```
1  package Basement_Syntax;
2  import java.util.Scanner;
3  /*
4     Data Type : range analysis
5   */
6  public class RangeOfType {
7     final static byte DEC = -1, ASC = +1, DOUBLE = 2,
8           BYTE = 0, SHORT = 1, INT = 2, LONG = 3, FINISH = 4;
9     final static String[] TYPE = {"byte ", "short ", "int ", "long "};
10    private static String msg = "The range of ",
11          prompt = "Enter an integer for type, 0 for byte, 1 for short, 2 for int,
    3 for long; Otherwise, Finish";
12    private static byte trend = ASC, type = BYTE;
13    private static long count = 0;
14
15    public static void main(String[] args) {
16       Scanner option = new Scanner(System.in);
17       while(type != FINISH){
18          System.out.println(prompt);
19          type = option.nextByte();
20          if((type == BYTE)||(type == SHORT)||(type == INT)||(type == LONG))
21             msg += TYPE[type] + "is:\n";
22          else
23             type = FINISH;
24          range_1_Multiplication(type);
25          msg = "The range of "; count = 0;//Reset
26       }
27    }
28
29
30    public static void range_0_Addition(byte type){
31       switch(type){
32          case BYTE: {
33             byte max = DOUBLE -1, min = DOUBLE;
34             while(max < min) {
35                min += trend;
36                max += trend;
37                count++;
38             }
39             msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
```

```
40              }
41              case SHORT:  {
42                  short max = DOUBLE -1, min = DOUBLE;
43                  while(max < min) {
44                      min += trend;
45                      max += trend;
46                      count++;
47                  }
48                  msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
49              }
50              case INT:  {
51                  int max = DOUBLE -1, min = DOUBLE;
52                  while(max < min) {
53                      min += trend;
54                      max += trend;
55                      count++;
56                  }
57                  msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
58              }
59              case LONG:  {
60                long max = DOUBLE -1, min = DOUBLE;//
61                  while(max < min) {
62                      min += trend;
63                      max += trend;
64                      count++;
65                  }
66                  msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
67              }
68              default: msg = "Your Entering Type Is Not Defined";
69          }
70          msg += "\n Times of Execution is : " + count+"\n";
71          System.out.println(msg);
72      }
73
74
75      public static void range_O_Multipication(byte type){
76          switch(type){
77              case BYTE: {
78                  byte max = DOUBLE -1, min = DOUBLE;//
79                  while(max < min) {
```

```
80                  min *= DOUBLE;
81                  max += DOUBLE;
82                  count++;
83              }
84              max = (byte)(min - 1);
85              msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
86          }
87          case SHORT: {
88              short max = DOUBLE -1, min = DOUBLE;//
89              while(max < min) {
90                  min *= DOUBLE;
91                  max += DOUBLE;
92                  count++;
93              }
94              max = (short)(min - 1);
95              msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
96          }
97          case INT: {
98              int max = DOUBLE -1, min = DOUBLE;//
99              while(max < min) {
100                 min *= DOUBLE;
101                 max += DOUBLE;
102                 count++;
103             }
104             max = min - 1;
105             msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
106         }
107         case LONG: {
108             long max = DOUBLE -1, min = DOUBLE;//
109             while(max < min) {
110                 min *= DOUBLE;
111                 max += DOUBLE;
112                 count++;
113             }
114             max = min - 1;
115             msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
116         }
117         default: msg = "Your Entering Type Is Not Defined";
118     }
119     msg += "\n Times of Execution is " + count+"\n";
```

```
120         System.out.println(msg);
121     }
122
123
124     public static void range_1_Multiplication(byte type){
125         long max = DOUBLE - 1;
126         switch(type){
127             case BYTE : {
128                 byte min = DOUBLE - 1;
129                 while(min == max){
130                     min *= DOUBLE;
131                     max *= DOUBLE;
132                     count++;
133                 }
134                 max -= 1;
135                 msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
136             }
137             case SHORT : {
138                 short min = DOUBLE - 1;
139                 while(min == max){
140                     min *= DOUBLE;
141                     max *= DOUBLE;
142                     count++;
143                 }
144                 max -= 1;
145                 msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
146             }
147             case INT : {
148                 int min = DOUBLE - 1;
149                 while(min == max){
150                     min *= DOUBLE;
151                     max *= DOUBLE;
152                     count++;
153                 }
154                 max -= 1;
155                 msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
156             }
157             case LONG : {
158                 long min = DOUBLE;
159                 while(min > max){
```

```
160                 min *= DOUBLE;
161                 max *= DOUBLE;
162                 count++;
163             }
164             max = min - 1;
165             msg += "\tMaximum : "+max+"\n\t"+"Minimum : "+min; break;
166         }
167         default : msg = "Your Entering Type Is Not Defined";
168       }
169       msg += "\n Times of Execution is " + count+"\n";
170       System.out.println(msg);
171    }
172
173
174    public static void range_1plus_Multiplication(byte type){
175       long min = DOUBLE - 1, max = DOUBLE - 1;
176       while(min == max) {
177          max *= DOUBLE; count++;
178          switch (type) {
179             case BYTE: min = (byte) max; break;
180             case SHORT: min = (short) max; break;
181             case INT: min = (int) max; break;
182             case LONG: min = max; break;
183             default: msg = "Your Entering Type Is Not Defined";
184          }
185          System.out.println(count + " : \t" + min + "\t;\t" + max);
186          if(max > max*DOUBLE){//Execute only if the type is long
187             count++;
188             max *= DOUBLE;
189             min = max;
190             System.out.println(count + " : \t" + min + "\t;\t" + max);
191             break;
192          }
193       }
194       max--;
195       msg += "\tMaximum : "+max+ " \n\tMinimum : "+min;
196       msg += "\n Times of Execution is " + count+"\n";
197       System.out.println(msg);
198    }
199 }
```