PROGRAM NAME: SummarizeFinances

AUTHORS: Brian Chang & Zahmayne Lindsay

PURPOSE: To examine the files within the Finances directory and make a summary of each year

REQUIRED PACKAGES: The program imports the following packages:

import java.util.Scanner

import java.util.Arrays

import java.util.regex.Pattern

import java.util.regex.Matcher

import java.io.IOException

import java.io.File

import java.io.PrintWriter

CLASSES: This program contains only one class, called SummarizeFinances

INPUT FORMAT: The program will read files from the Finances folder. For example, it will read the txt file 1-2013 which contains:

Day, Item, Cost

0, Rent, -$550.00

0, Utilities, -$30.00

0, Work Monthly Pay, +$1134.09

1, H&M, -$4.12

5, Ten Ren Bubble Tea, -$32.67

8, Grandpa, +$90.14

9, Triple Os, -$22.70

10, Triple Os, -$34.23

14, Tim Hortons, -$6.09

17, Triple Os, -$15.23

21, Tim Hortons, -$12.04

24, Roots Cafe, -$24.72

28, Roots Cafe, -$27.56

OUTPUT FORMAT: The output will display the net earning/spending in different months as follows

```
Month      Net Earning/Spending

==========================

January        $464.87

October        $303.14

February       $690.59

May          $1442.67

September      $827.75
```

LIMITATIONS:

- The program can only process files that exist between 2013-2015
- The program can only process files that have file name in the format of “MM-YYYY.txt” or “M-YYYY.txt”

IMPROVEMENTS: The program could be improved by aligning the net earnings in a better way

BUGS:

- The program is not able to print out the months from October to December in order. It will insert these months before February and after January

ALGORITHM:

```
1) Create an array of Files of the files we will be reading
2) Create a new folder “Summary” in the current folder to store our output files
3) Create File objects for each of the output files and point them to the non-existent yet files
4) If we successfully create the “Summary” folder, do the following:
     I.   Create a PrintWriter for each of the file objects we created in (3)
     II.  Write in the headers for each of the files
     III. For each of the files in the array we created in (1)
             • Extract the year of the file we’re reading
             • Get the correct output file
             • Extract the month of the file we’re reading
             • Calculate the transactions in the file we’re reading
             • Write to the output file the month, and the sum of the transactions
5) Close the PrintWriters
```