```java
1 import components.simplereader.SimpleReader;
6
7 /**
8  * The users are asked to type a constant number and 4 numbers which are
9  * meaningful. When the list of the numbers are provided, the meaningful numbers
10 * would combine with the list of the numbers to get the constant number that
11 * are shown above.
12 *
13 * @author Yiming Cheng
14 */
15 public class ABCDGuesser2 {
16
17     /**
18      * Repeatedly asks the user for a positive real number until the user enters
19      * one. Returns the positive real number.
20      *
21      * @param in
22      *            the input stream
23      * @param out
24      *            the output stream
25      * @return a positive real number entered by the user
26      */
27     private static double getPositiveDouble(SimpleReader in, SimpleWriter out) {
28         String positive = in.nextLine();
29
30         while (!(FormatChecker.canParseDouble(positive))) {
31             out.println("Type a mathamatical constant");
32             positive = in.nextLine();
33         }
34         double positiveNumber = Double.parseDouble(positive);
35         while (positiveNumber > 0) {
36             positive = in.nextLine();
37             positiveNumber = Double.parseDouble(positive);
38         }
39
40         return positiveNumber;
41     }
42
43     /**
44      * Repeatedly asks the user for a positive real number not equal to 1.0
45      * until the user enters one. Returns the positive real number.
46      *
47      * @param in
48      *            the input stream
49      * @param out
50      *            the output stream
51      * @return a positive real number not equal to 1.0 entered by the user
52      */
53     private static double getPositiveDoubleNotOne(SimpleReader in,
54             SimpleWriter out) {
55         String real = in.nextLine();
56         while (!(FormatChecker.canParseDouble(real))) {
57             out.println("Type a numbers which is meaningful to you");
58             real = in.nextLine();
59         }
60         double realNumber = Double.parseDouble(real);
61         while (realNumber == 1.0) {
62             real = in.nextLine();
63             realNumber = Double.parseDouble(real);
64         }
65         return realNumber;
66
```

```java
 67      }
 68
 69      /**
 70       * Main method.
 71       *
 72       * @param args
 73       *                the command line arguments
 74       */
 75      public static void main(String[] args) {
 76          SimpleReader in = new SimpleReader1L();
 77          SimpleWriter out = new SimpleWriter1L();
 78          out.println("Type a mathamatical constant");
 79          double u = getPositiveDouble(in, out);
 80          /*
 81           * People would be asked to type the numbers which are meaningful to
 82           * them.
 83           */
 84          out.println("Type a numbers which is meaningful to you");
 85          double w = getPositiveDoubleNotOne(in, out);
 86          out.println("Type a numbers which is meaningful to you");
 87          double x = getPositiveDoubleNotOne(in, out);
 88          out.println("Type a numbers which is meaningful to you");
 89          double y = getPositiveDoubleNotOne(in, out);
 90          out.println("Type a numbers which is meaningful to you");
 91          double z = getPositiveDoubleNotOne(in, out);
 92          /*
 93           * The list of numbers could combined with 4 meaningful numbers.
 94           */
 95          final double[] seriesNumber = { -5, -4, -3, -2, -1, -1.0 / 2, -1.0 / 3,
 96                  -1.0 / 4, 0, 1.0 / 4, 1.0 / 3, 1.0 / 2, 1, 2, 3, 4, 5 };
 97          double estimate = 0;
 98          /*
 99           * The numbers which are used to find the closer number for the constant
100           * number
101           */
102          int a = 0;
103          int b = 0;
104          int c = 0;
105          int d = 0;
106          /*
107           * The numbers which is related to the closet number are shown.
108           */
109          double fA = 0;
110          double fB = 0;
111          double fC = 0;
112          double fD = 0;
113          final int percentage = 100;
114          /*
115           * Find the closest number to the number which the users type.
116           */
117          for (a = 0; a < (seriesNumber.length - 1); a++) {
118              double num1 = Math.pow(w, seriesNumber[a]);
119              for (b = 0; b < (seriesNumber.length - 1); b++) {
120                  double num2 = Math.pow(x, seriesNumber[b]);
121                  for (c = 0; c < (seriesNumber.length - 1); c++) {
122                      double num3 = Math.pow(y, seriesNumber[c]);
123                      for (d = 0; d < (seriesNumber.length - 1); d++) {
124                          double num4 = Math.pow(z, seriesNumber[d]);
125                          double estimateNumber = num1 * num2 * num3 * num4;
126                          while (Math.abs(u - estimate) > Math
127                                  .abs(u - estimateNumber)) {
128                              estimate = estimateNumber;
```

```java
129                                    fA = seriesNumber[a];
130                                    fB = seriesNumber[b];
131                                    fC = seriesNumber[c];
132                                    fD = seriesNumber[d];
133                                }
134                            }
135                        }
136                    }
137                }
138        out.println(fA);
139        out.println(fB);
140        out.println(fC);
141        out.println(fD);
142        out.println("The closest number would be " + estimate);
143        out.println("The relative error would be "
144                + Math.abs(percentage * (1 - estimate / u)) + "%.");
145    }
146 }
147
```