Problem Statement: -

Check Writer: To print the given integer / float number available on Check in English equivalent word.

Ex:

Input: 10985.25 Output: Ten thousand nine hundred eighty-five dollars and 25/100

Input: 125.75 Output: One hundred twenty-five dollars and 75/100

Input: 95.00 Output: Ninety-five dollars only

Input: 69 Output: Sixty-nine dollars only

Approach Used:

I have used Java 1.8 language & some String, Stringbuilder class functions to solve this problem statement.

The steps are as follows:

1. Declared ‘number’ variable to get the input number (positive integers or float number) in string

format.

2. For ex. We get the input as 10985.12 in string form and it would be stored in **‘number’** named

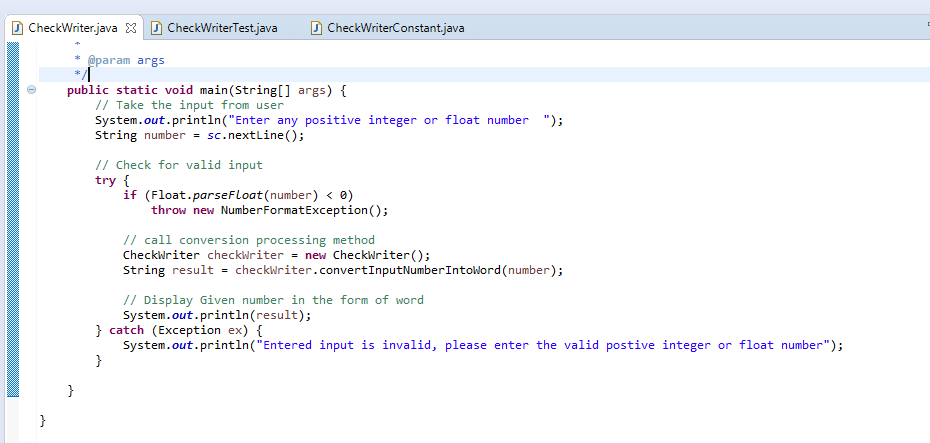
String variable.

3. Then we validated this number to check whether it is a valid float number or not.

If it is a valid number, then **convertInputNumberIntoWord (number)** method will get called. Else

**‘Entered input is invalid, please enter the valid positive integer or float number’** message would be

Displayed.

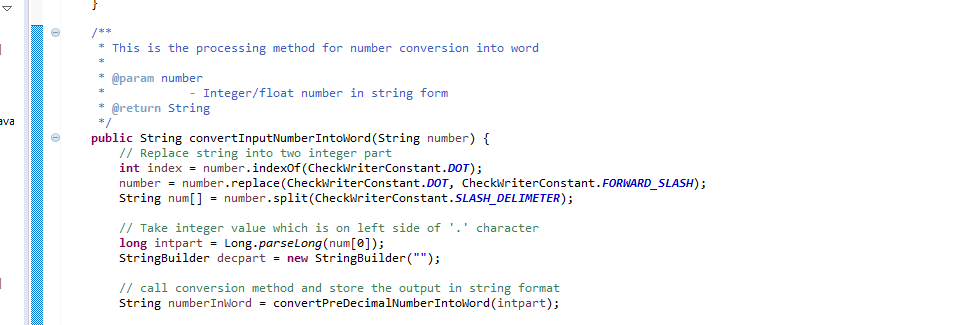


4. Then in **convertInputNumberIntoWord (number)** method, this number is separated into two parts –

one with the number preceding the decimal point and the other with the decimal part.

e.g. if number is 10985.12 then it would be separated into 10985 and 12. For pre-decimal part

***convertPreDecimalNumberIntoWord (intpart)***method is called for conversion.



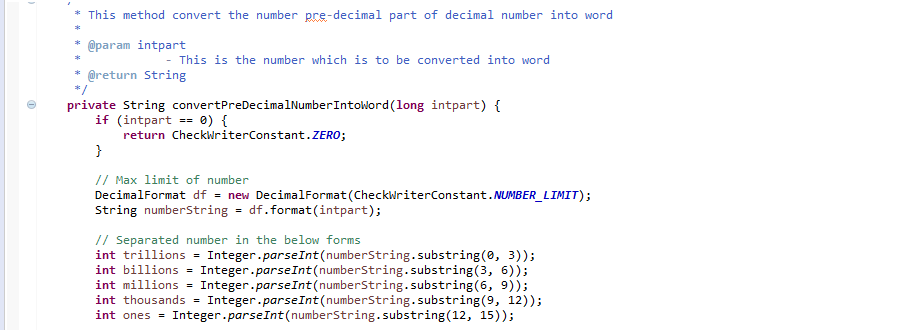
5. In ***convertPreDecimalNumberIntoWord* (long intpart)**method this number would be changed to a

fifteen-digit number by adding the appropriate number of zeros in front of it.

For example: - “56” would be converted to “000000000000056”

Then it would be separated in trillions, billions, millions, thousands, ones by Integer.parseInt() and

subString () method.



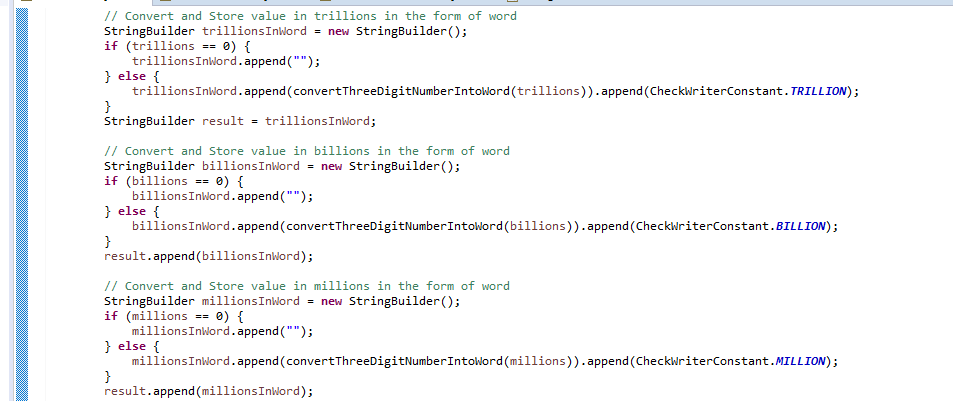
6. Now all trillions, billions, millions, thousands values are three-digit numeric values, so to convert

these numeric value into words, the **convertThreeDigitNumberIntoWord (number)** method gets

called.

Initially **convertThreeDigitNumberIntoWord (number)** method will be called by passing the trillions value inside the parameter. The result of this will be a work stored in “**trillionsInWord**” variable. Then a StringBuilder variable is used “result” to hold the value of the trillion word.

After that the same procedure is followed for billion, million, thousand and ones – all the while storing the word in the variable “result”.



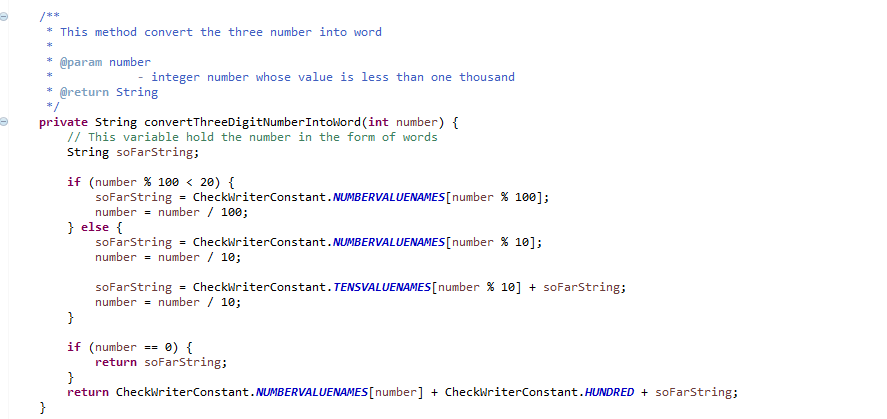
7. The functionality of the method **convertThreeDigitNumberIntoWord (number)** used in step 6 is as

follows:

In this method, three digits value of trillion, billion, million, thousand, ones will be converted into

word and the return value of the method is passed **to convertPreDecimalNumberIntoWord (long**

**intpart)**



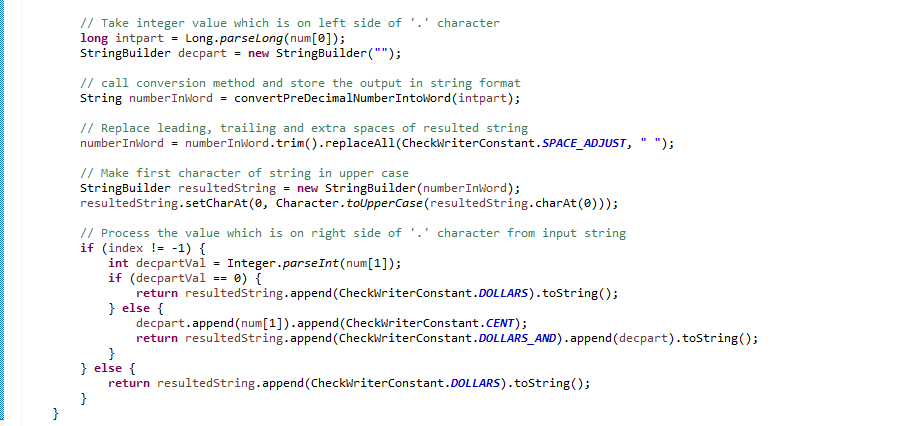
8. Then the Final result will be returned by ***convertPreDecimalNumberIntoWord (long intpart)*** & it will

be stored in “**numberInWord**” named variable in **convertInputNumberIntoWord (String number)**

method. Then after extra leading & trailing spaces will be removed.

9. Now the first letter of the resulted string would be converted in upper case and after some processing

the result will be returned back main () method as per the desired requirement.



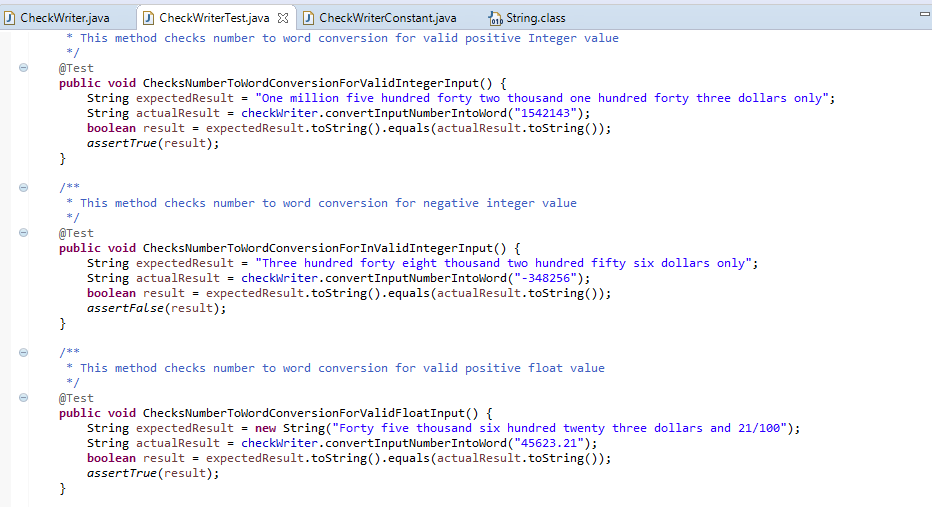
11. Finally the result will be displayed in main method.

Test Scenarios Tested:

Test valid conversion of Integer Number into word when user input is valid

Test valid conversion of Integer Number into word when user input is invalid

Test valid conversion of Float Number into word when user input is valid

Test valid conversion of Float Number into word when user input is invalid 

Reason to use this approach: I have written the code in Java 1.8. I think there is more reusability of code written and I also found it is very easy to understand and any positive Integer/float number can be displayed in word.

Flow Diagram

Start

Enter Input Number

Call convertInputNumberIntoWord () method

Take integer value which is on left side of '.' character and call convertPreDecimalNumberIntoWord () method for left site integer value

Split the string into parts and convert it into two integer number on the basis of ‘.’ character

**YES**

**No**

Display error Message

Check if the Input Number is valid or not

Rotate the numbers from 0 to input number within the matrix

Segregate this integer number into three digit trillions, billions, millions, thousands, ones

Convert these three digits trillion, billions, millions, thousands and ones value into word by calling convertThreeDigitNumberIntoWord () method

Stop

Result will be displayed in main method

resultedString value will be returned to main method by appending some string value

‘result’ variable value will be returned and stored into ‘numberInWord’ for removing leading, trailing and extra spaces and it will be stored into resultedString

Values returned by convertThreeDigitNumberIntoWord ()

is stored into ‘result’ variable