

プログラミング演習 2 (第 6 回) 課題

2023年10月27日

注意事項

ソースファイルの先頭には必ず「学籍番号」「氏名」「課題番号」を下記の例のようにコメントとして入れること

```
// 学籍番号:22H000, 氏名:産大太郎, 課題番号:課題1
```

```
class Kadai06_1
{
    public static void main(String[] args){
        {
            System.out.println("学籍番号:22H000, 氏名:産大太郎, 課題番号:課題1");
        }
    }
}
```

課題 1. (Kadai06_1.java) [出題目的:繰り返しの処理]

実行例のように、int 型で 2 以上の自然数をキーボードから読み込み、べき乗が何回までオーバーフローせずに計算できるか出力するプログラムを作成しなさい。ただし、オーバーフローは int 型より大きな値が扱える long 型で計算した値と、int 型で計算した値が同じであるかどうかで判定しなさい。

[実行例] (下線部分はキーボードからの入力)

自然数を入力してください 27

27 の 1 乗は 27

27 の 2 乗は 729

27 の 3 乗は 19683

27 の 4 乗は 531441

27 の 5 乗は 14348907

27 の 6 乗は 387420489

int 型では 27 の 7 乗でオーバーフローしました

自然数を入力してください 10

10 の 1 乗は 10

10 の 2 乗は 100

10 の 3 乗は 1000

10 の 4 乗は 10000

10 の 5 乗は 100000

10 の 6 乗は 1000000

10 の 7 乗は 10000000

10 の 8 乗は 100000000

10 の 9 乗は 1000000000

int 型では 10 の 10 乗でオーバーフローしました

課題 2. (Kadai06_2.java) [出題目的:2 次元配列, length]

下記の通りの 3 科目分の点数のデータあるもととして、配列を初期化し、各科目ごとの平均値を求め double 型の配列に格納し、出力しなさい。(各科目の個数が異なること注意)

```
int[][] tensuu = { {78, 59, 97, 82, 92, 60, 79, 87, 84, 72},
                   {9, 5, 8, 9, 10},
                   {88, 43, 55, 76, 84, 66, 75, 80}};
```

なお、配列の長さに対しては必ず length を使用してください。(10 や 5 と直接記述しないこと)

さらに、実行例のように、何人目かをキーボードで入力させ、対応する値を出力しなさい。その際、平均点よりも上が下か、また、値が存在しない場合はその旨出力すること。

[実行例] (下線部分 はキーボードからの入力)

1 つ目の科目の平均点は 79.0
2 つ目の科目の平均点は 8.2
3 つ目の科目の平均点は 70.875
何人目を知りたいですか？

3

1 つ目科目は 97 点で平均点以上です
2 つ目科目は 8 点で平均点未満です
3 つ目科目は 55 点で平均点未満です

1 つ目の科目の平均点は 79.0
2 つ目の科目の平均点は 8.2
3 つ目の科目の平均点は 70.875
何人目を知りたいですか？

7

1 つ目科目は 79 点で平均点以上です
2 つ目科目は点数がありません
3 つ目科目は 75 点で平均点以上です

1 つ目の科目の平均点は 79.0
2 つ目の科目の平均点は 8.2
3 つ目の科目の平均点は 70.875
何人目を知りたいですか？

10

1 つ目科目は 72 点で平均点未満です
2 つ目科目は点数がありません
3 つ目科目は点数がありません

1 つ目の科目の平均点は 79.0
2 つ目の科目の平均点は 8.2
3 つ目の科目の平均点は 70.875
何人目を知りたいですか？

12

1 つ目科目は点数がありません
2 つ目科目は点数がありません
3 つ目科目は点数がありません

課題3. (Kadai06_3.java)

〔出題目的: 繰り返しの処理〕

実行例に示すような、以下の数字を当てゲームを作成しなさい。

- 0~99 までの整数乱数を 1 つ発生させ、答えとする。
(乱数の発生 (int) (Math.random()*100))
- 正解ができるまで、キーボードから数字を入力させる。ただし、入力した値に対し、答えが大きいのか小さいのかヒントを出力する。
- 正解が入力されたら、実行例のように何回目かで正解したのか出力する。

[実行例] (下線部分 はキーボードからの入力)

数字を当ててください(0 から 99 まで)

50

50 より大きいです

78

78 より小さいです

67

67 より小さいです

58

58 より大きいです

62

62 より大きいです

64

64 より大きいです

65

7 回目で正解しました！

数字を当ててください(0 から 99 まで)

66

66 より小さいです

33

33 より大きいです

54

54 より大きいです

59

4 回目で正解しました！

課題 4. (Kadai06_4.java)

〔出題目的: 文字列の長さ, 文字の判定〕宿題可

参考プログラム 1 (Kadai06_4_1) は、指定したファイルの中身を表示するプログラムです。参考プログラム 2 (Kadai06_4_2) はキーボードから入力した文字列を 1 文字ずつ出力するプログラムです。これらプログラムを参考に、実行例のように指定したファイルの行数、大文字の数、小文字の数、その他の数を表示するプログラムを作成しなさい。(まずは、Kadai06_4_1.java, Kadai06_4_2.java を作成し、参考プログラムの動作を確認してください。)

〔実行例〕(下線部分はキーボードからの入力、

ファイル名は現在使っているディレクトリ内にあるファイル名を指定してください。
指定するファイルの中身によって値は異なります。)

ファイル名を入力してください

Kadai06_4_2.java

12 行ありました

小文字 : 191

大文字 : 18

その他 : 128

ファイル名を入力してください

Kadai06_4.java

38 行ありました

小文字 : 467

大文字 : 38

その他 : 507

〔参考プログラム 1〕

```
import java.io.*;

class Kadai06_4_1 {
    public static void main(String[] args) throws IOException {
        System.out.println("ファイル名を入力してください");

        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String file_name = br.readLine();

        try {
            BufferedReader br_f = new BufferedReader(new FileReader(file_name));
            String str;
            while ((str = br_f.readLine()) != null) {
                System.out.println(str);
            }
            br_f.close();
        } catch (IOException e) {
            System.out.println("入力エラーです");
        }
    }
}
```

〔参考プログラム 2〕

```
import java.io.*;

class Kadai06_4_2 {
    public static void main(String args[]) throws IOException {
        System.out.println("文字列を入力してください");
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String str = br.readLine();
        for (int i = 0; i < str.length(); i++) {
            System.out.println(str.charAt(i));
        }
    }
}
```

課題5. (Kadai06_5.java) [出題目的:二次元配列, break 文, Math クラス]宿題可

以下の参考プログラムは、5 枚のカードをランダムに配布するプログラムです(カードの重複を許しています)。参考プログラムの main 部分を以下のように変更し、ポーカーゲームのプログラムを作成してください。ただし、役の判定はプログラムでは行いません。

1. 同じカードが配られないよう次の処理を加えてください。
4 行 13 列の **boolean 型の二次元配列**を作成する。(作成時の値はすべて **false** となっている。)
配布したカードの値を **true** に変更する。(true になっていれば、既に配ったことを意味する)
配布前に決定したカードが配布済みであるか調べ、true であれば配らないようにする。
(例えば、配列変数 cards とすると、cards[0][0] は「スペードの 1」を示し、cards[0][0]の値が false であれば「スペードの 1」は配布前, true であれば配布済みと考える。)
2. 1 回だけカード交換の処理を加えてください。
交換する枚数を聞いてください。
何番目のカードを交換するか聞き、カードの再配布を行ってください。

[参考プログラムの実行結果例]

カードは	カードは	カードは
[1] ダイヤ 12	[1] ハート 12	[1] ダイヤ 11
[2] スペード 12	[2] スペード 2	[2] ダイヤ 12
[3] クラブ 4	[3] ダイヤ 2	[3] クラブ 3
[4] クラブ 10	[4] クラブ 3	[4] ダイヤ 12
[5] スペード 8	[5] スペード 2	[5] ダイヤ 12

[実行例] (下線部分 はキーボードからの入力)

カードは [1] クラブ 5 [2] ダイヤ 9 [3] スペード 8 [4] ハート 13 [5] ハート 5 交換する枚数は? <u>2</u> 1 枚目は何番のカードを交換しますか? <u>2</u> 2 枚目は何番のカードを交換しますか? <u>3</u> カードは [1] クラブ 5 [2] スペード 9 [3] クラブ 13 [4] ハート 13 [5] ハート 5	カードは [1] クラブ 10 [2] スペード 4 [3] クラブ 12 [4] ハート 3 [5] クラブ 7 交換する枚数は? <u>4</u> 1 枚目は何番のカードを交換しますか? <u>5</u> 2 枚目は何番のカードを交換しますか? <u>2</u> 3 枚目は何番のカードを交換しますか? <u>4</u> 4 枚目は何番のカードを交換しますか? <u>1</u> カードは [1] スペード 6 [2] ダイヤ 10 [3] クラブ 12 [4] スペード 8 [5] スペード 1
カードは [1] ダイヤ 5 [2] クラブ 2 [3] ダイヤ 3 [4] スペード 10 [5] クラブ 4 交換する枚数は? <u>1</u> 1 枚目は何番のカードを交換しますか? <u>4</u> カードは [1] ダイヤ 5 [2] クラブ 2 [3] ダイヤ 3 [4] スペード 6 [5] クラブ 4	カードは [1] スペード 11 [2] スペード 10 [3] ダイヤ 10 [4] ダイヤ 9 [5] ハート 9 交換する枚数は? <u>0</u> カードは [1] スペード 11 [2] スペード 10 [3] ダイヤ 10 [4] ダイヤ 9 [5] ハート 9

[課題5 参考プログラム]

```
import java.io.*;

class Kadai06_5_sample
{
    static int tefuda_m[] = new int[5];
    static int tefuda_n[] = new int[5];

    public static void showCards()
    {
        System.out.println("カードは");
        String str_out;
        for(int i=0;i<5;i++){
            System.out.print("[ " +(i+1)+"] ");
            switch (tefuda_m[i]){
                case 0:
                    str_out= "スペード¥t";
                    break;
                case 1:
                    str_out="ハート¥t";
                    break;
                case 2:
                    str_out="ダイヤ¥t";
                    break;
                case 3:
                    str_out="クラブ¥t";
                    break;
                default:
                    str_out="??? ";
            }
            System.out.println(str_out+(tefuda_n[i]+1));
        }
    }

    public static void main(String[] args) throws IOException
    {
        //課題では入力があるのであらかじめこのように記述しています。
        int count=0,mark,num;
        while(count<5){
            mark = (int)(Math.random()*4); // 0:スペード 1:ハート 2:ダイヤ 3:クラブ
            num = (int)(Math.random()*13);
            tefuda_m[count]=mark;
            tefuda_n[count]=num;
            count++;
        }
        showCards(); //showCards()メソッドを呼び出すことで出力している
    }
}
```

- 演習時間内に出来なかったプログラムのソースファイル(.java)は、**下記の期限までにWebClass**で提出すること。
- 提出されたプログラムは**すべてチェック**します。動作しないプログラムが提出されている場合、提出物を**すべて無効**とします。

提出期限: **再来週木曜日(11月9日)18:40** まで

提出場所: **WebClass**: 金4後 プログラミング演習2 H 松本 恵治 1 2023
06-3 課題(宿題) Kadai06_4-5

注意: 06-3 課題(宿題) Kadai06_4-5 には、課題チェックシート(提出用)でチェック(合格)されたファイルを**再度提出してはならない**。提出している場合、課題としてチェックされたものが取り消しとなる。

- ソースファイルの先頭には必ず「学籍番号」「氏名」「**宿題番号**」を下記の例のようにコメントとして入れること。

```
/* 学籍番号:22H000, 氏名:産大太郎, 宿題番号:課題 4 */  
class Kadai06_4  
{
```

- 期限に遅れた場合は、**評価外**とする。ただし、病気等の理由がある際には**診断書**等を持参の上、担当教員に了解を得ること。
- **Lesson8**の Sample1(Sample08_01.java)から Sample3(Sample08_3.java)までソースファイル(.java)3個を提出すること。

提出期限: **再来週木曜日(11月9日)18:40** まで

提出場所: **WebClass**: 金4後 プログラミング演習2 H 松本 恵治 1 2023
07-1 予習 Sample08_01-03

- ソースファイルの先頭には必ず「学籍番号」「氏名」「**Sample番号**」を下記の例のようにコメントとして入れること。

```
// 学籍番号:22H000, 氏名:産大太郎, Sample 番号:1  
class Sample08_01  
{
```

- **次回のLesson8前半**の説明、課題を行うので、**Lesson8 前半**の内容を熟読しておくこと。