

ロジックテスト

試験時間 90 分

【注意事項】

1. 試験問題は試験監督員の指示があるまで開けてはいけません。
2. 実際にプログラムを書いてもらう問題です。
3. IntelliJ IDEA を使用し、設問ごとにクラスファイルを作成してください。
4. Java の API リファレンスは閲覧しても OK です。
5. 問題に関しての質問は受け付けません。
6. コードが綺麗かどうか、コメントが書かれているかどうかは問いません。出題の意図を理解した処理が書けているかどうかだけを採点します。問いの途中までできているという場合でも配点の何割かを加点するということがあるため、途中であきらめずに最後まで取り組みましょう。

【問題 1】（15 点）

繰り返し構文を使用して、掛け算の九九表を出力するプログラムを作成してください。

※下記実行結果通りに出力してください。（計算結果が 1 桁の場合も 2 桁の場合も下 1 桁の位置は合わせてください。）

実行結果

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

confidential

【問題 2】（15 点）

任意の数値(時、分、秒)をそれぞれ変数に代入してください。そして、今代入した時分秒の 1 秒後の時刻を表示するプログラムを作成してください。

実行結果 例) 引数が 15 (時) , 4 (分) , 59 (秒) の場合

15 時 4 分 59 秒の 1 秒後は 15 時 5 分 0 秒です

【問題 3】（15 点）

西暦を表す任意の数値を変数 `seireki` に代入してください。その後、今代入した数値が何世紀であるかを出力するプログラムを作成してください。

ちなみに

19 世紀は 1801 年から 1900 年まで

20 世紀は 1901 年から 2000 年まで

です。

※上記はあくまで例で、1801 年から 2000 年まで出力できればよいわけではありません。

実行結果 例) 引数が 2000 の場合

2000 年は 20 世紀です

実行結果 例) 引数が 2001 の場合

2001 年は 21 世紀です

【問題 4】（15 点）

西暦を表す任意の数値を変数 `seireki` に代入してください。その後、今代入した数値がうるう年であるかどうかを出力するプログラムを作成してください。うるう年かどうかを判定する処理は、引数が `int` 型、戻り値が `boolean` 型のメソッドとして定義してください。

うるう年の定義は以下となります。

西暦年が 4 で割り切れる年はうるう年

ただし、西暦年が 100 で割り切れる年は平年

ただし、西暦年が 400 で割り切れる年はうるう年

実行結果 例) 引数が 2000 の場合

2000 年はうるう年です

実行結果 例) 引数が 2100 の場合

2100 年はうるう年ではありません

【問題 5】 (20 点)

以下の配列から、最も出現回数の多いスポーツとその出現回数を入力するプログラムを作成してください。

(最も出現回数の多いスポーツとその出現回数はプログラムで算出してください。自分の頭で考えて、それを `System.out.println()` するのは NG です。)

```
{"soccer","baseball","jogging","jogging","soccer","surfing","surfing","soccer","surfing","baseball","boxing","surfing","soccer","surfing","jogging","boxing","jogging","baseball","soccer","soccer"}
```

※上記、配列のテキストは「配列.txt」でデスクトップに保存してありますのでコピーアンドペーストして利用してかまいません。（「配列.txt」をファイルとして読み込んで処理をする、という意味ではありません。）。準備の都合で「配列.txt」がデスクトップに用意されていない場合は、お手数ですが、本問題用紙を元に、ご自身で打ち込んでください。

実行結果

最も出現回数の多いスポーツは xxxx で、出現回数は x 回です

(※) **x** の部分は算出した値

【問題 6】 (20 点)

3 つの異なる数字(例えば 3,5,9)の組み合わせでできる 3 桁の数字を、全て出力するプログラムを作成してください。

※3 桁の数字はプログラムでランダムに作成するようにしてください。

※単に 3 つの数字の組み合わせを自分の頭で考えて、それを `System.out.println()` するのは NG です。

※以下のようなランダムで作成していないコードも NG です。

```
int num1 = 8;
```

```
int num2 = 4;
```

```
int num3 = 6;
```

※先頭に 0 が来ても問題ありません。

実行結果 ※順不同

359
395
539
593
935
953