Practice Exercise #07: Reading Techniques

http://www.comp.nus.edu.sg/~cs1020/4_misc/practice.html

Reference: Week 2 OOP Part 1

Objectives:
To ensure that you know how to read inputs using any of the three methods which you will encounter in your take-home labs and sit-in labs.

Task statement:

There are various ways of parsing inputs. In this exercise, you will implement 3 common techniques:

1. Read an integer *n*, and then read *n* lines following it, each line containing some data.
2. Read until some special value or character is encountered (e.g. read until -1).
3. Read until the end of the file.

Write a program **Reading.java** that reads some input data in one of the following 3 formats.

Format 1: The first line of input contains the string "LIMIT". This means that the second line contains an integer *n*, which is the number of operations. The next *n* lines contain a string on each line, describing the operation to be performed.

Format 2: The first line of input contains the string "SENTINEL". This means that the subsequent lines contain a string on each line, describing the operation to be performed. The inputs end with a line containing the string "-1".

Format 3: The first line of input contains the string "EOF". This means that the subsequent lines contain a string on each line, describing the operation to be performed. You are to read until the end of file. (For interactive input on UNIX, the user enters **control-d** to terminate input.)

The string that describes the operation to be performed contains an operation (ADD, SUB, or MUL) followed by two integers *x* and *y*:

- ADD *x y*: Compute *x* + *y*
- SUB *x y*: Compute *x* – *y*
- MUL *x y*: Compute *x* * *y*

For each of the operations read, your program is to print out its result.

You should have a method

operate(String op, int operand1, int operand2)

in your program to perform the required computations.

Sample run #1:

```
LIMIT
2
ADD 14 32
46
MUL -6 20
-120
```

Sample run #2:

```
SENTINEL
MUL 15 4
60
ADD -17 30
13
-1
```

Sample run #3:

```
EOF
SUB 1234 5678
-4444
SUB -2 -300
298
SUB 99999 0
99999
(user pressed control-d here)
```