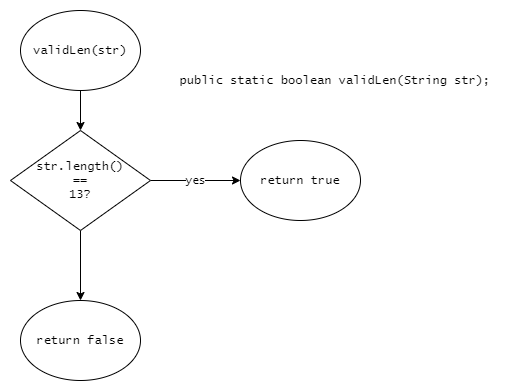
**CST-105: Exercise 7 - Input Validation**

The following exercise assesses your ability to do the following:

* Utilize abstraction to deconstruct a complex problem into individual tasks.
* Utilize methods in a programming solution
* Define methods using appropriate Java syntax

1. Read this document carefully and review the rubric for this assignment before beginning work. Be sure you are familiar with the criteria for successful completion. The rubric link can be found in the digital classroom under the assignment.
2. In this assignment you are designing the following methods by creating flowcharts and writing the method signature. You are not writing code for this assignment.
   1. A method that takes a String and returns true if the String has 13 digits and false otherwise.
   2. A method that takes a String of 13 digits creates and returns an array of ints with 13 positions. The array should contain the integer equivalents of the digits in the original string.
   3. A method that takes an array of 13 ints and returns an array of 13 ints that is computed as follows:
      1. Digits at even index locations in the original array are copied directly to the return array at corresponding positions
      2. Digits at odd index locations in the original array are multiplied by 3 and copied to the return array at corresponding positions.  
           
         ,8,3,6,6,3,9,6,9,5,24,1}
   4. A method that takes an array of 13 ints and calculates and returns the sum of the first 12 ints in the array
   5. A method that takes the sum calculated in part d and the last value in the integer array and returns
      1. True if the sum and the last integer are both 0
      2. True if (10 – (sum%10)) is equal to the last integer
      3. False otherwise  
           
         false  
         true
   6. Create a flowchart for a solution that:
      1. reads in strings of digits from a file
      2. calls the methods you’ve designed to determine whether or not each string of digits has the desired property
      3. outputs each string of digits that satisfies the property, one per line.
3. Create a video in which you present your method signatures and flowcharts. Videos over 5 minutes in length will not be accepted.
4. Submit the following:
5. A .png file containing the requested method signatures and flowcharts
6. A link to your video

  
Sample method signature and flowchart for method a:

TestA(str): boolean

Result = false

If str.length()=13

Result = true

Else

Result = false

Return Result

TestB(str): integer array

Result = true

Define array[13];integer

If(TestA(str))

For I = 0, I < str.length(), I = I + 1

Try

{

Array[I] = str.subString(I,I+1).parseInt()

}

Catch(exception e)

{

Result = false

}

Else

{

Result = false

}

If(result == false)

Make the array all -1

Return the array

MethodC (array:integer) : integer arrayCopy

Define arrayCopy[13]:integer

For I = 0, I < array.length(), I = I + I

If((I%2==0)

{

arrayCopy[I] = array[I]

}

Else

arrayCopy[I] = array[I]\*3

return arrayCopy[]

MethodD(array:integer) : integer sum

Define sum : integer

For I = 0, I < array.length() -1, I=I+1

Sum = sum + array[I]

Return sum

TestE(sum:integer, digit:integer) : Boolean Result

Result = false

If(sum == 0 and digit == 0)

Result = true

If((10-(sum%10)) == digit)

Result = true

Return Result

Program

Define a file path

Open a file

While there is a string

Str = reads in a strings of digits from a file

// calls the methods you’ve designed to determine whether or not each string of digits has the desired property

Result = TestA(str): Boolean

If (Result == true)

Array = TestB(Str): integer array

If(Array[1] != -1)

MethodC (array:integer) : integer arrayCopy

Sum = MethodD(array:integer) : integer sum

Result = TestE(sum:integer, array[array.length()-1] : integer)Boolean Result

Else

Result = false

If(Result == true)

outputs each string of digits that satisfies the property, one per line.