

注 意 事 項

※すべての問題は出題範囲の学習内容と記載する追加の説明を用いることで達成可能である。出題範囲以外の知識を用いてより良いコードを記述することは認める。また、出題範囲の学習内容を「必ず利用すること」を強制するものではない。

※各問題で作成する JAVA のソースファイル (○○○.java) のみを提出すること。

※各問題で作成する JAVA のソースファイル名は指定に従うこと。

※各問題で作成する JAVA のソースファイルの文字コードはシフト JIS とする。

※実行例の赤字はキーボード入力を表すものである。

※キーボード入力の数値の制約として〔型 (値の範囲)〕を記載している場合は従うこと。また〔型〕のみ記載している場合は当該型が扱える値の範囲の意味である。以下に例を示す。

〔int 型 (0 ~ 5)〕は 0 から 5 までの整数

〔int 型 (0 以上)〕は 0 以上で int 型が扱える値の範囲の上限までの整数

〔int 型〕は int 型が扱える値の範囲の整数

問題 1 統計クラス設計

次はインタフェース `Statistics` の宣言である。

〔インタフェース `Statistics`〕

```
interface Statistics{
    void setDataArray(int[] a); // 対象のデータ配列の設定
    double getStat1(); // 現在の処理対象のデータの 1 つ目の統計値の受取
    double getStat2(); // 現在の処理対象のデータの 2 つ目の統計値の受取
    double getStat3(); // 現在の処理対象のデータの 3 つ目の統計値の受取
}
```

このインタフェース `Statistics` を実装したクラス `MyStatistics` を次に説明するように宣言しなさい。クラス `MyStatistics` のメソッド `setDataArray(int[] a)` により処理の対象となる整数〔int 型〕の配列を設定することができる。その後、メソッド `getStat1()` により平均値、`getStat2()` により分散値、`getStat3()` により標準偏差をそれぞれ受け取ることができる。インタフェース `Statistics` を通してクラス `MyStatistics` を用いた処理を行うクラス `Final01` のコードとその実行結果を以下に示す。これらの実行例にならい、クラス `MyStatistics` を宣言するコードを記述しなさい。インタフェース `Statistics` とク

ラス `MyStatistics`, クラス `Final01` を 1 つのソースファイルにまとめ, ファイル名を `Final01.java` として保存し, コンパイルと実行ができることとする.

[クラス `Final01`]

```
class Final01{
    public static void main(String[] args){
        int[] scores = {78,68,56,93,41,66,82,95,36,89};
        Statistics mystat = new MyStatistics();
        mystat.setDataArray(scores);
        System.out.println("平均は"+mystat.getStat1());
        System.out.println("分散は"+mystat.getStat2());
        System.out.println("標準偏差は"+mystat.getStat3());
    }
}
```

[実行結果]

```
平均は 70.4
分散は 393.44000000000005
標準偏差は 19.835322029147903
```

問題2 ロジスティック写像

ロジスティック写像とは式1で与えられる関数 $f(x)$ をいう。ロジスティック写像を用いることにより、カオスという複雑な振る舞いを容易に生成することができる。

$$f(x) = a * x * (1.0 - x), 0.0 \leq x \leq 1.0 \quad (\text{式 1})$$

具体的には、式1を漸化式で表現した式2を用いる。係数 a [double型 (0.0 ~ 4.0)] の値を与えたのち、 $x_0, x_1, x_2, x_3 \dots$ と求めていくと、その変化が係数 a の値に応じて収束したり、または特定の値を周期的に繰り返したり、カオスという複雑な振る舞いをする。

$$x_{n+1} = a * x_n * (1.0 - x_n), n \geq 0, x_0 \text{ は乱数 [double型 (0.0 ~ 1.0)]} \quad (\text{式 2})$$

図1は、係数 a を 0.0 以上 4.0 未満まで 0.08 刻みの各刻みにおいて、初期値 x_0 を乱数 [double型 (0.0 ~ 1.0)] により 50 個生成し、各係数 a と各初期値 x_0 を用いて式2を繰り返し計算した結果の値 $x_{150} \sim x_{200}$ をプロット (散布図) したものである。横軸は係数 a であり、縦軸は値 $x_{150} \sim x_{200}$ である。図上で1点に見える箇所も複数の点が重なっていることに注意すること。

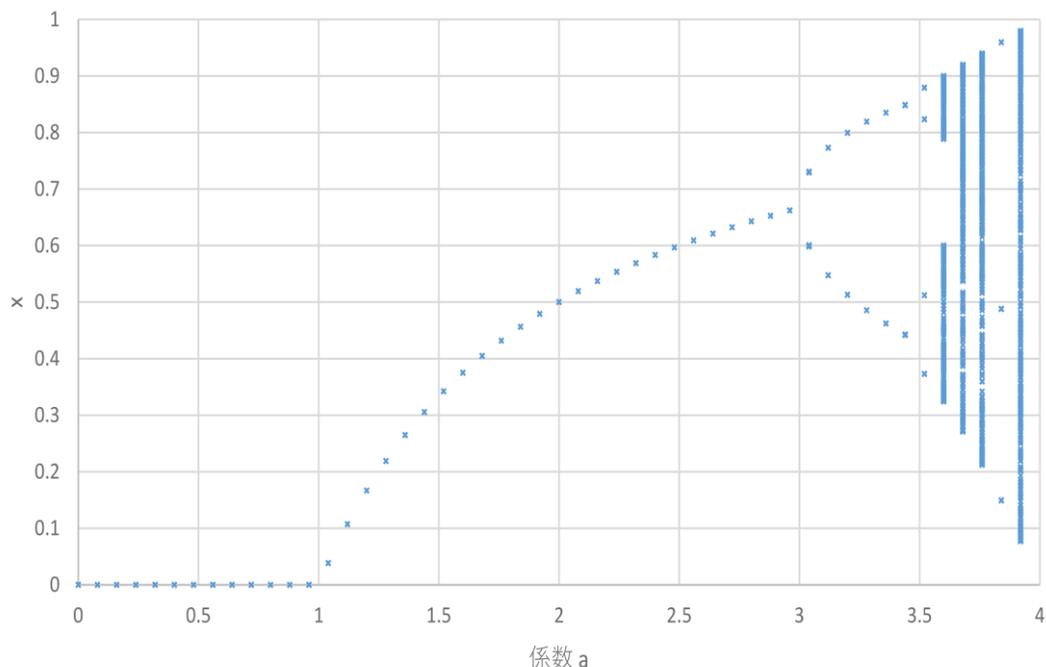


図1 ロジスティクス写像

図1をプロットする点 (横軸の係数 a 値と縦軸の値 x の組) の集まりを下記の出力形式で

ファイル `data2.csv` に出力するコードを記述しなさい。ここで、1つの係数 a と1つの初期値 x_0 が与えられたとき、式 2 を用いて値 $x_{150} \sim x_{200}$ を計算する処理を1つのスレッドが担当するマルチスレッド処理として実現することとし、結果は1つの出力ファイル `data2.csv` にまとめることとする。コードのファイル名を `Final2.java` として保存し、コンパイルと実行ができることとする。

〔出力形式〕

1行ごとに（与えられた係数 a ，得られた x ）を出力

〔出力ファイル `data2.csv` の中程の例〕

：

3.36, 0.8352435686114976	} 同じ係数 a において $x_{150} \sim x_{200}$ はそれぞれ別の行として出力する
3.36, 0.4623754790075504	
3.36, 0.8352435686114981	
3.36, 0.4623754900754933	
3.44, 0.8486710635660532	
3.44, 0.44179400364510757	

：

〔アドバイス〕

データが得られたら、エクセルでデータを読み込みグラフ表示してみる。上記の散布図と同じような図となるかを確認する。

問題3 コロナ感染者数解析

厚生労働省より新型コロナウイルス感染症に関するデータが公開されている。図2は公開されている日別の陽性者数データ (`pcr_positive_daily.csv`) をもとにしたグラフであり、2020年1月16日から2021年1月9日までの日別の陽性者数(青)、7日間移動平均(橙)、累積陽性者数(灰)を折れ線グラフとしてプロットしたものである。ここで7日間移動平均とは、日毎に計算される値であり、ある与えられた日の7日間移動平均はその日を含む過去7日間の平均値として計算される。

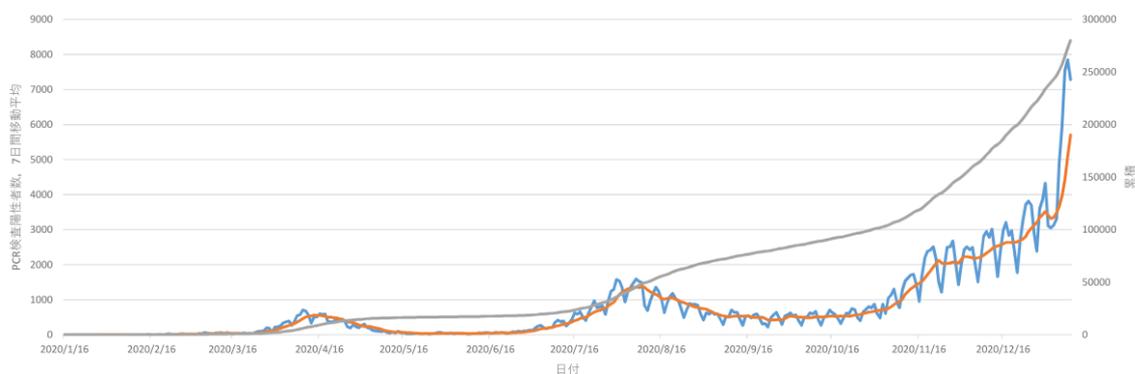


図2 新型コロナウイルス感染症に関する陽性者数の推移

講義 **JAVA** プログラミング2 のホームページに掲載している上記 **CSV** ファイルを各自のパソコンにダウンロードしなさい。このファイルに保存されている日別の陽性者数データを読み込み、図2にならぬ日別の陽性者数、7日間移動平均、累積陽性者数をファイル `data3.csv` に下記を参考にしながら出力するコードを記述しなさい。但し、7日間移動平均の計算を行うとき2020年1月15日以前の日別の陽性者数は0として扱うこと。コードのファイル名を `Final03.java` として保存し、コンパイルと実行ができることとする。

〔入力ファイル `pcr_positive_daily.csv` の冒頭部分例〕

日付,PCR 検査陽性者数(単日)

2020/1/16,1

2020/1/17,0

2020/1/18,0

2020/1/19,0

2020/1/20,0

2020/1/21,0

2020/1/22,0

2020/1/23,0

2020/1/24,1

2020/1/25,1

:

〔出力ファイル data3.csv の冒頭部分例〕

日付, PCR 検査陽性者数(単日), 7日間移動平均, 累計

2020/1/16, 1, 0.14285714285714285, 1

2020/1/17, 0, 0.14285714285714285, 1

2020/1/18, 0, 0.14285714285714285, 1

2020/1/19, 0, 0.14285714285714285, 1

2020/1/20, 0, 0.14285714285714285, 1

2020/1/21, 0, 0.14285714285714285, 1

2020/1/22, 0, 0.14285714285714285, 1

2020/1/23, 0, 0.0, 1

2020/1/24, 1, 0.14285714285714285, 2

2020/1/25, 1, 0.2857142857142857, 3

:

〔アドバイス1〕

データが得られたら、エクセルでデータを読み込み折れ線グラフ表示してみる。上記のグラフと同じような図となるかを確認する。

〔アドバイス2〕

入力ファイルは一行単位で日付と日別の陽性者数がカンマが区切られるテキストファイルである。クラス `BufferedReader` のメソッド `readLine()` は一行単位で文字列として入力されるため、カンマの前後で文字列の分割処理が必要である。クラス `String` のメソッド `indexOf()` とメソッド `substring()` を用いればよい。

問題4 BMI 計算

BMI を計算して表示する GUI アプリケーションを作成する。BMI とは肥満度を示す指標であり、式3で計算できる。

$$\text{BMI 値} = \text{体重 kg} \div (\text{身長 m} \times \text{身長 m}) \quad (\text{式 3})$$

図3はアプリケーションの実行例である。テキストフィールドに身長〔double型(0.0以上)〕と体重〔double型(0.0以上)〕を入力し、BMI計算ボタンを押下すると、右側にBMI値が表示される。

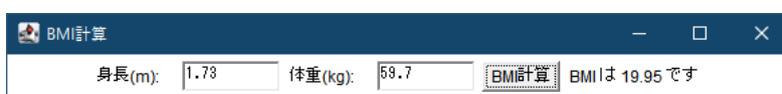


図3 BMI を計算するアプリケーション

上記の実行例にならい、BMI 計算を行う GUI アプリケーションのコードを記述しなさい。色やフォントサイズなどは図3を参考におおよそ同程度とすること。コードのファイル名を Final04.java として保存し、コンパイルと実行ができることとする。

〔アドバイス1〕

GUI 部品は左から、ラベル、テキストフィールド、ラベル、テキストフィールド、ボタン、ラベルを用いている。複数の GUI 部品を画面に配置するには Panel クラスを用いればよい。第12回課題プリントの課題5を参考にすること。

〔アドバイス2〕

BMI 計算ボタンを押下したとき、テキストフィールドにアクセスしデータを受け取る。その後、BMI 値を計算したら右端のラベルにアクセスしてその文字列データを更新する。これらのテキストフィールドやラベルはクラスのメンバーとして宣言しておき、BMI 計算ボタンにマウスイベントのリスナを登録すれば良い。

問題5 ピンボール

ピンボールを実行する GUI アプリケーションを作成する。図4はアプリケーションの実行例である。ボールが水面に落ちないように落ちてくるボールをラケットで打ち返すゲームである。ラケットはマウスの水平の動きに同期して動く。ボールは45度の傾き方向（左上から右下，右上から左下）の直線運動のみとする。ボールは画面の左端や右端，上端に到達すると跳ね返る。水面に到達した際にラケットがあれば跳ね返り，そうでなければアプリケーションを終了する。

※JAVA プログラミング2のホームページにある動作例の動画も併せてご覧ください。

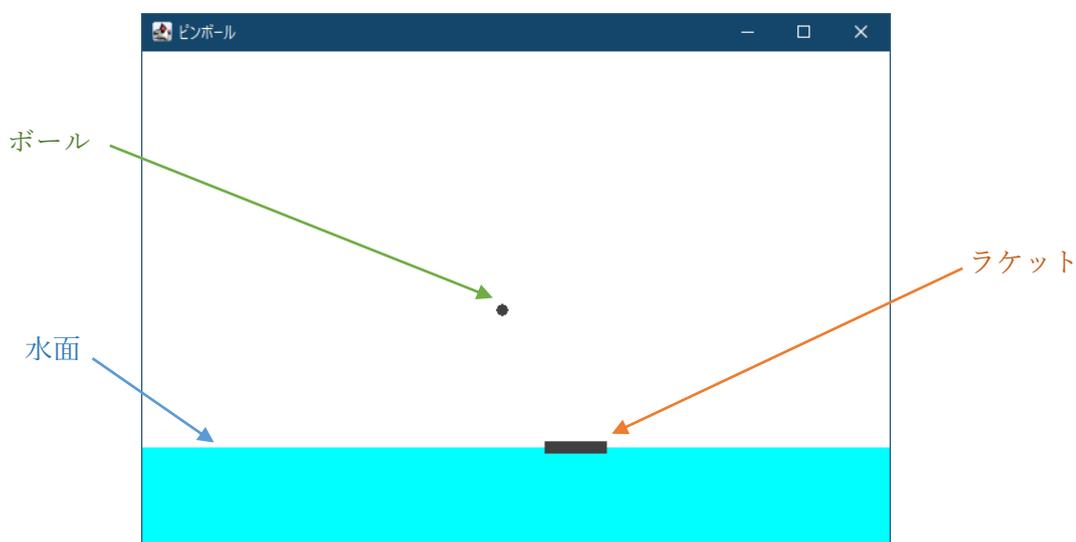


図4 ピンボールを実行するアプリケーション

上記の実行例にならい，ピンボールを実行する GUI アプリケーションのコードを記述しなさい。ウィンドウサイズ，背景の色，ボールの色やサイズ，ボールの移動速度，水面の高さ，ラケットの色やサイズは各自で設定を行っても良い。但し，コードのファイル名を **Final05.java** として保存し，コンパイルと実行ができることとする。

〔アドバイス1〕

スレッドを用いてボールの動きを処理する。スレッドは無限ループ（10ms 前後のスリープを入れる）で処理を実行する。毎回ボールの位置を更新し，端または水面に到達したら，状況に応じて跳ね返りまたはアプリケーション終了を処理する。

〔アドバイス2〕

ボールの座標を (ball_x, ball_y) として表現し，その移動量を (ball_x_delta, ball_y_delta) と表現する。例えば，ボールの初期座標は (0,0) とし，移動量の初期値は

(1,1)としておく。上記スレッドは無限ループの中で毎回次のようにボールの位置を更新する。

```
ball_x += ball_x_delta;
```

```
ball_y += ball_y_delta;
```

このあと、更新された `ball_x` と `ball_y` をみながら、端や水面に到達したかを判断する。跳ね返りは、状況に応じて次のいずれかを実行し、移動量の符号を反転させれば良い。

```
ball_x_delta *= -1;
```

```
ball_y_delta *= -1;
```

以上