Multiplication of two numbers

Author: Azahar uddin

Problem Statement: Multiply two numbers. Assume each can have 30-40 digits. Numbers will be positive.

Solution:

I have enhanced the solution such that each number can have number of digits <= integer max limit.

This program will calculate multiplication of two numbers. Two numbers can have N number of digits where N <= integer max limit. It means number of digit in the number should be less than or equal to Integer.MAX_VALUE.

Program has been tested for numbers who has 100 digits.

E.g.

Number 1=
99977777777777777777777779999999999999999999999999999999999999999999999999999999999999999999999999999999999

Number 2 =

99977777777777777777777779999999999999999999999999999999999999999999999999999999999999999999999999999999999

But it will work for more than 100 and less than or equal to Integer.MAX_VALUE.

Bare minimum number of list has been used and they are cleaned and reused to save memory space.

You can provide numbers in file folder:

number1.txt and number2.txt
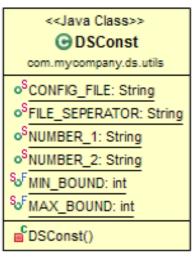
Folder structure:

- bin : It has runnable jar of the game: multiply.jar
- config: It has config.properties file to provide path of files having numbers. Do not change the property. Just change its value if you want.
- file: It has files which will have numbers.
- doc: It has documentation.
- src: Folder having source code. Comments have been provided in codes. If any portion is not
- clear please let me know.
- Log4j: Log4j plugin xml configuration file
- multiply.bat: Run this batch to multiply two numbers given in files. Before running this batch file please read
  readme.txt.
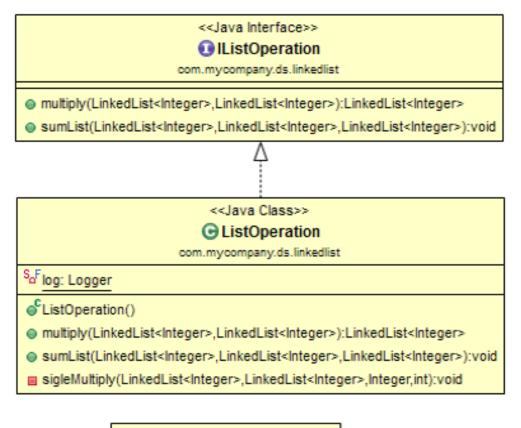- readme.txt: Please read this file before launching multiply.bat.

Program can be enhanced to support multiplications of numbers which can have any number of digits.
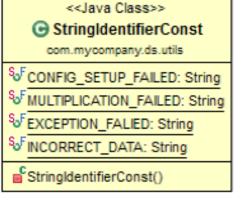
For that each input has to be broken into multiple linked list and then process them accordingly just like big data system. Map, calculate and then reduce. Please let me know to clarify what approach it means.
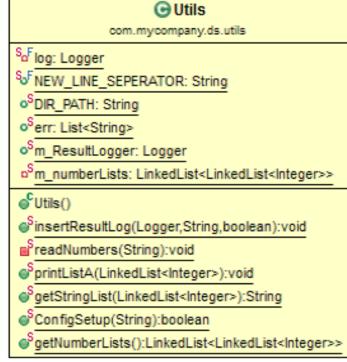
UML:

MultiplicationDriver
<<Java Class>>
com.mycompany.ds.multiplication.driver

- S,F log: Logger

- C MultiplicationDriver()
- S main(String[]):void

IListOperation
<<Java Interface>>
com.mycompany.ds.linkedlist

- multiply(LinkedList<Integer>,LinkedList<Integer>):LinkedList<Integer>
- sumList(LinkedList<Integer>,LinkedList<Integer>,LinkedList<Integer>):void

ListOperation
<<Java Class>>
com.mycompany.ds.linkedlist

- S,F log: Logger

- C ListOperation()
- multiply(LinkedList<Integer>,LinkedList<Integer>):LinkedList<Integer>
- sumList(LinkedList<Integer>,LinkedList<Integer>,LinkedList<Integer>):void
- sigleMultiply(LinkedList<Integer>,LinkedList<Integer>,Integer,int):void

DSConst
<<Java Class>>
com.mycompany.ds.utils

- S CONFIG_FILE: String
- S FILE_SEPERATOR: String
- S NUMBER_1: String
- S NUMBER_2: String
- S,F MIN_BOUND: int
- S,F MAX_BOUND: int

- C DSConst()

StringIdentifierConst
<<Java Class>>
com.mycompany.ds.utils

- S,F CONFIG_SETUP_FAILED: String
- S,F MULTIPLICATION_FAILED: String
- S,F EXCEPTION_FALIED: String
- S,F INCORRECT_DATA: String

- C StringIdentifierConst()

Utils
<<Java Class>>
com.mycompany.ds.utils

- S,F log: Logger
- S,F NEW_LINE_SEPERATOR: String
- S DIR_PATH: String
- S err: List<String>
- S m_ResultLogger: Logger
- S m_numberLists: LinkedList<LinkedList<Integer>>

- C Utils()
- S insertResultLog(Logger,String,boolean):void
- S readNumbers(String):void
- S printListA(LinkedList<Integer>):void
- S getStringList(LinkedList<Integer>):String
- S ConfigSetup(String):boolean
- S getNumberLists():LinkedList<LinkedList<Integer>>