

提出日 令和2年07月29日

アルゴリズムとデータ構造 第8回課題レポート

学籍番号 (A19117)

氏名 (永尾優磨)

【課題】

1. コンピュータで数値計算を行う際の以下の事項について調べよ。単なる語句の説明にとどまらず、具体例を挙げるとなおい。
- 1.1. 情報落ち  
コンピュータで絶対の大きさが極端に異なる数字を足したり引いたりしたときに、小さい値の情報が無視されてしまう現象。
- 1.2. オーバーフロー  
コンピュータ上で、数値の計算結果がその格納領域に収まる範囲を超えること。
- 1.3. アンダーフロー  
浮動小数点演算処理について、計算結果の指数部が小さくなりすぎ、使用している記述方式では数値が表現できなくなることである。
- 1.4. 変換誤差  
ほとんどの10進数の「小数」は、2進数に正しく変換できない。そのため、変換するときに誤差が出来てしまう。
- 1.5. 桁落ち  
桁落ちとは、非常に近い大きさの小数同士で減算を行ったときに、有効数字が減る現象の事。コンピュータでは浮動小数点数の数値計算において生じる
2. 課題1～3の作業を行え。
  1. Mathクラスのメソッドを用いて、以下の数値計算を行え。計算式と計算結果の両方を表示すること。
    - [1]  $\sqrt{76}$  を計算せよ。
    - [2]  $2^{16}$  を計算せよ。
    - [3]  $e^2$  を計算せよ。
    - [4]  $\sin 30^\circ$  を計算せよ。
    - [5]  $\log_2 4294967296$  を計算せよ。
  2. コンピュータにサイコロを1万回振らせてみよ。randomメソッドを用いて、1から6の乱数を10000回発生させ、以下のような形で表示すること
    - 1の目の数=n, 確率=N
    - 2の目の数=n, 確率=N
    - 3の目の数=n, 確率=N

4の目の数=n, 確率=N  
 5の目の数=n, 確率=N  
 6の目の数=n, 確率=N

3.  $y=\sin x$ のグラフを出力せよ。ただし、 $x=0$ から $2\pi$ まで刻み幅 $\pi/12$ で変えて、 $y=\sin x$ の値を求めて画面上に文字で表示させる。プログラムを簡単にするため、右図のように90度反時計方向に回転したものを描けばよい。

【作成したプログラム】

```
public class Caluc {
    public static void main(String[] args) {
        double first = Math.sqrt(76);
        double second = Math.pow(2, 16);
        double third = Math.sin(30);
        double fourth = Math.log(4294967296L) / Math.log(2);

        System.out.println(first);
        System.out.println(second);
        System.out.println(third);
        System.out.println(fourth);
    }
}
```

プログラム1 Caluc

```
public class Glaph {
    public static void main(String[] args) {
        int lis[] = new int[6];

        int num = 0;
        for (int i = 0; i < 10000; i++) {
            num = (int) (Math.random() * 6 + 1);
            if (num == 1) {
                lis[0] += 1;
            } else if (num == 2) {
                lis[1] += 1;
            } else if (num == 3) {
                lis[2] += 1;
            } else if (num == 4) {
                lis[3] += 1;
            } else if (num == 5) {
                lis[4] += 1;
            } else if (num == 6) {
                lis[5] += 1;
            }
        }
    }
}
```

```

    }

}

for (int i = 0; i < lis.length; i++) {
    int num1 = (int) (lis[i]);
    double res = num1 / 10000.0;
    System.out.println(i + 1 + "の目の数=" + num1 + ", 確率=" + res);
}
}
}

```

プログラム2 Glaph

```

import java.lang.Math;

public class Probab {
    public static void main(String[] args) {
        int m = 10;
        for (int i = 0; i <= 24; i++) {
            int res;
            res = m + (int) (m * Math.sin(i * Math.PI / 12.0f));
            for (int j = 0; j < res; j++) {
                System.out.print(" ");
            }
            System.out.println("*");
        }
    }
}

```

プログラム3 Probab

## 【プログラムの解説】

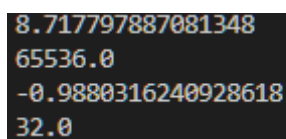
javaのMath関数を用いて、出来るだけシンプルに記述するように心がけた。

Calucは少しの記述量で実装することが出来た。

Probabは、if文を用いる事で、シンプルに記述を行った。

Glaphは、sinを出力する時に、空白を入れることで、わかりやすく仕上げる事が出来た。

## 【結果】



```

8.717797887081348
65536.0
-0.9880316240928618
32.0

```

図1 Calucの出力結果

```
1の目の数=1614 , 確率=0.1614
2の目の数=1708 , 確率=0.1708
3の目の数=1664 , 確率=0.1664
4の目の数=1621 , 確率=0.1621
5の目の数=1697 , 確率=0.1697
6の目の数=1696 , 確率=0.1696
```

図2 Glaphの出力結果

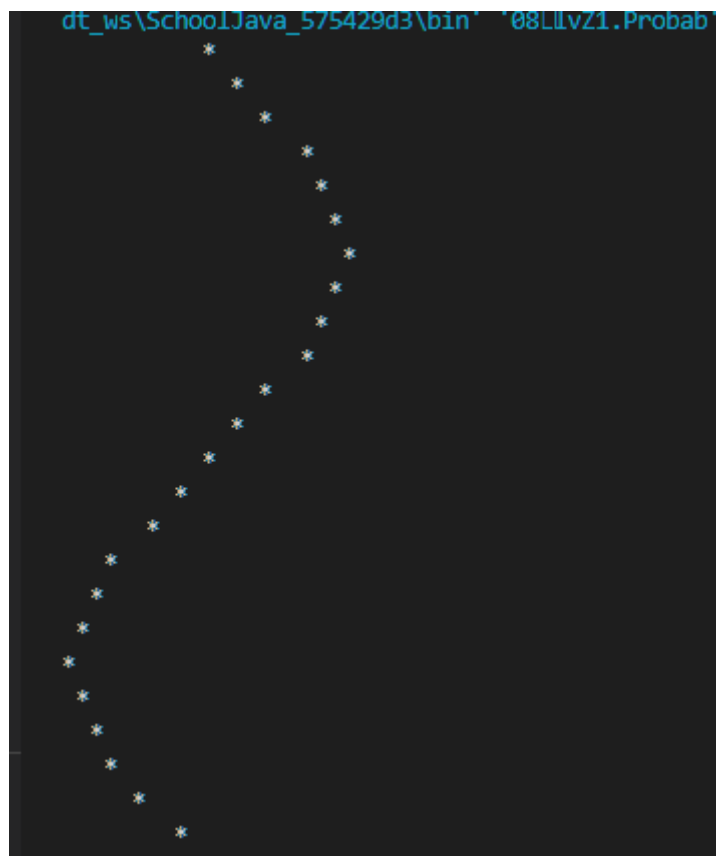


図3 Probabの出力結果

### 【考察】

Math関数を使うことでシンプルに記述出来る事がわかった。

これから、計算をする時は、すでに使われているライブラリを使って行こうと思った。

### 【参考文献】

1. 「アルゴリズムとデータ解析の授業スライド、HashMapについて」
2. 「情報落ち 【 loss of trailing digits 】 情報落ち誤差」  
<http://e-words.jp/w/%E6%83%85%E5%A0%B1%E8%90%BD%E3%81%A1.html#:~:text=%E6%83%85%E5%A0%B1%E8%90%BD%E3%81%A1%E3%81%A8%E3%81%AF%E3%80%81%E3%82%B3%E3%83%B3%E3%83%94%E3%83%A5%E3%83%BC%E3%82%BF,%E3%81%AB%E3%82%88%E3%81%A3%E3%81%A6%E8%B5%B7%E3%81%8D%E3%82%8B%E8%A8%88%E7%AE%97%E3%81%AE%E8%AA%A4%E5%B7%AE%E3%80%82>

3. 「オーバーフロー 【 overflow 】 桁あふれ」 <<http://e-words.jp/w/%E3%82%AA%E3%83%BC%E3%83%90%E3%83%BC%E3%83%95%E3%83%AD%E3%83%BC.html#:~:text=%E3%82%AA%E3%83%BC%E3%83%90%E3%83%BC%E3%83%95%E3%83%AD%E3%83%BC%E3%81%A8%E3%81%AF%E3%80%81%E3%81%82%E3%81%B5%E3%82%8C%E3%82%8B,%E3%83%90%E3%83%83%E3%83%95%E3%82%A1%E3%82%AA%E3%83%BC%E3%83%90%E3%83%BC%E3%83%95%E3%83%AD%E3%83%BC%E3%81%AA%E3%81%A9%E3%82%92%E6%8C%87%E3%81%99%E3%80%82>>
4. 「アンダーフロー」 <<https://www.weblio.jp/content/%E3%82%A2%E3%83%B3%E3%83%80%E3%83%BC%E3%83%95%E3%83%AD%E3%83%BC#:~:text=%E3%82%A2%E3%83%B3%E3%83%80%E3%83%BC%E3%83%95%E3%83%AD%E3%83%BC%E3%81%A8%E3%81%AF%E3%80%81%E6%B5%AE%E5%8B%95,%E3%81%93%E3%81%A8%E3%81%8C%E3%82%AA%E3%83%BC%E3%83%90%E3%83%BC%E3%83%95%E3%83%AD%E3%83%BC%E3%81%A8%E5%91%BC%E3%81%B0%E3%82%8C%E3%82%8B%E3%80%82>>
5. 「小数計算の誤差 0.1 + 0.2 が 0.3000000000000004 になる理由」 <<https://blog.apar.jp/program/8900/>>
6. 「桁落ち」 <<http://e-words.jp/w/%E6%A1%81%E8%90%BD%E3%81%A1.html#:~:text=%E6%A1%81%E8%90%BD%E3%81%A1%E3%81%A8%E3%81%AF%E3%80%81%E4%B8%B8%E3%82%81%E8%AA%A4%E5%B7%AE,%E3%81%AE%E6%95%B0%E5%80%A4%E8%A8%88%E7%AE%97%E3%81%AB%E3%81%8A%E3%81%84%E3%81%A6%E7%94%9F%E3%81%98%E3%82%8B%E3%80%82>>
7. 「参考文献の書き方」 <[http://www7a.biglobe.ne.jp/nifongo/ron/ron\\_04.html](http://www7a.biglobe.ne.jp/nifongo/ron/ron_04.html)>