

Project 1-1

Snake-Like Increment Table*

Jiashuo WANG
5100309436

April 3, 2013

Instructor: Ling Gong

I. OBJECTIVE

This program will create small, snake-like increment tables, up to 9x9 in size. It starts from 1, and the value keep increasing by one. For the first row, it increase from left to right; for the second row, it increase from right to left; the third row, from left to right, ... , etc. Just like a snake.

I.1 Sample

```
>java SnakeTable
```

```
Enter table size 1-9, 0 to exit: 9
```

1	2	3	4	5	6	7	8	9
18	17	16	15	14	13	12	11	10
19	20	21	22	23	24	25	26	27
36	35	34	33	32	31	30	29	28
37	38	39	40	41	42	43	44	45
54	53	52	51	50	49	48	47	46
55	56	57	58	59	60	61	62	63
72	71	70	69	68	67	66	65	64
73	74	75	76	77	78	79	80	81

```
Enter table size 1-9, 0 to exit: 3
```

1	2	3
6	5	4
7	8	9

```
Enter table size 1-9, 0 to exit: 1
```

```
1
```

```
Enter table size 1-9, 0 to exit: 746
```

*Designed by L^AT_EX

please enter a number in the range 0-9
Enter table size 1-9, 0 to exit: -231
please enter a number in the range 0-9
Enter table size 1-9, 0 to exit: 0

```
F:\Course\OS\Project1-1\src\project1>javac Project1.java
F:\Course\OS\Project1-1\src\project1>java Project1
Enter table size 1-9, 0 to exit:9
1      2      3      4      5      6      7      8      9
18     17     16     15     14     13     12     11     10
19     20     21     22     23     24     25     26     27
36     35     34     33     32     31     30     29     28
37     38     39     40     41     42     43     44     45
54     53     52     51     50     49     48     47     46
55     56     57     58     59     60     61     62     63
72     71     70     69     68     67     66     65     64
73     74     75     76     77     78     79     80     81
Enter table size 1-9, 0 to exit:5
1      2      3      4      5
10     9      8      7      6
11     12     13     14     15
20     19     18     17     16
21     22     23     24     25
Enter table size 1-9, 0 to exit:2
1      2
4      3
Enter table size 1-9, 0 to exit:19
please enter a number in the range 0-9
Enter table size 1-9, 0 to exit:-7
please enter a number in the range 0-9
Enter table size 1-9, 0 to exit:7
1      2      3      4      5      6      7
14     13     12     11     10     9      8
15     16     17     18     19     20     21
28     27     26     25     24     23     22
29     30     31     32     33     34     35
42     41     40     39     38     37     36
43     44     45     46     47     48     49
Enter table size 1-9, 0 to exit:a
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Unknown Source)
    at java.util.Scanner.next(Unknown Source)
    at java.util.Scanner.nextInt(Unknown Source)
    at java.util.Scanner.nextInt(Unknown Source)
    at Project1.main(Project1.java:18)
F:\Course\OS\Project1-1\src\project1>
```

Figure 1: Screenshots of Snake-Like Increment Table

I.2 Specification

The program should repeatedly request input in the range 0-9 (sample code to get console input is below). Numbers outside this range should be politely rejected. 0 will stop execution of the program. Numbers not parseable as integers should also cause the program to exit.

For input in the 1-9 range, the program should produce a multiplication table as shown above. The formatting does not have to precisely match the above, but the numbers should be in columns. You may use any combination of tabs and spaces for formatting. I would avoid Java's NumberFormatter classes for this one, but if you're feeling daring, go for it.

II. ALGORITHM

As is shown above, the numbers in odd-numbered lines are increase, while in even-numbered lines, they are decrease. So we can classify this project into two cases. Assume that the first line is numbered as line $i = 0$, and a is the order of the table.

- For **odd-numbered** lines the numbers will range from $(a \times i + 1)$ to $(i + 1) \times a$.
- For **even-numbered** lines, the numbers will range from $(i + 1) \times a$ down to $(a \times i + 1)$.

There are also some details to be considered. For example, the robustness of the input of a. As I use the class Scanner to input from my keyboard, anything that is not integer, such as decimals, letters, will result in an exception, which will cause the process (thread, in Java) to exit. Even if you input an integer, it may be also undesirable. According to the demands, only numbers from 0 to 9 can be used. So we have to add a limit to the input, which is achieved by IF statement.

III. RESULTS AND CONCLUSIONS

III.1 Environment

- Windows 8
- NetBeans IDE 7.3

III.2 Screenshots of the result

Use Command Line to compile and execute the program in Figure 1.

III.3 Thoughts

This is my first Java program. I find it not hard after I know the differences between Java and C++. I think Java has many things such as different classes that can make program even easier.