

●2. コラッツの問題

コラッツの問題

出典: フリー百科事典『ウィキペディア (Wikipedia) 』

コラッツの問題（コラッツのもんだい、Collatz problem）は、数論の未解決問題のひとつである。1937年にローター・コラッツが問題を提示した。問題の結論の予想を指してコラッツの予想と言う。固有名詞に依拠しない表現としては $3n+1$ 問題とも言われ、初期にこの問題に取り組んだ研究者の名を冠して、角谷の問題、米田の予想、ウラムの予想、他にはSyracuse問題などとも呼ばれる。数学者ポール・エルデシュは「数学はまだこの種の問題に対する用意ができていない」と述べ、解決した人に500ドルを提供すると申し出た。ジェフリー・ラガリアスは2010年に、コラッツの予想は「非常に難しい問題であり、現代の数学では完全に手が届かない」と述べた[1]。

コンピュータを用いた計算により、268 までには反例がないことが確かめられている[2]。また、2011年度大学入試センター試験数学IIB第6問に題材として取り上げられた[3]。

●プログラムでは次のようなアルゴリズムになります。

1. 任意の自然数 n を入力
2. n が1であれば終了する
3. n が偶数の場合 n を2で割る
 n が奇数の場合 n に3を掛けて1を足す
4. 2へ戻る
5. 終了

【課題1】 $n=9$ の場合で実行したとき何回の繰り返してで終了するか？

- ・手計算で回数を求めてください
- ・プログラムを実装し処理の状況を表示してください
（ただし任意の自然数を入力できるようにしてください）

【クラス図】

クラス名

Collatz

フィールド定義

-number:int

メソッド定義

+Collatz():

+calcPrint():void

+setNumber(n:int):void

+main():int

-:private

+:public

【実行例】

Enter a number: 9

1 : 28

2 : 14

3 : 7

4 : 22

5 : 11

6 : 34

7 : 17

8 : 52

9 : 26

10 : 13

11 : 40

12 : 20

13 : 10

14 : 5

15 : 16

16 : 8

17 : 4

18 : 2

19 : 1

【課題 2】

シミュレーションの途中経過を出力しようとした場合、処理の開始時点では経過が何件出力されるかわかりません。

つまり経過のデータをカプセル化するためには可変長配列でデータを保持しなければなりません。

可変長配列のArrayListクラスを用いてコラッツの問題を実装してください。

クラス名

CollatzLogic

フィールド定義

-number:int

-list:ArrayList<Integer>

メソッド定義

+CollatzLogic():

+calcArray():void

+setNumber(n:int):void

+getList():ArrayList

-:private

+:public

(mainメソッド例)

```
public static void main(String[] args) {  
    int input_n;  
    System.out.print("Enter a number: ");  
    input_n = new java.util.Scanner(System.in).nextInt();  
  
    CollatzLogic cl=new CollatzLogic();  
    cl.setNumber(input_n);  
    cl.calcArray();  
    //List配列出力  
    ArrayList al=cl.getList();  
    for (int i=0;i<al.size();i++) {  
        System.out.println((i+1)+":"+al.get(i));  
    }  
}
```

