

プログラム言語 B サイコロの問題

2023 年 11 月 19 日

指導教員：武田先生

6321120

横溝 尚也

1 課題内容

以下に述べるサイコロの問題を解く Java プログラムを作成しなさい。

出る目の確率が次のようなサイコロがある。

1 と 3 の目が出る確率はそれぞれ 0.2

2、4、6 の目が出る確率はそれぞれ 0.1

5 の目が出る確率は 0.3

これらの確率を `double` 型の配列に格納してあると仮定する。

```
double p[] = { 0.2, 0.1, 0.2, 0.1, 0.3, 0.1 };
```

(1) この配列を引数に与えて呼び出すと、サイコロをふった時に出る目を結果として返すメソッド

```
int dice(double p[])
```

を定義しなさい。

なお、`[0,1)` の範囲の擬似乱数は `Math.random()` メソッドを使用してよい。

(2) サイコロを 1 億回ふって目が出た回数をカウントして、与えられた確率に近い目が出ていることを確かめなさい。

(3) 1 億回の処理時間が短くなるようプログラムを考察し、可能ならば改良を行いなさい。

2 プログラムの内容

プログラム 1 : Dice.java

```
1 public class Dice {
2     public static void main(String args[]) {
3         double p[] = { 0.2, 0.1, 0.2, 0.1, 0.3, 0.1 };
4         Dice d = new Dice();
5         d.start(p);
6     }
7     // 処理時間の計測するメソッド
8     void start(double p[]) {
9         int n = p.length;
10        int count[] = new int[n + 1];
11        // 処理前の時刻を取得
12        long startTime = System.currentTimeMillis();
13        for (int i = 0; i < 100000000; i++) {
14            count[dice(p)] += 1;
15        }
16        // 処理後の時刻を取得
17        long endTime = System.currentTimeMillis();
18        // 処理時間の表示
19        System.out.println("処理時間：" + (endTime - startTime) + " msec");
```

```

20         for (int i = 1; i <= n; i++) {
21             System.out.printf("%d:%10d\n", i, count[i]);
22         }
23     }
24     // サイコロの出る目を返すメソッド
25     int dice(double p[]){
26         double r = Math.random();
27         if ((r >= 0)&&(r < p[0])){
28             return 1;
29         }else if((r >= p[0])&&(r < p[0]+p[1])){
30             return 2;
31         }else if((r >= p[0]+p[1])&&(r < p[0]+p[1]+p[2])){
32             return 3;
33         }else if((r >= p[0]+p[1]+p[2])&&(r < p[0]+p[1]+p[2]+p[3])){
34             return 4;
35         }else if((r >= p[0]+p[2]+p[3])&&(r < p[0]+p[1]+p[2]+p[3]+p[4])){
36             return 5;
37         }else {
38             return 6;
39         }
40     }
41 }

```

1～6 行目

Dice クラス内におけるメインメソッドの記述である。配列 p[] に確率を格納し、後述する start,dice メソッドを実行している

7～23 行目

処理時間を計測する start メソッドについての記述

- 9 行目 引数として受け取った配列の長さを n に格納
- 10 行目 ある事象が起こった回数を格納する配列 count[] を宣言
- 12 行目 処理前の時刻を取得
- 13～15 行目 dice メソッドを呼び出し、その戻り値を引数とする count[] に対してインクリメントするプログラムを 1 億回ループ
- 17 行目 12 行目からここまでの計測時間を取得
- 19 行目 取得した処理時間を出力
- 20～22 行目 各事象の出現回数である count[] の要素をそれぞれ出力

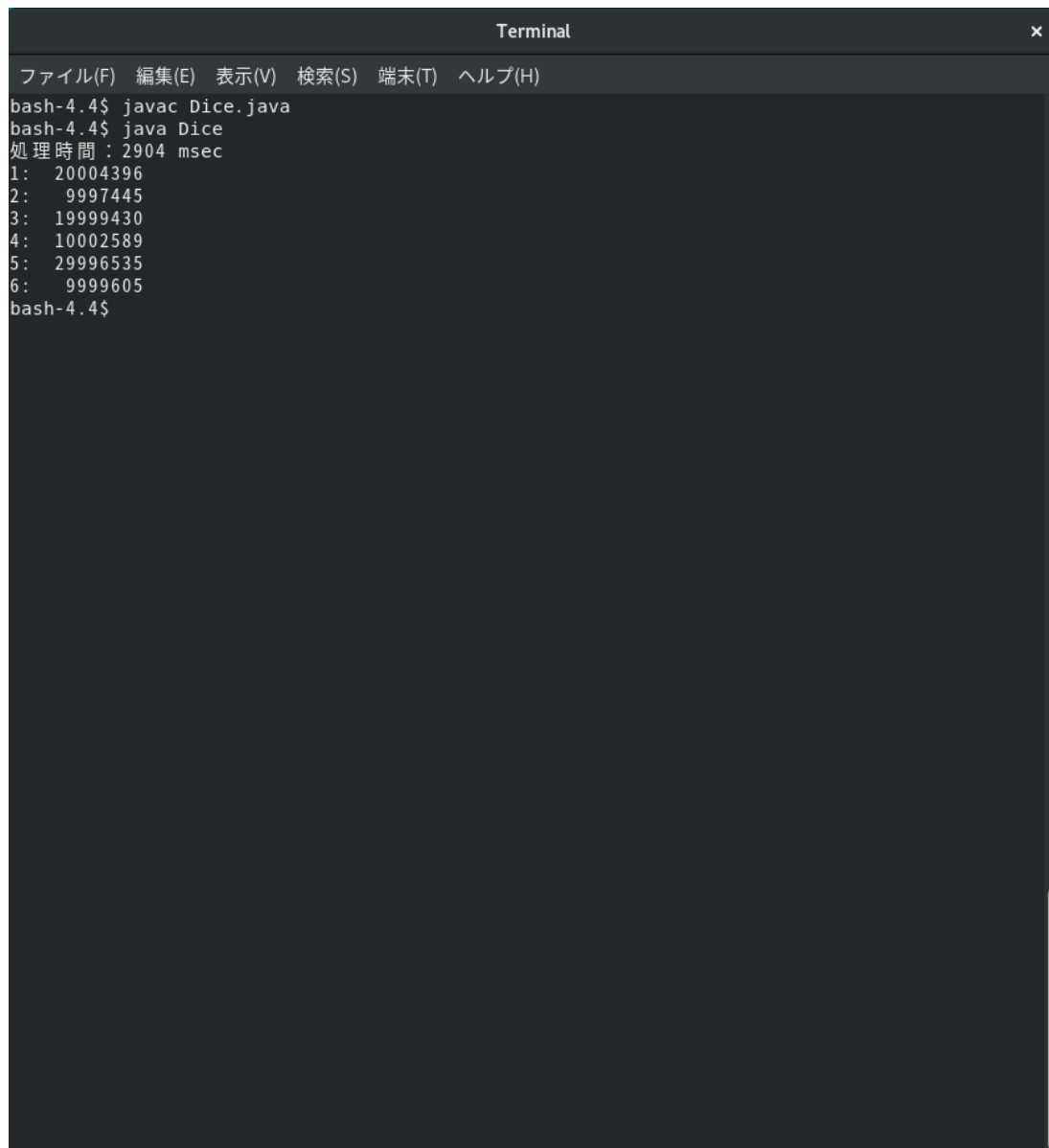
24～40 行目

サイコロの目を出力するメソッド dice についての記述

- 26 行目 [0,1) の範囲の擬似乱数を random メソッドで出力し、r に格納
- 27～38 行目 p[] に格納されている確率を左から [0,1) の数直線上に配置し、r が数直線上のどの事象

に位置しているのか条件分岐を行い、第 k 番目の事象として k を返す

3 実行結果



```
Terminal
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
bash-4.4$ javac Dice.java
bash-4.4$ java Dice
処理時間：2904 msec
1: 20004396
2: 9997445
3: 19999430
4: 10002589
5: 29996535
6: 9999605
bash-4.4$
```

図1 Dice.py の出力結果

1 億回実行サイコロを振ったときに1～6のそれぞれの出た目のカウントが {2 千万, 1 千万, 2 千万, 1 千万, 3 千万, 1 千万} に近い値が出力されていることから、引数として与えた確率 {0.2, 0.1, 0.2, 0.1, 0.3, 0.1} に収束していることがわかる