Java Varargs - Simple Addition

by aa1992

Problem | Submissions | Leaderboard | Discussions | Editorial

You are given a class *Solution* and its *main* method in the editor.
Your task is to create the class *Add* and the required methods so that the code prints the *sum of the numbers* passed to the function *add*.

Note: Your *add* method in the *Add* class must print the *sum* as given in the *Sample Output*

Input Format

There are six lines of input, each containing an integer.

Output Format

There will be only four lines of output. Each line contains the sum of the *integers* passed as the parameters to *add* in the *main* method.

Sample Input

```
1
2
3
4
5
6
```

Sample Output

```
1+2=3
1+2+3=6
1+2+3+4+5=15
1+2+3+4+5+6=21
```

Current Buffer (saved locally, editable)

Java 7

```
import java.io.*;
import java.lang.reflect.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

class Add {
    void add(int... nums) {
        int sum = 0, i = 0;
        for (int n : nums) {
            sum = sum + n;
            if (i++ != nums.length - 1)
                System.out.print(n + "+");
            else
                System.out.print(n + "=");
        }

        System.out.println(sum);
    }
}


public class Solution {

    public static void main(String[] args) {
       try{
            BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
            int n1=Integer.parseInt(br.readLine());
            int n2=Integer.parseInt(br.readLine());
            int n3=Integer.parseInt(br.readLine());
            int n4=Integer.parseInt(br.readLine());
            int n5=Integer.parseInt(br.readLine());
            int n6=Integer.parseInt(br.readLine());
            Add ob=new Add();
            ob.add(n1,n2);
            ob.add(n1,n2,n3);
            ob.add(n1,n2,n3,n4,n5);
            ob.add(n1,n2,n3,n4,n5,n6);
            Method[] methods=Add.class.getDeclaredMethods();
            Set<String> set=new HashSet<>();
            boolean overload=false;
            for(int i=0;i<methods.length;i++)
            {
                if(set.contains(methods[i].getName()))
                {
                    overload=true;
                    break;
                }
                set.add(methods[i].getName());

            }
            if(overload)
            {
                throw new Exception("Overloading not allowed");
            }
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
        }



}
```

```
70
```

Line: 22 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code

Processing

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4