Package stringCalculatorTDD;

Import java.util.ArrayList;

Import java.util.regex.Matcher;

Import java.util.regex.Pattern;

Public class Calculator {

Public static int add(String numbers) {

String[] num = splitter(numbers);

Int size=num.length;

throwExceptionIfAnyNegative(num, size);

return findSum(num, size);

}

Private static void throwExceptionIfAnyNegative(String[] num, int size) {

ArrayList<String> negative = new ArrayList<String>();

For(int i=0; i<size; i++) {

If(toInt(num[i])<0) {

Negative.add(num[i]);

}

}

If(negative.size()>0) {

Throw new RuntimeException(“negatives not allowed: “ + String.join(“, “,negative));

}

}

Private static int findSum(String[] num, int size) {

Int sum=0;

For(int i=0; i<size; i++){

Sum = sum + toInt(num[i]);

}

Return sum;

}

Private static String[] splitter(String numbers) {

If(numbers.isEmpty()) {

Return new String[0];

} else if(isCustomDelimiterString(numbers)) {

Return splitUsingCustomDelimiter(numbers);

}

Return splitUsingCommaAndNewLine(numbers);

}

Private static boolean isCustomDelimiterString(String numbers) {

Return numbers.startsWith(“//”);

}

Private static String[] splitUsingCommaAndNewLine(String numbers) {

String[] num=numbers.split(“,|\n”);

Return num;

}

Private static String[] splitUsingCustomDelimiter(String numbers) {

Matcher m = Pattern.compile(“//(.)\n(.\*)”).matcher(numbers);

m.matches();

String customDelim = m.group(1);

String num=m.group(2);

Return num.split(Pattern.quote(customDelim));

}

Private static int toInt(String numbers) {

Return Integer.parseInt(numbers);

}

}