](#)

1.4 扩展(选做)

计算“百钱买百鸡”的问题。

百钱买百鸡：公鸡5文钱一只，母鸡3文钱一只，小鸡1文钱3只，如何用百文钱买百只鸡？

系统交互情况如图-37所示：

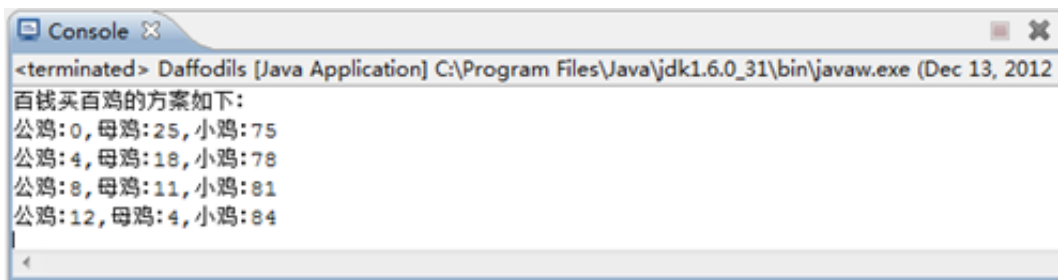


图-37

```
01.     public class DaffodilsExt {
02.         public static void main(String[] args) {
03.             int cock, hen, chick;
04.             System.out.println("百钱买百鸡的方案如下: ");
05.             for (cock = 0; cock <= 20; cock++) {
06.                 for (hen = 0; hen <= 33; hen++) {
07.                     chick = 100 - cock - hen;
08.                     if (chick % 3 == 0) {
09.                         int total = cock * 5 + hen * 3 + chick / 3;
10.                         if (total == 100) {
11.                             System.out.println("公鸡: " + cock + ", 母鸡: "
12.                                 + hen + ", 小鸡: " + chick);
13.                         }
14.                     }
15.                 }
16.             }
17.         }
18.     }
```

隐藏

2 猜字母游戏

2.1 问题

猜字母游戏。系统随机产生5个按照一定顺序排列的字符，然后由用户输入一个5个字符的字符串，由程序判断这五个字符和系统所产生的5个字符是否相同（字母和位置均相同）。如果相同，则程序结束，并计算得分；如果不相同，则输出比较结果以提示用户继续游戏。系统交互情况如图-48所示：



```
<terminated> GuessingGame [Java Application] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe (Dec 19, 2011 10:10:10 AM)
GuessingGame>欢迎尝试猜字母游戏!
GuessingGame>游戏开始, 请输入你所猜的5个字母序列: (exit—退出)
ABCDE
GuessingGame>你猜对1个字符, 其中0个字符的位置正确! (总次数=1, exit—退出)
DEFGH
GuessingGame>你猜对1个字符, 其中1个字符的位置正确! (总次数=2, exit—退出)
DIJKL
GuessingGame>你猜对2个字符, 其中2个字符的位置正确! (总次数=3, exit—退出)
DXYZL
GuessingGame>你猜对4个字符, 其中3个字符的位置正确! (总次数=4, exit—退出)
DYZL
GuessingGame>恭喜你猜对了! 你的得分是: 450
```

图-48

由图 - 48可以看出, 程序每次比较用户所录入的字符时, 会提示5个字符中正确的字符个数, 以及位置正确的个数作为提示, 以便于用户判断。

游戏的得分规则为: 字符的个数乘以100为总分, 即此游戏的总分为 500 分。用户如果第一次就猜对, 则得满分 (500分); 每多猜测一次, 则扣10分。

2.2 方案

定义一个 5 个长度的字符数组用于存储系统所产生的5个字符, 并定义变量记载用户所猜测的次数。代码如下:

```
01.     int level = 5;
02.     char[] chs = new char[level];
03.     int score = 0;
```

定义一个数组letters存放所有字符, 定义一个boolean类型的数组大小和letters数组大小相同, 用来标识letters数组中的元素是否被选中, 代码如下:

[Top](#)

```
01.     char[] letters = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
02.                        'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V',
03.                        'W', 'X', 'Y', 'Z' };
04.     boolean[] flags = new boolean[letters.length];
```

根据所需要产生的字符个数 (5个) 随机产生字符数组, 且字符不能重复。代码如下:

```
01.     for (int i = 0; i < chs.length; i++) {
02.         int index;
03.         do {
```

```

04.         index = RandomUtils.nextInt(letters.length);
05.     } while (flags[index]);
06.     chs[i] = letters[index];
07.     flags[index] = true;
08. }

```

然后，提示用户录入5个字符后，并转换为字符数组，代码如下：

```

01.     String inputStr = scanner.next().trim().toUpperCase();
02.     char[] input = inputStr.toCharArray();

```

判断用户的录入的正确性（需要同时判断字符和位置的正确性），代码如下：

```

01.     int[] result = new int[2];
02.     // result[0]: 存储正确的位置个数
03.     // result[1]: 存储正确的字符个数
04.     for (int i = 0; i < input.length; i++) {
05.         for (int j = 0; j < chs.length; j++) {
06.             if (input[i] == chs[j]) {
07.                 result[1]++;
08.                 if (i == j) {
09.                     result[0]++;
10.                 }
11.                 break;
12.             }
13.         }
14.     }

```

判断完毕后，需要输出提示信息到界面，并需要计算分数。如果5个字符的位置均正确，则表示游戏结束，计算并输出用户的得分；否则，将猜测的次数累加1，并提示用户所猜对的字符个数以及位置个数，游戏继续。代码如下：

```

01.     if (result[0] == level) {
02.         score = 100 * level - score * 10;
03.         System.out.println("GuessingGame>恭喜你猜对了！你的得分是：" + score);

```

```
04.         break;
05.     } else {
06.         score++;
07.         System.out.println("GuessingGame>你猜对" + result[1] + "个字符，其中"
08.     }
```

2.3 实现

系统代码实现如下：

```
01.     import java.util.Arrays;
02.     import java.util.Scanner;
03.     import org.apache.commons.lang.math.RandomUtils;
04.
05.     public class GuessingGame {
06.         public static void main(String[] args) {
07.             Scanner scanner = new Scanner(System.in);
08.             System.out.println("GuessingGame>欢迎尝试猜字母游戏!");
09.
10.             int level = 5;
11.             int score = 0;
12.             char[] chs = new char[level];
13.             //产生5个长度的字符数组
14.             generate(chs);
15.             System.out.println("GuessingGame>游戏开始，请输入你所猜的5个字母序列");
16.             while (true) {
17.                 String inputStr = scanner.next().trim().toUpperCase();
18.                 if ("EXIT".equals(inputStr)) {
19.                     System.out.println("GuessingGame>谢谢你的尝试，再见!");
20.                     break;
21.                 }
22.                 //比较
23.                 char[] input = inputStr.toCharArray();
24.                 int[] result = check(chs, input);
25.                 if (result[0] == level) {
26.                     score = 100 * level - score * 10;
27.                     System.out.println("GuessingGame>恭喜你猜对了！你的得分是：");
28.                     break;
```

```
29.         } else {
30.             System.out.println("GuessingGame>你猜对" + result[1] + "个
31.             score++;
32.         }
33.     }
34.
35.     scanner.close();
36. }
37.
38. public static void generate(char[] chs) {
39.     char[] letters = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', '
40.     boolean[] flags = new boolean[letters.length];
41.     for (int i = 0; i < chs.length; i++) {
42.         int index;
43.         do {
44.             index = RandomUtils.nextInt(letters.length);
45.         } while (flags[index]);
46.         chs[i] = letters[index];
47.         flags[index] = true;
48.     }
49. }
50.
51. public static int[] check(char[] chs, char[] input) {
52.     int[] result = new int[2];
53.     // result[0]: 存储正确的位置个数
54.     // result[1]: 存储正确的字符个数
55.     for (int i = 0; i < input.length; i++) {
56.         for (int j = 0; j < chs.length; j++) {
57.             if (input[i] == chs[j]) {
58.                 result[1]++;
59.                 if (i == j) {
60.                     result[0]++;
61.                 }
62.                 break;
63.             }
64.         }
65.     }
66.     return result;
67. }
```

68. }

隐藏

2.4 扩展

为猜字母游戏添加游戏等级。游戏等级设为3等：5、7和9，代表所需要猜测的字母个数。游戏开始时，由用户选择游戏等级（5，7，9）。如果选择9，则会随机产生9个字符，然后用户输入一个字符串包含9个字符，看这9个字符和随机产生的9个字符比较，看是否正确，并统计分数。系统交互情况如图-49所示：

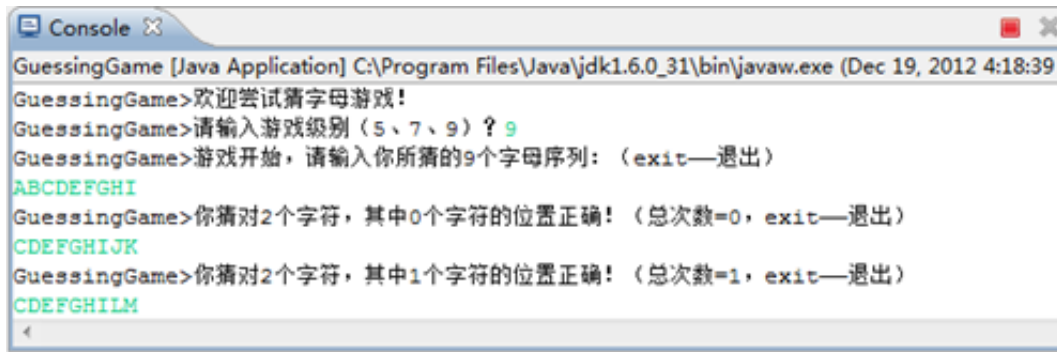


图-49

系统代码实现如下：

```
01. import java.util. Arrays;
02. import java.util. Scanner;
03. import org. apache. commons. lang. math. RandomUtils;
04.
05. public class GuessingGame {
06.     public static void main(String[] args) {
07.         Scanner scanner = new Scanner(System.in);
08.         System.out.println("GuessingGame>欢迎尝试猜字母游戏!");
09.         String levelStr;
10.         do {
11.             System.out.print("GuessingGame>请输入游戏级别（5、7、9）? ");
12.             levelStr = scanner.next().trim();
13.         } while (!"5".equals(levelStr) && !"7".equals(levelStr)
14.             && !"9".equals(levelStr));
15.
16.         int level = Integer.parseInt(levelStr);
17.
18.         int score = 0;
19.
20.         char[] chs = new char[level];
```



```

21.         generate(chs);
22.         System.out.println("GuessingGame>游戏开始, 请输入你所猜的" + level
23.             + "个字母序列: (exit——退出)");
24.         while (true) {
25.             String inputStr = scanner.next().trim().toUpperCase();
26.             if ("EXIT".equals(inputStr)) {
27.                 System.out.println("GuessingGame>谢谢你的尝试, 再见!");
28.                 break;
29.             }
30.
31.             char[] input = inputStr.toCharArray();
32.             int[] result = check(chs, input);
33.             if (result[0] == level) {
34.                 score = 100 * level - score * 10;
35.                 System.out.println("GuessingGame>恭喜你猜对了! 你的得分是:
36.                     break;
37.             } else {
38.                 System.out.println("GuessingGame>你猜对" + result[1] + "个
39.                     + result[0] + "个字符的位置正确! (总次数=" + score
40.                     score++;
41.             }
42.         }
43.
44.         scanner.close();
45.     }
46.
47.     public static void generate(char[] chs) {
48.         char[] letters = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' };
49.         boolean[] flags = new boolean[letters.length];
50.         for (int i = 0; i < chs.length; i++) {
51.             int index;
52.             do {
53.                 index = RandomUtils.nextInt(letters.length);
54.             } while (flags[index]);
55.             chs[i] = letters[index];
56.             flags[index] = true;
57.         }
58.     }
59.

```

```

60.     public static int[] check(char[] chs, char[] input) {
61.         int[] result = new int[2];
62.         // result[0]: 存储正确的位置个数
63.         // result[1]: 存储正确的字符个数
64.         for (int i = 0; i < input.length; i++) {
65.             for (int j = 0; j < chs.length; j++) {
66.                 if (input[i] == chs[j]) {
67.                     result[1]++;
68.                     if (i == j) {
69.                         result[0]++;
70.                     }
71.                     break;
72.                 }
73.             }
74.         }
75.         return result;
76.     }
77. }

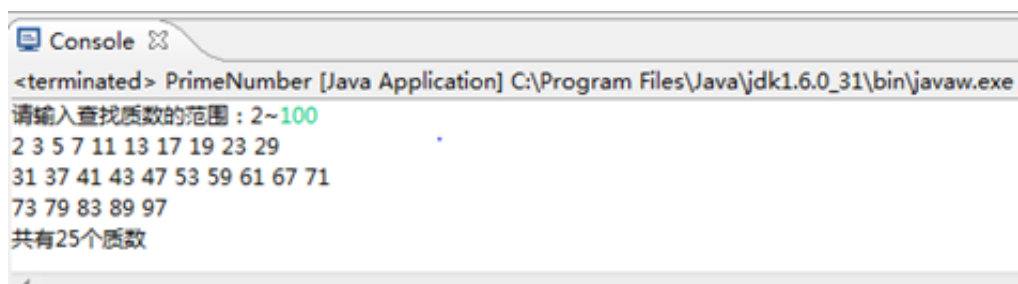
```

隐藏

3 质数问题（选做）

3.1 问题

求某个范围内所有质数和质数的个数。用户输入2到某个范围的，系统输出这个范围内的所有质数和质数的个数，质数是指只有1和它本身两个约数的数。控制台输出情况如图-38所示：



```

Console
<terminated> PrimeNumber [Java Application] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe
请输入查找质数的范围：2~100
2 3 5 7 11 13 17 19 23 29
31 37 41 43 47 53 59 61 67 71
73 79 83 89 97
共有25个质数

```

图-38

3.2 方案

系统使用双层for循环判断某个数是否为水质数，外层循环确保每个用户输入的范围内的数都判断到，代码如下所示：

```

01.     for (int n = 2; n <= max; n++) {}

```

里层循环判断某个数是否为质数。判断数n是否为质数，让n除以2到n的平方根之间的每一个数，如果n能被2到n(n大于2)的平方根之间的某个数整除，则说明n不是素数，否则n一定是素数。如果一个数不是素数，它一定能找到两个数化成乘积，如果其中一个大于平方根，那么必定能找到一个小于平方根的数和它成对，例如81的平方根是9，那只需要从2试到9就可以了，因为超过平方根的结果会比平方根小。代码如下所示：

```
01.     int m = (int) Math.sqrt(n);
02.     int i = 2;
03.     for (; i <= m; i++) {
04.         if (n % i == 0)
05.             break;
06.     }
07.     if (i > m) {
08.         System.out.print(n + " ");
09.         if (++numOfPrime % 10 == 0)
10.             System.out.println();
11.     }
```

3.3 实现

系统代码实现如下：

```
01.     import java.util.Scanner;
02.     public class PrimeNumber {
03.         public static void main(String[] args) {
04.             Scanner scanner = new Scanner(System.in);
05.             System.out.print("请输入查找质数的范围：2~");
06.             int max = scanner.nextInt();
07.             scanner.close();
08.             int numOfPrime = 0;
09.             for (int n = 2; n <= max; n++) {
10.                 // 判断n是否是质数
11.                 int m = (int) Math.sqrt(n);
12.                 int i = 2;
13.                 for (; i <= m; i++) {
14.                     if (n % i == 0)
15.                         break;
16.                 }
```

```

17.         if (i > m) {
18.             System.out.print(n + " ");
19.             if (++numOfPrime % 10 == 0)
20.                 System.out.println();
21.         }
22.     }
23.     System.out.println("\n共有" + numOfPrime + "个质数");
24. }
25. }

```

[隐藏](#)

3.4 扩展

求两个整数的最大公约数。由用户输入2个正整数，系统求出这两个正整数的最大公约数，并输出。

注：最大公约数，指某几个整数共有因子中最大的一个。如果一个数既是数A的约数，又是数B的约数，称为A,B的公约数；而A,B的公约数中最大的一个（可以包括AB自身）称为AB的最大公约数。

系统交互过程如图-39所示：

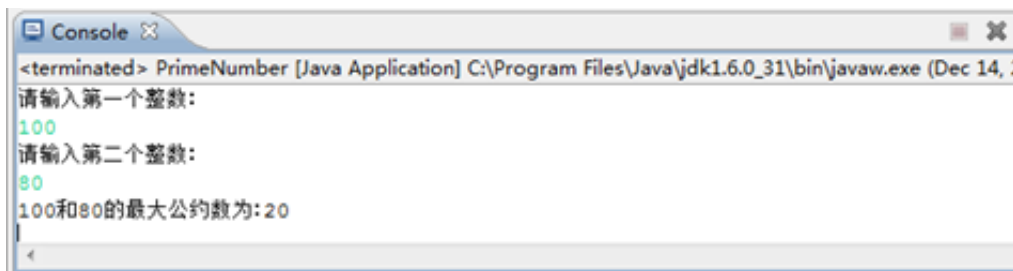


图-39

系统代码实现如下：

```

01.     import java.util.Scanner;
02.     public class CommonDivisor {
03.         public static void main(String[] args) {
04.             Scanner scanner = new Scanner(System.in);
05.             System.out.println("请输入第一个整数:");
06.             int first = scanner.nextInt();
07.             System.out.println("请输入第二个整数:");
08.             int second = scanner.nextInt();
09.             scanner.close();
10.
11.             // 从两个整数中较小的数开始
12.             int start = first > second ? second : first;

```

```

13.         int n = 1;
14.         for (n = start; n >= 1; n--) {
15.             // 判断n是否是两个整数的约数
16.             if ((first % n == 0) && (second % n == 0))
17.                 break;
18.         }
19.         System.out.println(first + "和" + second
20.                             + "的最大公约数为：" + n);
21.     }
22. }

```

[隐藏](#)

4 随机加法运算器（选做）

4.1 问题

随机加法运算器。系统提示用户输入“Y”或是“Q”，输入Y表示开始测试，输入Q表示退出系统；系统随机出题，最多10道题，题目为两个100以内的随机数的和，用户输入这两个数的和，系统根据用户输入的和值，系统判断用户计算的是否正确，正确输出“Correct!”，错误系统输出“Error”，最后计算用户做题的得分输出到控制台。系统交互情况如图-40所示：

```

Console
<terminated> AdditionTest [Java Application] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe
输入Y开始测验，输入Q退出: Y
(1). 46 + 94 = ?140
Correct!
(2). 34 + 80 = ?114
Correct!
(3). 88 + 69 = ?185
Error
(4). 83 + 89 = ?172
Correct!
(5). 24 + 81 = ?105
Correct!
(6). 77 + 72 = ?149
Correct!
(7). 54 + 66 = ?120
Correct!
(8). 73 + 16 = ?90
Error
(9). 97 + 66 = ?163
Correct!
(10). 11 + 27 = ?38
Correct!
此次测验结束，你的分数是：80
输入Y开始测验，输入Q退出: Q

```

图-40

4.2 方案

系统使用while (true) 使程序一直循环，当遇到break程序规程循环，代码如下：

```

01.     while (true) {}

```

系统使用String类的trim () 方法，去除用户输入字符串两端的空白，使用String类的equalsIgnoreCase () 方法，比较两个字符串内容是否相等，忽略大小写。代码如下所示：

```
01.    String c = scanner.next();
02.    if ("Q".equalsIgnoreCase(c.trim())) {
03.        break;
04.    }
05.    if (!"Y".equalsIgnoreCase(c.trim())) {
06.        continue;
07.    }
```

系统使用Random的nextInt () 方法生成100以内的随机数，代码如下所示：

```
01.    Random ran = new Random();
02.    a = ran.nextInt(100);
```

系统使用for循环产生10道题目,代码如下所示：

```
01.    for (int i = 1; i <= numOfQuestion; i++) {}
```

4.3 实现

系统代码实现如下：

```
01.    import java.util.Scanner;
02.    import java.util.Random;
03.    public class AdditionTest {
04.        public static void main(String[] args) {
05.            Scanner scanner = new Scanner(System.in);
06.            while (true) {
07.                System.out.print("输入Y开始测验，输入Q退出：");
08.                String c = scanner.next();
09.                if ("Q".equalsIgnoreCase(c.trim())) {
10.                    break;
11.                }
```

```

12.         if (!"Y".equalsIgnoreCase(c.trim())) {
13.             continue;
14.         }
15.         int score = 0;
16.         int numOfQuestion = 10;
17.         Random ran = new Random();
18.         int a, b, answer, key;
19.         for (int i = 1; i <= numOfQuestion; i++) {
20.             a = ran.nextInt(100);
21.             b = ran.nextInt(100);
22.             key = a + b;
23.             System.out.print("(" + i + "). " + a + " + " + b + " = ?");
24.             answer = scanner.nextInt();
25.             if (answer == key) {
26.                 score += 100 / numOfQuestion;
27.                 System.out.println("Correct!");
28.             } else {
29.                 System.out.println("Error");
30.             }
31.         }
32.         System.out.println("此次测验结束，你的分数是：" + score);
33.     }
34.     scanner.close();
35. }
36. }

```

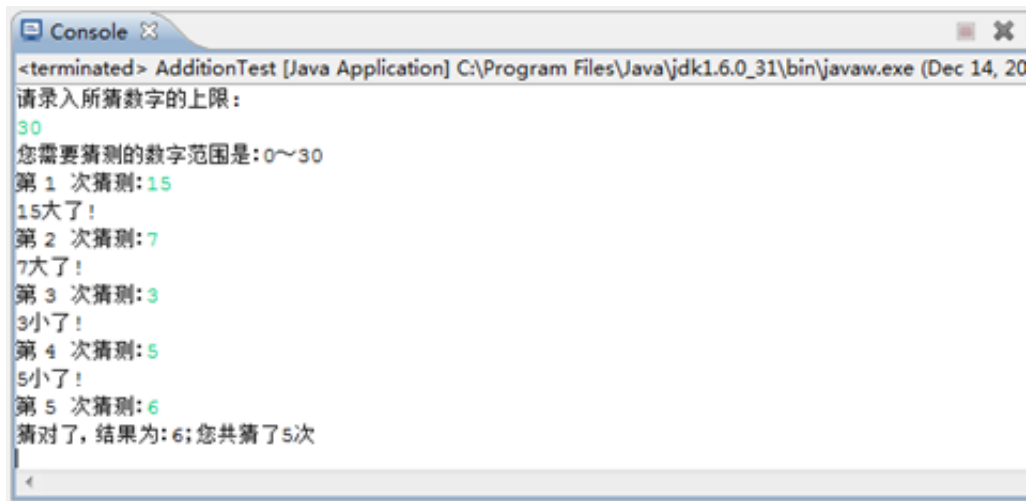
隐藏

4.4 扩展

猜数字游戏。

用户录入一个大于1的正整数，然后由系统随机抽取一个从1到此整数范围内的数值作为需要猜测的数字，然后提示用户猜测此数值。用户录入某数值后，系统判断是否猜测正确：如果录入的数值比答案数值大，则输出“大了”，并提示用户继续猜；如果录入的数值比答案数值小，则输出“小了”，并提示用户继续猜。直到录入的数值等于答案，则输出所猜中的数字，并结束游戏。

系统交互情况如图-41所示：



```
<terminated> AdditionTest [Java Application] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe (Dec 14, 20
请输入所猜数字的上限:
30
您需要猜测的数字范围是:0~30
第 1 次猜测:15
15大了!
第 2 次猜测:7
7大了!
第 3 次猜测:3
3小了!
第 4 次猜测:5
5小了!
第 5 次猜测:6
猜对了, 结果为:6;您共猜了5次
```

图-41

系统代码实现如下：

```
01.     import java.util.Scanner;
02.     import java.util.Random;
03.     public class GuessingTest {
04.         public static void main(String[] args) {
05.             Scanner scanner = new Scanner(System.in);
06.             System.out.println("请输入所猜数字的上限:");
07.             int max = scanner.nextInt();
08.             System.out.println("您需要猜测的数字范围是: 0~" + max);
09.
10.             //统计猜测的次数
11.             int count = 0;
12.             //内置结果
13.             Random ran = new Random();
14.             int result = ran.nextInt(max + 1);
15.
16.             //猜测开始
17.             while (true) {
18.                 count++;
19.                 System.out.print("第 " + count + " 次猜测:");
20.                 int number = scanner.nextInt();
21.                 if (number == result) {
22.                     System.out.println("猜对了, 结果为: "
23.                         + result + "; 您共猜了" + count + "次");
24.                 }
25.
```



```
26.         if ( number > result ) {
27.             System.out.println( number + "大了!");
28.             continue;
29.         }
30.         if ( number < result ) {
31.             System.out.println( number + "小了!");
32.             continue;
33.         }
34.     }
35.     scanner.close();
36. }
37. }
```

[隐藏](#)

5 成绩统计程序

5.1 问题

成绩管理系统。有如下功能共用户选则：录入成绩，成绩列表，查询成绩，退出。当用户在控制台输入1，用户选择的功能为录入成绩；输入2，用户选择的功能为成绩列表；输入3，用户选择的功能为根据姓名查找成绩；输入4，退出。系统交互情况如图-42所示：

```
Console X
<terminated> ScoreManager [Java Application] C:\Program Files\Java\jre6\bin\javaw.exe
*****欢迎进入成绩管理系统*****
请选择功能：1—成绩录入，2—成绩列表，3—成绩查询，4—退出1
开始录入成绩：
1. 学生姓名：Eric，成绩：?87
2. 学生姓名：Samuel，成绩：?76
3. 学生姓名：Edgar，成绩：?65
4. 学生姓名：Randy，成绩：?98
5. 学生姓名：Kenneth，成绩：?88
6. 学生姓名：Andy，成绩：?94
7. 学生姓名：Mike，成绩：?64
8. 学生姓名：Steven，成绩：?78
9. 学生姓名：Adam，成绩：?85
10. 学生姓名：Ken，成绩：?100
录入完毕。
请选择功能：1—成绩录入，2—成绩列表，3—成绩查询，4—退出2
编号      姓名      成绩
-----
1         Eric      87
2         Samuel    76
3         Edgar     65
4         Randy     98
5         Kenneth   88
6         Andy      94
7         Mike      64
8         Steven    78
9         Adam      85
10        Ken      100
-----
平均成绩：83
请选择功能：1—成绩录入，2—成绩列表，3—成绩查询，4—退出3
请输入您要查询的学生姓名：Ken
10. 学生姓名：Ken，成绩：100
请选择功能：1—成绩录入，2—成绩列表，3—成绩查询，4—退出4
*****谢谢使用*****
<
```

图-42

5.2 方案

系统使用一个整形数组和一个字符串数组分别存放学生成绩信息和姓名。其中 NameUtils.randomNames()方法能获取一个大小为numOfStudents的字符串数组，且数组的元素为学员的姓名，代码如下所示：

```
01.    int numOfStudents = 10;
02.    String[] students = NameUtils.randomNames(numOfStudents);
03.    int[] scores = new int[numOfStudents];
```

录入成绩通过循环遍历数组给数组的每个元素初始化。代码如下所示：

```
01.    System.out.println("开始录入成绩：");
```

```

02.     for (int i = 0; i < students.length; i++) {
03.         System.out.print((i + 1) + ". 学生姓名: " + students[i] + ", 成绩: ?")
04.         scores[i] = scanner.nextInt();
05.     }
06.     System.out.println("录入完毕。");

```

成绩列表功能通过遍历两个数组，打印所有信息，`StringUtils.rightPad()`方法是一个格式化输出的方法，代码如下所示：

```

01.     for (int i = 0; i < students.length; i++) {
02.         System.out.println(StringUtils.rightPad(i + 1 + "", 10, " ")
03.             + StringUtils.rightPad(students[i], 10, " ")
04.             + StringUtils.rightPad(scores[i] + "", 10, " "));
05.     }

```

查询成绩功能，首先根据姓名在姓名数组中查出该姓名所对应的下标，该下标就是成绩数组中该学员成绩所对应的下标，代码如下所示：

```

01.     int i = 0;
02.     for (; i < students.length; i++) {
03.         if (student.equalsIgnoreCase(students[i])) {
04.             break;
05.         }
06.     }
07.     if (i < students.length) {
08.         System.out.println((i + 1) + ". 学生姓名: " + students[i]
09.             + ", 成绩: " + scores[i]);
10.     } else {
11.         System.out.println("对不起，找不到学员的信息。");
12.     }

```

5.3 实现

系统代码实现如下：

```
01. import java.util.Scanner;
02. import org.apache.commons.lang.StringUtils;
03. import tts.fundamental.utils.NameUtils;
04. public class ScoreManager {
05.     public static void main(String[] args) {
06.         int numOfStudents = 10;
07.         String[] students = NameUtils.randomNames(numOfStudents);
08.         int[] scores = new int[numOfStudents];
09.         Scanner scanner = new Scanner(System.in);
10.         System.out.println("*****欢迎进入成绩管理系统*****");
11.         while (true) {
12.             System.out.print("请选择功能: 1——成绩录入, 2——成绩列表, 3——成绩查询,
13.             String c = scanner.next().trim();
14.             if ("1".equals(c)) {
15.                 System.out.println("开始录入成绩: ");
16.                 for (int i = 0; i < students.length; i++) {
17.                     System.out.print((i + 1) + ". 学生姓名: "
18. + students[i] + ", 成绩: ?");
19.                     scores[i] = scanner.nextInt();
20.                 }
21.                 System.out.println("录入完毕。");
22.             } else if ("2".equals(c)) {
23.                 int avg = 0;
24.                 for (int i = 0; i < scores.length; i++) {
25.                     avg += scores[i];
26.                 }
27.                 avg /= scores.length;
28.                 System.out.println(StringUtils.rightPad("编号", 20, " ")
29. + StringUtils.rightPad("姓名", 20, " ")
30. + StringUtils.rightPad("成绩", 20, " "));
31.                 System.out.println(StringUtils.repeat("-", 30));
32.                 for (int i = 0; i < students.length; i++) {
33.                     System.out.println(StringUtils.rightPad(i + 1 + "", 10, " ")
34. + StringUtils.rightPad(students[i], 10, " ")
35. + StringUtils.rightPad(scores[i] + "", 10, " "));
36.                 }
37.                 System.out.println(StringUtils.repeat("-", 30));
38.                 System.out.println("平均成绩: " + avg);
```

```

39.     } else if ("3".equals(c)) {
40.         System.out.print("请输入您要查询的学生姓名: ");
41.         String student = scanner.next().trim();
42.         int i = 0;
43.         for (; i < students.length; i++) {
44.             if (student.equalsIgnoreCase(students[i])) {
45.                 break;
46.             }
47.         }
48.         if (i < students.length) {
49.             System.out.println((i + 1) + ". 学生姓名: " + students[i]
50.                                 + ", 成绩: " + scores[i]);
51.         } else {
52.             System.out.println("对不起, 找不到学员的信息。");
53.         }
54.     } else if ("4".equals(c)) {
55.         System.out.println("*****谢谢使用*****");
56.         break;
57.     }
58. }
59. scanner.close();
60. }
61. }

```

隐藏

5.4 扩展

为成绩管理系统添加统计功能, 即: 1为录入成绩; 2为输出成绩列表; 3为查找成绩; 4为统计成绩; 5为退出。

其中, 功能1由用户录入成绩; 功能2则输出成绩的列表; 功能3为查询成绩; 功能4为统计, 即统计分数分别为不及格(0 - 59)、及格(60 - 79)、良好(80 - 89)、优秀(90 - 99)和满分(100)的人数; 功能5为退出。系统交互情况如图-43所示:



```
<terminated> ScoreManager [Java Application] C:\Program Files\Java\jdk1.6.0_31\bin\javaw.exe (Dec 17, 2017)
*****欢迎进入成绩管理系统*****
请选择功能: 1—成绩录入, 2—成绩列表, 3—成绩查询, 4—成绩统计, 5—退出
1
开始录入成绩:
1. 学生姓名: Stuart, 成绩: ?23
2. 学生姓名: Patrick, 成绩: ?67
3. 学生姓名: Laurent, 成绩: ?79
4. 学生姓名: Daniel, 成绩: ?89
5. 学生姓名: Oliver, 成绩: ?99
6. 学生姓名: Garfield, 成绩: ?69
7. 学生姓名: Tommy, 成绩: ?56
8. 学生姓名: Bill, 成绩: ?76
9. 学生姓名: Angus, 成绩: ?100
10. 学生姓名: Alva, 成绩: ?70
录入完毕。
请选择功能: 1—成绩录入, 2—成绩列表, 3—成绩查询, 4—成绩统计, 5—退出
4
满分(100): 1人
优秀(99-90): 1人
良好(89-80): 1人
及格(79-60): 5人
不及格(59-0): 2人
请选择功能: 1—成绩录入, 2—成绩列表, 3—成绩查询, 4—成绩统计, 5—退出
5
*****谢谢使用*****
```

图-43

系统代码实现如下：

```
01. import java.util.Scanner;
02. import org.apache.commons.lang.StringUtils;
03. import tts.fundamental.utils.NameUtils;
04. public class ScoreManager {
05.     public static void main(String[] args) {
06.         int numOfStudents = 10;
07.         String[] students = NameUtils.randomNames(numOfStudents);
08.         int[] scores = new int[numOfStudents];
09.
10.         Scanner scanner = new Scanner(System.in);
11.
12.         System.out.println("*****欢迎进入成绩管理系统*****");
13.         while (true) {
14.             System.out.println("请选择功能: 1—成绩录入, 2—成绩列表, 3—成绩查询, 4—成绩统计, 5—退出");
15.
16.             String c = scanner.next().trim();
17.             if ("1".equals(c)) {
18.                 System.out.println("开始录入成绩: ");
```

```
19.         for (int i = 0; i < students.length; i++) {
20.             System.out.print((i + 1) + ". 学生姓名: " + students[i]
21.                 + ", 成绩: ?");
22.             scores[i] = scanner.nextInt();
23.         }
24.         System.out.println("录入完毕。");
25.     } else if ("2".equals(c)) {
26.
27.         int avg = 0;
28.         for (int i = 0; i < scores.length; i++) {
29.             avg += scores[i];
30.         }
31.         avg /= scores.length;
32.
33.         System.out.println(StringUtils.rightPad("编号", 20, " ")
34.             + StringUtils.rightPad("姓名", 20, " ")
35.             + StringUtils.rightPad("成绩", 20, " "));
36.         System.out.println(StringUtils.repeat("-", 30));
37.         for (int i = 0; i < students.length; i++) {
38.             System.out.println(StringUtils
39.                 .rightPad(i + 1 + "", 10, " ")
40.                 + StringUtils.rightPad(students[i], 10, " ")
41.                 + StringUtils.rightPad(scores[i] + "", 10, " ")
42.             );
43.             System.out.println(StringUtils.repeat("-", 30));
44.             System.out.println("平均成绩: " + avg);
45.
46.     } else if ("3".equals(c)) {
47.         System.out.print("请输入您要查询的学生姓名: ");
48.         String student = scanner.next().trim();
49.         int i = 0;
50.         for (; i < students.length; i++) {
51.             if (student.equalsIgnoreCase(students[i])) {
52.                 break;
53.             }
54.         }
55.         if (i < students.length) {
56.             System.out.println((i + 1) + ". 学生姓名: " + students
57.                 + ", 成绩: " + scores[i]);
```

```
58.         } else {
59.             System.out.println("对不起，找不到学员的信息。");
60.         }
61.     } else if ("4".equals(c)) {
62.         // 统计不及格（0—59）、及格（60—79）、良好（80—89）、优秀
63.         int failed = 0;
64.         int pass = 0;
65.         int fine = 0;
66.         int excellent = 0;
67.         int full = 0;
68.
69.         for (int i = 0; i < scores.length; i++) {
70.             if (scores[i] < 60)
71.                 failed++;
72.             else if (scores[i] >= 60 && scores[i] <= 79)
73.                 pass++;
74.             else if (scores[i] >= 80 && scores[i] <= 89)
75.                 fine++;
76.             else if (scores[i] >= 90 && scores[i] <= 99)
77.                 excellent++;
78.             else if (scores[i] == 100)
79.                 full++;
80.         }
81.
82.         // 输出
83.         System.out.println("满分(100): " + full + "人");
84.         System.out.println("优秀(99-90): " + excellent + "人");
85.         System.out.println("良好(89-80): " + fine + "人");
86.         System.out.println("及格(79-60): " + pass + "人");
87.         System.out.println("不及格(59-0): " + failed + "人");
88.
89.     } else if ("5".equals(c)) {
90.         System.out.println("*****谢谢使用*****");
91.         break;
92.     }
93. }
94. scanner.close();
95. }
96. }
```


6 费氏数列问题

6.1 问题

费氏数列的全称为费波那契数列，指的是这样一个数列：1、1、2、3、5、8、13、21、34.....。这个数列的第一项和第二项均为1，从第三项开始，每一项都等于前两项之和。

现需要程序解决费氏数列的计算问题。即由用户输入需要计算的费氏序列的项数（项数从1开始），然后程序算出费氏序列中该项的数值，并打印到控制台。

系统交互情况如图-46所示：

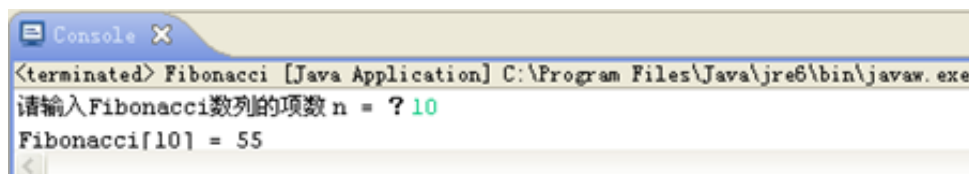


图-46

6.2 方案

首先找出数列的规律，第一项、第二项值都是1，从第三项开始，每一项的值都等于该项前两项的和。代码如下

```
01.    public static long fibonacci(int n) {  
02.        if (n == 1 || n == 2)  
03.            return 1;  
04.        long f1 = 1;  
05.        long f2 = 1;  
06.        for (int i = 3; i <= n; i++) {  
07.            f2 = f1 + f2;  
08.            f1 = f2 - f1;  
09.        }  
10.        return f2;  
11.    }
```

6.3 实现

系统代码实现如下：

```
01.    import java.util.Scanner;  
02.    public class Fibonacci {  
03.        public static void main(String[] args) {
```

```

04.         Scanner scanner = new Scanner(System.in);
05.         System.out.print("请输入Fibonacci数列的项数 n = ? ");
06.         int n = scanner.nextInt();
07.         scanner.close();
08.         long result = fibonacci(n);
09.         if (result > 0) {
10.             System.out.println("Fibonacci[" + n + "] = " + result);
11.         } else {
12.             System.out.println("Fibonacci[" + n + "] 数值太大，溢出！");
13.         }
14.
15.     }
16.     public static long fibonacci(int n) {
17.         if (n == 1 || n == 2)
18.             return 1;
19.         long f1 = 1;
20.         long f2 = 1;
21.         for (int i = 3; i <= n; i++) {
22.             f2 = f1 + f2;
23.             f1 = f2 - f1;
24.         }
25.         return f2;
26.     }
27. }

```

[隐藏](#)

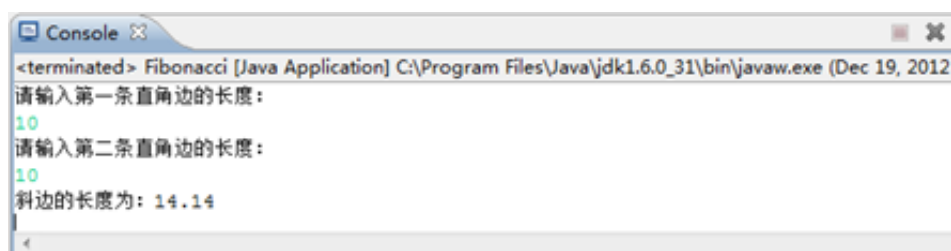
6.4 扩展

根据勾股定理计算直角三角形的斜边的边长。勾股定理：在任何一个直角三角形中，两条直角边的平方之和一定等于斜边的平方。程序要求如下：

- 1.定义方法，该方法需要传入直角三角形的两条直角边的长度，然后计算并返回其斜边的长度；

- 2.在main 方法中，先由用户在界面上录入两个长度以作为直角三角形两条直角边的长度，然后调用上一步中所定义的方法计算斜边的长度，并保留小数点后2位，在控制台输出结果。

系统交互情况如图-47所示：



系统代码实现如下：

```
01. import java.util.Scanner;
02. public class Hypotenuse {
03.     public static void main(String[] args) {
04.         Scanner scanner = new Scanner(System.in);
05.         System.out.println("请输入第一条直角边的长度：");
06.         double first = scanner.nextDouble();
07.         System.out.println("请输入第二条直角边的长度：");
08.         double second = scanner.nextDouble();
09.         scanner.close();
10.
11.         double hypotenuse = getHypotenuse(first, second);
12.         hypotenuse = Math.round(hypotenuse * 100) / 100.0;
13.         System.out.println("斜边的长度为：" + hypotenuse);
14.     }
15.     /**
16.      * 计算直角三角形的斜边的长度
17.      *
18.      * @param a
19.      * @param b
20.      * @return
21.      */
22.     public static double getHypotenuse(double a, double b) {
23.         double sum = Math.pow(a, 2) + Math.pow(b, 2);
24.         return Math.sqrt(sum);
25.     }
26. }
```