```java
import components.simplereader.SimpleReader;

/**
 * This program illustrates the charming theory.
 *
 * @author Ansh Pachauri
 *
 */
public class ABCDGuesser2 {
    /**
     * Repeatedly asks the user for a positive real number
until the user enters
     * one. Returns the positive real number.
     *
     * @param in
     *                  the input stream
     * @param out
     *                  the output stream
     * @return a positive real number entered by the user
     */
    private static double getPositiveDouble(SimpleReader in,
SimpleWriter out) {
        double num = -1.0;
        while (num <= 0) {
            out.print("enter a positive number: ");
            String input = in.nextLine();
            if (FormatChecker.canParseDouble(input)) {
                num = Double.parseDouble(input);
            } else {
                out.println("enter a positive number ");
            }
        }
        return num;
    }

    /**
     * Repeatedly asks the user for a positive real number not
equal to 1.0
     * until the user enters one. Returns the positive real
```

```java
number.
     *
     * @param in
     *            the input stream
     * @param out
     *            the output stream
     * @return a positive real number not equal to 1.0 entered by the user
     */
    private static double getPositiveDoubleNotOne(SimpleReader in,
            SimpleWriter out) {

        double num = -1.0;
        while (num <= 1) {
            out.print("enter a positive number other than 1 ");
            String input = in.nextLine();
            if (FormatChecker.canParseDouble(input)) {
                num = Double.parseDouble(input);
            } else {
                out.println("enter a positive number ");
            }
        }
        return num;

    }

    private static double getError(double mu, double w, double x, double y,
            double z, double a, double b, double c, double d) {
        int i = 0, j = 0, k = 0, l = 0;

        double[] array1 = { -5, -4, -3, -2, -1, -1.0 / 2, -1.0 / 3, -1.0 / 4, 0,
                1.0 / 4, 1.0 / 3, 1.0 / 2, 1, 2, 3, 4, 5 };

        double estimate = (Math.pow(w, a) * Math.pow(x, b) * Math.pow(y, c)
                * Math.pow(z, d));
        double error = Math.abs(estimate - mu) / mu;
        for (error = Math.abs(estimate - mu) / mu; error > 0.01;) {
            for (i = 0; i < 17; i++) {
                b = array1[i];
                estimate = (Math.pow(w, a) * Math.pow(x, b) * Math.pow(y, c)
                        * Math.pow(z, d));
                error = Math.abs(estimate - mu) / mu;
                i++;
                j = 0;
                for (j = 0; j < 17; j++) {
                    b = array1[j];
                    estimate = (Math.pow(w, a) * Math.pow(x, b) * Math.pow(y, c)
                            * Math.pow(z, d));
                    error = Math.abs(estimate - mu) / mu;
                    j++;
                    k = 0;
                    for (k = 0; k < 17; k++) {
                        c = array1[k];
                        estimate = (Math.pow(w, a) * Math.pow(x, b)
                                * Math.pow(y, c) * Math.pow(z, d));
                        error = Math.abs(estimate - mu) / mu;
                        k++;
                        l = 0;
                        for (l = 0; l < 17; l++) {
                            d = array1[l];
                            estimate = (Math.pow(w, a) * Math.pow(x, b)
                                    * Math.pow(y, c) * Math.pow(z, d));
                            error = Math.abs(estimate - mu) / mu;

                            l++;
                        }
                    }
```

```java
                }
            }
        }
        return estimate;

    }

    /**
     * main program.
     *
     * @param args
     */
    public static void main(String[] args) {
        SimpleReader in = new SimpleReader1L();
        SimpleWriter out = new SimpleWriter1L();
        out.print("For the value of µ ");
        double mu = getPositiveDouble(in, out);
        out.print("For the value of w ");
        double w = getPositiveDoubleNotOne(in, out);
        out.print("For the value of x ");
        double x = getPositiveDoubleNotOne(in, out);
        out.print("For the value of y ");
        double y = getPositiveDoubleNotOne(in, out);
        out.print("For the value of z ");
        double z = getPositiveDoubleNotOne(in, out);
        double a = 0, b = 0, c = 0, d = 0;
        double estimate = getError(mu, w, x, y, z, a, b, c, d);
        out.println("the answer is " + estimate);
        double error = (Math.abs(estimate - mu) / mu) * 100;
        out.println("the value of a, b, c, d " + a + " " + b + " " + c + " " + d
                + "and the error percentage is " + error + "%");

    }

}

```