นายณัฐภัทร คำมูล

6008111006

14. Java Coding : 1100 line

```
1.  class J0100 {
2.  public static void main(String args[]) {
3.  System.out.println(args.length);
4.  System.out.println(args[0]);   // abc
5.  }
6.  }
7.  class J0101 {
8.  public static void main(String args[]) {
9.  boolean b = true;
10. System.out.println("boolean = "+b);
11. char y;
12. y = 'a';
13. System.out.println("character = "+y);
14. byte c;
15. c = 127;
16. System.out.println("byte = "+c);
17. short a;
18. a = 32767;
19. System.out.println("Short = "+a);
20. int x;
21. x = 2147483647;
22. System.out.println("Integer = "+x);
23. long b;
24. b = 9223372036854775807L;
25. System.out.println("long = "+b);
26. }
```

```
27. }
28. class J0102 {
29. public static void main(String args[]) {
30. float d;
31. d = 340000000000000000000000000000000000000f;
32. System.out.println("float = "+d);
33. double e;
34. e = 179000000000000000000000000000000000000000d;
35. System.out.println("double = "+e);
36. String z ="ThaiAll";
37. System.out.println("string = "+z);
38. System.out.println(z.substring(0,4));  // Thai
39. System.out.println(z.substring(2,5));  // aiA
40. System.out.println(z.substring(4));    // All
41. System.out.println(z.toUpperCase());   // THAIALL
42. System.out.println(z.toLowerCase());   // thaiall
43. char ar[] = new char[128];
44. ar = z.toCharArray();
45. System.out.println((char)ar[0]);       // T
46. System.out.println(ar[0]);             // T
47. System.out.println(ar[2] + ar[4]);     // 162 (97 + 65)
48. z = "1234.1";
49. int m = Integer.parseInt(z.substring(0,3)) + 5;  // 123 + 5
50. double n = Double.parseDouble(z) + 0.2;          // 1234.3
51. System.out.println(m + n);           // 128 + 1234.3 = 1362.3
52. System.out.println(Integer.toString(m) + 5);      // 1285
53. }
54. }
55. class J0201 {
56. public static void main(String args[]) {
57. int x;
58. x = 6;
```

```
59. if (x > 5) System.out.println("more than 5:" + x);
60. if (x > 5 && x < 10) System.out.println("five to ten");
61. if (x > 5 || x < 10) System.out.println("all numbers");
62. if (x > 10) {
63. System.out.print("more than 10:");
64. System.out.println(x);
65. }
66. }
67. }
68. import java.lang.*;
69. class J0202 {
70. public static void main(String args[]) {
71. int x;
72. x = 6;
73. if (x > 5) System.out.println("more than 5");
74. else System.out.println("less than or equal 5");
75. if (x > 10) System.out.println("more than 10");
76. else { System.out.println("less than or equal 10"); }
77. Comparable a[] = new Comparable[5];
78. a[0] = new Integer(3);
79. a[1] = new Integer(10);
80. a[2] = "abc";
81. System.out.println(a[0] + " " + a[1] + " " + a[2]);
82. if (a[2].equals("abc")) { System.out.println("equal"); }
83. if (a[0].compareTo(a[1]) < 0) System.out.print(a[0]); // 3
84. if (a[1].compareTo(a[0]) > 0) System.out.print(a[0]+""+a[1]); // 310
85. if (a[0].compareTo(a[0]) == 0) System.out.print("equal"); // equal
86. System.out.print(a[0].compareTo(a[1])); // -1
87. }
88. }
89. import java.util.Date;
90. class J0203 {
```

```
91.  public static void main(String args[]) {
92.  byte a = (byte) (new Date().getTime() % 5);
93.  switch (a) {
94.  case 1:
95.  System.out.println("one"); break;
96.  case 2:
97.  System.out.println("two"); break;
98.  default:
99.  System.out.println("not found" + a);
100.     break;
101.     }
102.     }
103.     class J0204 {
104.     public static void main(String args[]) {
105.     System.out.println("ASCII character :: ");
106.     for (int i=0; i<256; i++) {
107.     System.out.print((char)i + " ");
108.     }
109.     String s = "thaiall";
110.     System.out.println(s + s.length());
111.     }
112.     }
113.     class J0205 {
114.     public static void main(String args[]) {
115.     System.out.println("print 1 to 10 :: ");
116.     int i;
117.     i = -5;
118.     while (i <= 5) {
119.     try {
120.     i++;
121.     System.out.println((double)5/i); //Infinity
122.     System.out.println(5/i); //catch ok
```

```
123.    }
124.    catch (ArithmeticException e) {
125.    System.out.println("may divide by zero");
126.    }
127.    }
128.    int k = 0;
129.    i = 0;
130.    while (i < 5) {
131.    System.out.print(++k);
132.    k = k + (i++);
133.    System.out.print(k--);
134.    } // 11122447711
135.    }
136.    }
137.    class J0206 {
138.    public static void main(String args[]) {
139.    System.out.println("print 1 to 10 :: ");
140.    int i;
141.    i = 1;
142.    try {
143.    do {
144.    System.out.println(i);
145.    i++;
146.    } while (i <= 10);
147.    }
148.    catch (ArrayIndexOutOfBoundsException e) {
149.    System.out.println("over index of array");
150.    }
151.    }
152.    }
153.    import java.io.*;
154.    class J0301 {
```

```
155.    public static void main(String args[]) throws IOException {
156.    char buf;
157.    buf = (char)System.in.read();
158.    System.out.println("Output is "+buf);
159.    }
160.    } import java.io.*;
161.    class J0302 {
162.    public static void main(String args[]) throws IOException {
163.    char buf1,buf2;
164.    buf1 = (char)System.in.read();
165.    buf2 = (char)System.in.read();
166.    System.out.println("Output is "+buf1+buf2);
167.    }
168.    }
169.    import java.io.*;
170.    class J0303 {
171.    public static void main(String args[]) throws IOException {
172.    System.out.println("Get until receive 0 [hidden is 13, 10]");
173.    char buf;
174.    do {
175.    buf = (char)System.in.read();
176.    System.out.println("Output is "+buf);
177.    } while (buf != '0');
178.    }
179.    }
180.    import java.io.*;
181.    class J0304 {
182.    public static void main(String args[]) throws IOException {
183.    BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
184.    String buf;
185.    int i1,i2,i3;
186.    buf = stdin.readLine();
```

```
187.    i1 = Integer.parseInt(buf);
188.    buf = stdin.readLine();
189.    i2 = Integer.parseInt(buf);
190.    i3 = i1 + i2;
191.    System.out.println("Output is "+i1+" + "+i2+" = "+i3);
192.    }
193.    }
194.    import java.io.*;
195.    class J0305 {
196.    public static void main(String args[]) throws IOException {
197.    BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
198.    String buf;
199.    int i;
200.    System.out.println("Get until receive 0");
201.    do {
202.    buf = stdin.readLine();
203.    i = Integer.parseInt(buf);
204.    System.out.println("Output is "+i);
205.    } while (i != 0);
206.    }
207.    }
208.    class J0401 {
209.    public static void main(String args[]) {
210.    sub1(); sub2(); sub1();
211.    }
212.    static void sub1() {
213.    System.out.print("x");
214.    }
215.    static void sub2() { System.out.print("y"); }
216.    }
217.    class J0402 {
218.    public static void main(String args[]) {
```

```
219.    int s = 0;
220.    s = sub(2,8,s);
221.    s = sub(7,3,s);
222.    s = sub(4,6,s);
223.    System.out.println("Sum = "+s);
224.    }
225.    public static int sub(int x, int y, int z) {
226.    int a = y + x + z;
227.    return (a + y + x + z);
228.    }
229.    }
230.    class J0403 {
231.    public static void main(String args[]) {
232.    int j = 3;
233.    System.out.println(doubleofnumber(j));
234.    }
235.    static int doubleofnumber(int i) {
236.    i = i * 2;
237.    return (i);
238.    }
239.    }
240.    class sub01 {
241.    void subx() {
242.    System.out.println("subx in sub01");
243.    }
244.    }
245.    class sub02 {
246.    void subx() {
247.    System.out.println("subx in sub02");
248.    }
249.    }
250.    class J0404 extends sub02 {
```

```
j0404() {
super.subx(); // subx in sub02
this.subx(); // subx in main
}
public static void main(String args[]) {
sub01 x = new sub01();
System.out.println("main"); // main
x.subx();      // subx in sub01
j0404 y = new j0404();
}
void subx() {
System.out.println("subx in main");
}
}
class J0501 {
public static void main(String args[]) {
int x[] = {4,18,12};
System.out.println("Amount of array = " + x.length);
for (int i = 0; i < x.length; i++) {
System.out.println("element "+i+" = "+x[i]);
}
}
}
class J0502 {
public static void main(String args[]) {
String a[][] = new String[2][3];
a[0][0] = "101";
a[0][1] = "102";
a[0][2] = "103";
int i = 0;
a[1][i++] = "tom";  // 1,0
a[1][i++] = "dang"; // 1,1
```

```
283.    a[1][i++] = "boy";  // 1,2
284.    for (i = 0; i < a[0].length; i++) {
285.    System.out.println("element of 0,"+i+" = "+a[0][i]);
286.    }
287.    for (i = 0; i < a[1].length; i++) {
288.    System.out.println("element of 1,"+i+" = "+a[1][i]);
289.    }
290.    }
291.    }
292.    import java.io.*;
293.    class J0601 {
294.    public static void main (String args[]) throws IOException {
295.    File f = new File("j0601.java");
296.    System.out.println("getName: "+f.getName());
297.    System.out.println("getPath: "+f.getPath());
298.    System.out.println("getAbsolutePath: "+f.getAbsolutePath());
299.    System.out.println("exists: "+f.exists());
300.    System.out.println("isFile: "+f.isFile());
301.    System.out.println("isDirectory: "+f.isDirectory());
302.    System.out.println("canWrite: "+f.canWrite());
303.    System.out.println("canRead: "+f.canRead());
304.    System.out.println("length: "+f.length());
305.    File file = new File("hello.txt");
306.    boolean success = file.createNewFile();
307.    File file2 = new File("hello.java");
308.    success = file.renameTo(file2);
309.    File b = new File("c:/");
310.    success = file2.renameTo(new File(b, file2.getName()));
311.    success = (new File("hello.java")).delete();
312.    System.out.println(success); // false
313.    }
314.    }
```

```
315.  import java.io.*;
316.  class J0602 {
317.  public static void main (String args[]) {
318.  File d = new File(args[0]);
319.  String n[] = d.list();
320.  for (int i = 0; i<n.length; i++) {
321.  File f = new File(args[0] + '/' + n[i]);
322.  System.out.println(i+" : "+n[i]+" Size="+f.length());
323.  }
324.  System.out.println("directory: "+d.getPath());
325.  }
326.  }
327.  import java.io.*;
328.  class J0603 {
329.  public static void main (String args[]) throws IOException {
330.  int n = 0;
331.  byte b[] = new byte[128];
332.  FileInputStream fin = new FileInputStream("j0603.java");
333.  while ((n = fin.read(b)) != -1) {
334.  for(int i=0;i<n;i++) System.out.print((char)b[i]);
335.  }
336.  System.out.println(n = fin.read(b)); // -1
337.  fin.close();
338.  }
339.  }
340.  import java.io.*;
341.  class J0604 {
342.  public static void main (String args[]) throws IOException {
343.  FileOutputStream fout = new FileOutputStream("tmp.txt");
344.  for(int i=0;i<256;i++) {
345.  fout.write(i);
346.  }
```

```
347.    fout.close();
348.    }
349.    }
350.    import java.io.*;
351.    class J0605 {
352.    public static void main (String args[]) throws IOException {
353.    FileOutputStream fout = new FileOutputStream("tmp.txt");
354.    for(int i=1;i<=10;i++) {
355.    fout.write(i+47);
356.    fout.write(13);
357.    fout.write(10);
358.    }
359.    fout.close();
360.    }
361.    }
362.    import java.io.*;
363.    class J0606 {
364.    public static void main (String args[]) throws IOException {
365.    int i = 0, n = 0;
366.    char b[] = new char[1];
367.    FileReader fin = new FileReader("tmp.txt");
368.    while ((n =  fin.read(b)) != -1) {
369.    System.out.println(i+" : "+b[0]);
370.    i = i + 1;
371.    }
372.    fin.close();
373.    }
374.    }
375.    import java.io.*;
376.    class J0607 {
377.    public static void main (String args[]) throws IOException {
378.    int i = 1, n = 0;
```

```
379.    char b[] = new char[16];
380.    FileReader fin = new FileReader("tmp.txt");
381.    while ((n =  fin.read(b)) != -1) {
382.    System.out.print((i-1)*16 + " - " + (i*16-1) + ":");
383.    System.out.print(b[0]+b[1]+b[2]+b[3]+b[4]+b[5]+b[6]+b[7]+b[8]);
384.    System.out.println(b[9]+b[10]+b[11]+b[12]+b[13]+b[14]+b[15]);
385.    i = i + 1;
386.    }
387.    fin.close();
388.    }
389.    }
390.    import java.io.*;
391.    class J0608 {
392.    public static void main (String args[]) throws IOException {
393.    int i = 1;
394.    String b;
395.    FileReader fin = new FileReader("data.txt");
396.    BufferedReader bin = new BufferedReader (fin);
397.    while ((b =  bin.readLine()) != null) {
398.    System.out.println(i + " : " +b);
399.    i = i + 1;
400.    }
401.    System.out.println(b = bin.readLine()); // null
402.    fin.close();
403.    }
404.    }
405.    import java.io.*;
406.    class J0701 {
407.    public static void main (String args[]) throws IOException {
408.    int i = 1;
409.    int tot = 0;
410.    String b;
```

```
411.  String[] fields;
412.  String patternStr = ",";
413.  FileReader fin = new FileReader("data.txt");
414.  BufferedReader bin = new BufferedReader (fin);
415.  while ((b =  bin.readLine()) != null) {
416.  fields = b.split(patternStr);
417.  System.out.println(i + " : " + fields[0]);
418.  System.out.println("Name : " + fields[1]);
419.  System.out.println("Salary : " + fields[2]);
420.  System.out.println("Status : " + fields[3]);
421.  tot = tot + Integer.parseInt(fields[2]);
422.  i = i + 1;
423.  }
424.  System.out.println("Total : " + tot);
425.  fin.close();
426.  }
427.  }
428.  import java.io.*;
429.  import java.lang.*;
430.  class J0702 {
431.  public static void main (String args[]) throws IOException {
432.  int i = 1;
433.  String b;
434.  String[] fields;
435.  String patternStr = ",";
436.  FileReader fin = new FileReader("data.txt");
437.  BufferedReader bin = new BufferedReader (fin);
438.  FileOutputStream fout = new FileOutputStream("data.htm");
439.  BufferedOutputStream bout = new BufferedOutputStream(fout);
440.  PrintStream pout = new PrintStream(bout);
441.  pout.println("<body bgcolor=yellow><table border=1 width=100%>");
442.  while ((b =  bin.readLine()) != null) {
```

```
443.    fields = b.split(patternStr);
444.    pout.println("<tr>");
445.    pout.println("<td>"+i+"</td>");
446.    pout.println("<td>"+"ID = " + fields[0]+"</td>");
447.    pout.println("<td>"+"Name = " + fields[1]+"</td>");
448.    pout.println("<td>"+"Salary = " + fields[2]+"</td>");
449.    pout.println("<td>"+"Status = " + fields[3]+"</td>");
450.    pout.println("</tr>");
451.    i = i + 1;
452.    }
453.    pout.println("</table></body>");
454.    fin.close();
455.    pout.close();
456.    }
457.    }
458.    import java.io.*;
459.    class J0703 {
460.    public static void main (String args[]) throws IOException {
461.    int i = 0,d;
462.    String b;
463.    String[] fields;
464.    String[] recs = {"","",""};
465.    String patternStr = ",";
466.    FileReader fin = new FileReader("data.txt");
467.    BufferedReader bin = new BufferedReader (fin);
468.    while ((b =  bin.readLine()) != null) {
469.    recs[i] = b;
470.    i = i + 1;
471.    }
472.    fin.close();
473.    FileOutputStream fout = new FileOutputStream("data.htm");
474.    BufferedOutputStream bout = new BufferedOutputStream(fout);
```

```
475.    PrintStream pout = new PrintStream(bout);
476.    for(int j=0;j<i;j++) {
477.    fields = recs[j].split(patternStr);
478.    pout.print(fields[0]+","+fields[1]+",");
479.    // pout.print(Double.valueOf(fields[2]).doubleValue());
480.    d = Integer.valueOf(fields[2]).intValue() + 100;
481.    pout.print(d);
482.    pout.println(","+fields[3]);
483.    }
484.    pout.close();
485.    }
486.    }
487.    import java.io.*;
488.    class J0801 {
489.    public static void main (String args[]) throws IOException {
490.    int found=0;
491.    char buf;
492.    String b,g = "";
493.    String[] fields;
494.    String patternStr = ",";
495.    System.out.println("Wait id and end character with [x]");
496.    buf = (char)System.in.read();
497.    while (buf != 'x') {
498.    g = g + buf;
499.    buf = (char)System.in.read();
500.    }
501.    FileReader fin = new FileReader("data.txt");
502.    BufferedReader bin = new BufferedReader (fin);
503.    while ((b =  bin.readLine()) != null) {
504.    fields = b.split(patternStr);
505.    if (fields[0].equals(g)) {
506.    System.out.println(fields[1]);
```

```
507.    found = 1;
508.    }
509.    }
510.    if (found == 0) System.out.println("Not found");
511.    fin.close();
512.    }
513.    }
514.    import java.io.*;
515.    class J0802 {
516.    public static void main (String args[]) throws IOException {
517.    int found=0;
518.    String b,g = "";
519.    String[] fields;
520.    System.out.println("Wait string and enter");
521.    BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
522.    g = stdin.readLine();
523.    String patternStr = g;
524.    FileReader fin = new FileReader("data.txt");
525.    BufferedReader bin = new BufferedReader (fin);
526.    while ((b =  bin.readLine()) != null) {
527.    fields = b.split(patternStr);
528.    if (fields.length > 1) {
529.    fields = b.split(",");
530.    System.out.println(fields[0] + fields[1] + fields[2] + fields[3]);
531.    found = 1;
532.    }
533.    }
534.    if (found == 0) System.out.println("Not found");
535.    fin.close();
536.    }
537.    }
538.    import java.io.*;
```

```
539.  class J0901 {
540.  public static void main (String args[]) throws IOException {
541.  int i = 0,t1,t2;
542.  String b,status;
543.  String fields[];
544.  String[] recs1 = new String[10];
545.  String[] recs2 = {"A,Active","R,Retire"};
546.  String patternStr = ",";
547.  FileReader fin = new FileReader("data.txt");
548.  BufferedReader bin = new BufferedReader (fin);
549.  while ((b =  bin.readLine()) != null) {
550.  recs1[i] = b;
551.  i = i + 1;
552.  }
553.  fin.close();
554.  t1 = i;
555.  t2 = recs2.length;
556.  for(int j=0;j<t1;j++) {
557.  fields = recs1[j].split(patternStr);
558.  System.out.print(fields[0] + fields[1] + fields[2]+fields[3]);
559.  status = fields[3];
560.  for(int k=0;k<t2;k++) {
561.  fields = recs2[k].split(patternStr);
562.  if (fields[0].equals(status)) {
563.  System.out.println(fields[1]);
564.  }
565.  }
566.  }
567.  }
568.  }
569.  import java.io.*;
570.  class J0902 {
```

```
571.  public static void main (String args[]) throws IOException {
572.  int i = 0,t1,t2;
573.  String b,status;
574.  String[] fields;
575.  String[] recs1 = {"","","","","","",""};
576.  String[] recs2 = new String[2];
577.  FileReader fin = new FileReader("data.txt");
578.  BufferedReader bin = new BufferedReader (fin);
579.  while ((b =  bin.readLine()) != null) {
580.  recs1[i] = b;
581.  i = i + 1;
582.  }
583.  fin.close();
584.  t1 = i;
585.  i = 0;
586.  FileReader fin2 = new FileReader("datas.txt");
587.  BufferedReader bin2 = new BufferedReader (fin2);
588.  while ((b =  bin2.readLine()) != null) {
589.  recs2[i] = b;
590.  i = i + 1;
591.  }
592.  fin2.close();
593.  t2 = i;
594.  for(int j=0;j<t1;j++) {
595.  fields = recs1[j].split(",");
596.  System.out.print(fields[0] + fields[1] + fields[2]+fields[3]);
597.  status = fields[3];
598.  for(int k=0;k<t2;k++) {
599.  fields = recs2[k].split(",");
600.  if (fields[0].equals(status)) {
601.  System.out.println(fields[1]);
602.  }
```

```
603.    }
604.    }
605.    }
606.    }
607.    class J1001 {
608.    public static void main (String args[]) {
609.    int tmp,x[] = {5,6,1,2,9,12,9,3};
610.    for(int i=1;i<x.length;i++) {
611.    for(int j=x.length-1;j>=i;j--) {
612.    if(x[j-1] > x[j]) {
613.    tmp = x[j];
614.    x[j] = x[j-1];
615.    x[j-1] = tmp;
616.    }
617.    }
618.    }
619.    for(int i=0;i<x.length;i++) {
620.    System.out.println(x[i]);
621.    }
622.    }
623.    }
624.    import java.lang.*;
625.    class J1002 {
626.    public static void main (String args[]) {
627.    String tmp,x[] = {"ac","abc","adb","a","aa","acd","a a","a d"};
628.    System.out.println("Before sorting");
629.    prtlist(x);
630.    for(int i=1;i<x.length;i++) {
631.    for(int j=x.length-1;j>=i;j--) {
632.    if(x[j-1].compareTo(x[j])>0) {
633.    tmp = x[j];
634.    x[j] = x[j-1];
```

```
635.    x[j-1] = tmp;
636.    }
637.    }
638.    }
639.    System.out.println("After sorting");
640.    prtlist(x);
641.    }
642.    public static void prtlist(String[] x) {
643.    for(int i=0;i<x.length;i++) {
644.    System.out.println(x[i]);
645.    }
646.    }
647.    }
648.    import java.applet.*;
649.    import java.awt.*;
650.    public class J1101 extends java.applet.Applet {
651.    public void paint(Graphics g) {
652.    g.setColor(new Color(240,240,240));
653.    g.drawString("test",10,20);
654.    }
655.    }
656.    import java.applet.*;
657.    import java.awt.*;
658.    public class J1102 extends Applet {
659.    int i,j;
660.    String istr,p;
661.    public void init() {
662.    setBackground(Color.yellow);
663.    p = getParameter("x");
664.    }
665.    public void paint(Graphics g) {
666.    g.setColor(Color.black);
```

```
667.    g.drawString(p,0,10);
668.    i = 1;
669.    while (i <= 10) {
670.    j = 10 * i;
671.    istr= Integer.toString(i);
672.    g.drawString(istr,72,j); // column = 1 inch
673.    i++;
674.    }
675.    }
676.    }
677.    import java.applet.*;
678.    import java.awt.*;
679.    public class J1103 extends Applet implements Runnable{
680.    Thread timer;
681.    int row = 10;
682.    public void paint(Graphics g) {
683.    row = row + 2;
684.    g.drawLine(5,row,30,row);
685.    }
686.    public void start() {
687.    timer = new Thread(this);
688.    timer.start(); // start clock
689.    }
690.    public void run() {
691.    Thread me = Thread.currentThread();
692.    while (timer == me) {
693.    try {
694.    Thread.currentThread().sleep(1000);
695.    } catch (InterruptedException e) { }
696.    repaint();
697.    }
698.    }
```

```
699.    }
700.    import java.applet.*;
701.    import java.awt.*;
702.    public class J1104 extends Applet {
703.    Image img;
704.    public void init() {
705.    setBackground(Color.green);
706.    img = getImage(getDocumentBase(),"x.gif");
707.    }
708.    public void paint(Graphics g) {
709.    g.setColor(Color.black);
710.    g.drawLine(5,10,30,40);
711.    g.drawRect(50,50,80,80);
712.    g.drawOval(50,50,20,30);
713.    g.setColor(Color.white);
714.    g.fillOval(50,50,20,30); // backgound is white
715.    g.setColor(Color.red);
716.    g.drawArc(40,30,55,55,0,120);
717.    int[] x={0,80,100,5,10};
718.    int[] y={0,50,80,80,30};
719.    g.drawPolygon(x,y,5);
720.    g.drawImage(img, 0, 200, this);
721.    }
722.    }
723.    import java.applet.*;
724.    import java.awt.*;
725.    import java.awt.event.*;
726.    public class J1105 extends Applet implements ActionListener {
727.    Button b1 = new Button("1");
728.    Label l1 = new Label("Hello");
729.    TextField t1 = new TextField("1");
730.    int row = 10;
```

```
731.    public void paint(Graphics g) {
732.    row = row + 10;
733.    g.drawLine(5,row,30,row);
734.    }
735.    public void init() {
736.    setBackground(Color.red);
737.    add(l1);
738.    add(b1);
739.    add(t1);
740.    t1.addActionListener(this);
741.    b1.addActionListener(this);
742.    }
743.    public void actionPerformed(ActionEvent e) {
744.    int intb1 = Integer.parseInt(e.getActionCommand());
745.    intb1 = intb1 + 1;
746.    String s = Integer.toString(intb1);
747.    l1.setText(s);
748.    b1.setLabel(s);
749.    t1.setText(s);
750.    repaint();
751.    }
752.    }
753.    import java.io.*;
754.    class J1201 {
755.    public static void main(String args[]) throws IOException {
756.    int buf=49;
757.    while (buf != 51) {
758.    if (buf >= 49 && buf <= 51) {
759.    System.out.println("What is your option?");
760.    System.out.println("1. print 1 to 10");
761.    System.out.println("2. print 'ok'");
762.    System.out.println("3. exit");
```

```
763.    }
764.    buf = System.in.read();
765.    switch (buf) {
766.    case 49: // character 1
767.    for (int i=1;i<=10;i++) {
768.    System.out.println(i);
769.    }
770.    break;
771.    case 50: // character 2
772.    System.out.println("ok");
773.    break;
774.    case 51: break; // character 3
775.    case 13: break;
776.    case 10: break;
777.    default:
778.    System.out.println("Nothing to do");
779.    break;
780.    }
781.    }
782.    System.out.println("See you again");
783.    }
784.    }
785.    import java.io.*;
786.    class J1202 {
787.    public static void main(String args[]) throws IOException {
788.    BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
789.    String buf=" ";
790.    while (!buf.equals("3")) {
791.    System.out.println("What is your option?");
792.    System.out.println("1. print 1 to 10");
793.    System.out.println("2. print 'ok'");
794.    System.out.println("3. exit");
```

```
795.    buf = stdin.readLine();
796.    if (buf.equals("1"))
797.    for (int i=1;i<=10;i++) System.out.println(i);
798.    if (buf.equals("2")) System.out.println("ok");
799.    }
800.    System.out.println("See you again");
801.    }
802.    }
803.    import java.io.*;
804.    class J1203 {
805.    public static void main(String args[]) throws IOException {
806.    BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
807.    String buf=" ";
808.    while (!buf.equals("3")) {
809.    System.out.println("What is your option?");
810.    System.out.println("1. print 1 to 10");
811.    System.out.println("2. print 'ok'");
812.    System.out.println("3. exit");
813.    buf = stdin.readLine();
814.    if (buf.equals("1")) oho1();
815.    if (buf.equals("2")) { oho2(); }
816.    }
817.    System.out.println("See you again");
818.    }
819.    public static void oho1() {
820.    for (int i=1;i<=10;i++) {
821.    System.out.println(i);
822.    }
823.    }
824.    public static void oho2() {
825.    System.out.println("ok");
826.    }
```

```
827.    }
828.    ------------------------------------------------------------
829.    import java.io.*;
830.    class Pollweb {
831.    public static void main (String args[]) throws IOException {
832.    int i=0;
833.    int questionhave = 14;
834.    int q[] = new int[questionhave];
835.    String b;
836.    String[] fields;
837.    String patternStr = ",";
838.    FileReader fin = new FileReader("pollweb.txt");
839.    BufferedReader bin = new BufferedReader (fin);
840.    while ((b =  bin.readLine()) != null) {
841.    fields = b.split(patternStr);
842.    for (int j=1;j<=questionhave-1;j++)
843.    q[j]+= Integer.parseInt(fields[j]);
844.    i = i + 1;
845.    }
846.    System.out.println("Total questions: " + i);
847.    for (int j=1;j<=questionhave-1;j++)
848.    System.out.println(j+":"+q[j]+" | "+(q[j] * 100 / i)+"%");
849.    fin.close();
850.    }
851.    }
852.    -------------------------------------
853.    class Hello1 {
854.    public static void main(String args[]) {
855.    System.out.println("hello");
856.    }
857.    }
858.    -------------------------------------
```

```
859.  import java.lang.*;
860.  import java.applet.*;
861.  import java.awt.Graphics;
862.  public class Hello2 extends java.applet.Applet {
863.  public void paint(Graphics g){
864.  g.drawString("hello",10,10);
865.  }
866.  }
867.  ------------------------------------
868.  class Pyramid01 {
869.  public static void main(String args[]) {
870.  int k = 4;
871.  for (int i=1;i<=k;i++) {
872.  for (int j=2;j<=i;j++) { System.out.print(" "); }
873.  System.out.print(i+""+i);
874.  for (int j=k;j>=(i+1);j--) { System.out.print("**"); }
875.  System.out.println(i+""+i);
876.  } } }
877.  ------------------------------------
878.  class Pyramid02 {
879.  public static void main(String args[]) {
880.  int k = 4;
881.  for (int i=1;i<=k;i++) {
882.  for (int j=i;j<=(i+2);j++) { System.out.print(j); }
883.  for (int j=1;j<=(2+i);j++) { System.out.print("*"); }
884.  System.out.println();
885.  } } }
886.  ------------------------------------
887.  class Pyramid03 {
888.  public static void main(String args[])  {
889.  int k = 4;
890.  for (int i=1;i<=k;i++) {
```

```
891.    System.out.print(i+""+(i+4));
892.    for (int j=1;j<=(4+i);j++) {
893.    System.out.print("*");
894.    }
895.    System.out.println();
896.    } } }
897.    ------------------------------------
898.    class Pyramid04 {
899.    public static void main(String args[]) {
900.    int k = 4;
901.    for (int i=1;i<=k;i++) {
902.    for (int j=1;j<=i;j++) { System.out.print("*"); }
903.    for (int j=i;j>=2;j--) { System.out.print(j); }
904.    for (int j=1;j<=i;j++) { System.out.print(j); }
905.    System.out.println();
906.    } } }
```