

2024 年 1 月 11 日 (木) 実施

## 応用プログラム (1) — 統計処理 —

ここでは、統計処理で最も基本的な平均値と分散とを求めるプログラムを扱う。平均値  $\langle x \rangle$  はデータの値の合計をデータ数で割ったもので、 $i$  番目のデータ値を  $x_i$ 、データ数を  $N$  とすれば、

$$\langle x \rangle = \frac{1}{N} (x_1 + x_2 + \dots + x_N) = \frac{1}{N} \sum_{i=1}^N x_i$$

と表される。分散  $\sigma^2$  は平均値と個々のデータ値との差の二乗の平均値で、

$$\begin{aligned} \sigma^2 &= \frac{1}{N} [(x) - x_1]^2 + ((x) - x_2)^2 + \dots + ((x) - x_N)^2] = \frac{1}{N} \sum_{i=1}^N ((x) - x_i)^2 \\ &= \frac{1}{N} \sum_{i=1}^N [(x)^2 - 2(x)x_i + (x_i)^2] = \langle x \rangle^2 - \frac{2\langle x \rangle}{N} \sum_{i=1}^N x_i + \langle x^2 \rangle = \langle x^2 \rangle - \langle x \rangle^2 \end{aligned}$$

と表される。ここで、 $\langle x^2 \rangle$  はデータ値の二乗の平均値で、

$$\langle x^2 \rangle = \frac{1}{N} \sum_{i=1}^N (x_i)^2$$

である。したがって、平均値とデータ値の二乗の平均値とを求めておけば、分散も求まる。

## Statistics クラス

Eclipse で jimbo の様に『自分の名前』のパッケージを指定して、次の Statistics クラスを作成せよ。(これは直接実行出来ないクラス ⇒ 後の例題及び演習に利用する)

```
package jimbo;

public class Statistics {
    double average;
    double variance;

    Statistics(double[] data) {
        CalcDb1(data);
    }

    Statistics(int[] data) {
        CalcInt(data);
    }

    void CalcDb1(double[] data) {
        double sum = 0;
        double sum2 = 0;
```

```
        for (int i=0; i<data.length; i++) {
            sum += data[i];
            sum2 += data[i]*data[i];
        }

        average = sum/data.length;
        variance = sum2/data.length - average*average;
    }

    void CalcInt(int[] data) {
        int sum = 0;
        int sum2 = 0;

        for (int i=0; i<data.length; i++) {
            sum += data[i];
            sum2 += data[i]*data[i];
        }

        average = (double)sum/data.length;
        variance = (double)sum2/data.length - average*average;
    }

    double getAverage() {
        return average;
    }

    double getVariance() {
        return variance;
    }
}
```

【解説】 **Statistics** クラスでは 2 つのコンストラクタを設け、インスタンス化の折に **double** 型または **int** 型の何れの配列が引数として渡されたかによって、それぞれに対応して**平均値**及び**分散**を計算する **CalcDb1** メソッドまたは **CalcInt** メソッドを呼び出す仕様となっている。

### 例題 1

次のプログラムを入力し、ビルドして、実行せよ。ここで、クラス名は **Sample11\_1**、ソースファイル名は **Sample11\_1.java** とする。なお、**data.txt** は一様乱数から平均値 0.5、分散 0.01 の**ガウス分布(正規分布)**に従う乱数を 10000 個発生させて、ファイルに書き出したものである。

(Eclipse では、**workspace** の直下の **Prog1** に、ダウンロードした **data.txt** を配置してから実行する)

```
package jimbo;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class Sample11_1 {
```

```

public static void main(String[] args) {
    // TODO 自動生成されたメソッド・スタブ
    double[] x = new double[10000];

    try {
        BufferedReader br = new BufferedReader(new FileReader("data.txt"));
        String iline = null;

        for (int i=0; i<x.length; i++) {
            iline = br.readLine();
            x[i] = Double.parseDouble(iline);
        }

        Statistics stat = new Statistics(x);
        System.out.println("平均値="+stat.getAverage());
        System.out.println("分散="+stat.getVariance());
        br.close();

    } catch (FileNotFoundException fe) {
        System.out.println("data.txt というファイルが開けません");
    } catch (IOException err) {
        System.out.println("IOException をキャッチ"+err);
    } catch (NumberFormatException exnf) {
        System.out.println("double 型でないものが含まれています");
    }
}
}

```

【解説】 `Double` は、`double` 型のラッパークラスであり、`parseDouble(iline)` は `iline` が表す値に初期化された新しい `double` 値を返す。

## 例題 2

次のプログラムリストの空欄を埋めたプログラムを作成し、ビルドして実行せよ。ここで、クラス名は `Sample11_2`、ソースプログラム名は `Sample11_2.java` とする。このプログラムは、サイコロを振って出る目のシミュレーションで、サイコロを 10000 回振って出る目の平均値を 10000 個取り、その分布を調べるために平均値と分散とを計算するものである。

```

package jimbo;

public class Sample11_2 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int MAX1 = 10000;
        final int MAX2 = 10000;
        int[] dice = new int[MAX1];
        double[] ave = new double[MAX2];

        for (int j=0; j<MAX2; j++) {

```

```

        for (int i=0; i<MAX1; i++)
            dice[i] = (int) (Math.random()*6)+1;

        Statistics stat1 = new Statistics(dice);
        ave[j] = stat1.getAverage ();
    }

    Statistics stat2 = new Statistics(ave);

    System.out.println(MAX1+"回分の平均値 "+MAX2+"個 に対する平均値="
        +stat2.getAverage ());
    System.out.println(MAX1+"回分の平均値 "+MAX2+"個 に対する分散="
        +stat2.getVariance ());
}
}

```

【解説】 `Math` クラスの `random` メソッドは、0.0 以上 1.0 未満の `double` 型の乱数を返す。

$0.0 \leq \text{Math.random()} * 6 < 6.0$  となる。



`(int) (Math.random()*6)` から得られる乱数は 0, 1, 2, 3, 4, 5 の何れか



`(int) (Math.random()*6)+1` から得られる乱数は 1, 2, 3, 4, 5, 6 の何れか

### 演習

次のプログラムリストの空欄を埋めたプログラムを作成し、ビルドして実行せよ。ここで、クラス名は `Ex11`、ソースプログラム名は `Ex11.java` とする。このプログラムは、コインを投げて表が出たら `1`、裏が出たら `0` と数えるものとし、`コインを 10000 回投げた場合の平均値を 10000 個取り、その分布を調べるために平均値と分散とを計算する` プログラムを作成し、ビルドして実行せよ。

```

package jimbo;

public class Ex11 {

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        final int MAX1 = 10000;
        final int MAX2 = 10000;
        int[] 1)_____ = new int[MAX1];
        double[] 2)_____ = new double[MAX2];

        for (int j=0; j<MAX2; j++) {
            for (int i=0; i<MAX1; i++)
                coin[i] = (int) (Math.random()*2);

            Statistics 3)_____ = new Statistics(coin);
            ave[j] = stat1.getAverage();
        }
    }
}

```

```
Statistics 4) _____ = new Statistics(ave);

System.out.println(MAX1+"回分の平均値 "+MAX2+"個 に対する平均値="
    +stat2.getAverage());
System.out.println(MAX1+"回分の平均値 "+MAX2+"個 に対する分散="
    +stat2.getVariance());
}
}
```

**提出物 :**

- 1) **例題 1**, **例題 2** 及び**演習**のプログラムの**コンソールへの出力結果**をコピーして貼り付けたテキストファイル **res11.txt** をメールに添付する。
- 2) **演習**で空欄に適切な語句を埋めたソースプログラムのファイル **Ex11.java** をメールに添付する。
- 3) 第 11 回理解度確認用の**質問ファイル** **Prog1\_Questions\_11th.txt** に解答を記入して、メールに添付する。

\* メールのは名は『**プログラミング 1 第 11 回課題**』(鍵括弧は要らない) とする。