Name: Sufiyan Memon

- ## __StringCalculator__

public class StringCalculator {

```
    private float result;

    private String customDelimiter;


    private static final String DEFAULT_DELIMITER = ",";

    private static final String NEWLINE = "\n";

    private static final String CUSTOM_DELIMITER_PREFIX = "/";

    private static final String CUSTOM_DELIMITER_SUFFIX = NEWLINE;


    StringCalculator() {
        result = 0;
        customDelimiter = "";
    }


    public String sum(String numbers) {
        if (numbers.isEmpty())
            return String.format("%.0f", result);


        if (isInvalidLastCharacterIn(numbers))
            return "Number expected but EOF found.";


        if (numbers.startsWith(CUSTOM_DELIMITER_PREFIX))
            numbers = setCustomDelimiter(numbers);


        if (isNewlineAtInvalidPositionIn(numbers))
            return String.format("Number expected but '\n' found at position %d.",
numbers.lastIndexOf('\n'));


        if (containsNegative(numbers).length() > 0)
```

```java
        return String.format("Negative not allowed: %s", containsNegative(numbers));


    calculateSumOf(getStringArray(numbers));


    return hasDecimalPlaces() ? printFloat() : printInteger();
}


private boolean isInvalidLastCharacterIn(String numbers) {
    return Character.digit(numbers.charAt(numbers.length() - 1), 10) < 0;
}


private boolean isNewlineAtInvalidPositionIn(String numbers) {
    return numbers.lastIndexOf(NEWLINE) > numbers.lastIndexOf(DEFAULT_DELIMITER);
}


private StringBuilder containsNegative(String numbers) {
    StringBuilder negativeNumbers = new StringBuilder();


    for (String number : getStringArray(numbers))
        if (Float.valueOf(number) < 0) negativeNumbers.append(number + ",");


    boolean commaIsLastChar = negativeNumbers.length() > 0 &&
negativeNumbers.charAt(negativeNumbers.length() -1) == ',';


    return commaIsLastChar ? negativeNumbers.deleteCharAt(negativeNumbers.length() - 1)
                : negativeNumbers;
}


private String setCustomDelimiter(String numbers) {
    int customDelimiterStart = numbers.lastIndexOf(CUSTOM_DELIMITER_PREFIX) + 1;
    int customDelimiterEnd = numbers.indexOf(CUSTOM_DELIMITER_SUFFIX);
```

```
        customDelimiter = numbers.substring(customDelimiterStart, customDelimiterEnd);


        return numbers.substring(customDelimiterEnd + 1).replace(customDelimiter,
DEFAULT_DELIMITER);
    }


    private String[] getStringArray(String numbers) {
        return numbers.replace(NEWLINE, DEFAULT_DELIMITER).split(DEFAULT_DELIMITER);
    }


    private void calculateSumOf(String[] numbers) {
        for (String number : numbers)
            result = Float.sum(result, Float.parseFloat(number));
    }


    private boolean hasDecimalPlaces() {
        return result % 1 != 0;
    }


    private String printFloat() {
        return Float.toString((float) (Math.round(result * 100.0) / 100.0));
    }


    private String printInteger() {
        return String.valueOf((int) result);
    }
}
```