```java
package hyperDap.generator.presInterface;

import java.util.ArrayList;
import java.util.Map;
import java.util.Random;
import hyperDap.base.types.dataSet.ValueDataSet;
import hyperDap.generator.main.GenMain;

/**
 * This class is used to make use of generator functionality from the {@link
 guiPresentation}
 * module.
 *
 * @author soenk
 *
 */
public class PresGenerator {

  private static double precision = 0.001;

  public static ValueDataSet<Double> generate(Map<String, Double> map) {
    Random rand = new Random();
    double base = map.remove("base");
    double step = map.remove("step");
    Double precision = map.remove("precision");
    if (precision == null) {
      precision = PresGenerator.precision;
    }
    int length = map.remove("length").intValue();
    int biasNumber = 0;
    if (map.remove("bias") != null) {
      biasNumber = rand.nextInt(length / 10) + 1;
    }
    Double noise = map.remove("noise");
    if (noise == null) {
      noise = 0.0;
    } else {
      noise = 0.01;
    }
    // convert encodings to correct format
    ArrayList<String> encodings = new ArrayList<>();
    for (String encoding : map.keySet()) {
      encodings.add(encoding);
    }
    // add some randomness to encodings
    if (encodings.size() > 1) {
      for (int i = 0; i < encodings.size(); i++) {
        if (rand.nextBoolean() == true) {
          encodings.add(0, encodings.remove(i));
        }
      }
      for (int i = 0; i < encodings.size(); i++) {
        if (rand.nextInt(7) > 6) {
          encodings.add(encodings.get(i));
          length += 10;
        }
      }
    }
    // complete
    return GenMain.newDataSet(encodings, biasNumber, base, step, length, noise,
        precision.doubleValue());
  }

  public void setPrecision(double precision) {

  }

}

```