Summation for Wrapper Types (Integer, Long, Double, etc.)

What is Summation? Addition of sequence results in their sum or total. That sum is Summation of the sequence.

So question is simple. Scan the list for the elements and add them. Now return the sum.

Let us first design a simple solution.

1. Get an Iterator.
2. Check if next value is not null (Wrapper types allow nulls).
3. If not null then add the value.
4. Return the value.

Now we will design this solution for List<Integer>.

**public** **static** **int** sumInteger(**final** List<Integer> list) {

**int** sum = 0;

Iterator<Integer> iterator = list.iterator();

**while**(iterator.hasNext()){

Integer val = iterator.next();

**if** (val != **null**) {

sum = sum + val;

}

}

**return** sum;

}

The above solution works well. Now think that you want to add contents of Double list or Long lists then you need to duplicate the code. See the below code.

**public** **static** **double** sumDouble(List<Double> list) {

**double** sum = 0;

Iterator<Double> iterator = list.iterator();

**while**(iterator.hasNext()) {

Double val = iterator.next();

**if** (val != **null**) {

sum = sum + val;

}

}

**return** sum;

}

**public** **static** **long** sumLong(List<Long> list) {

**long** sum = 0;

Iterator<Long> iterator = list.iterator();

**while**(iterator.hasNext()) {

Long val = iterator.next();

**if** (val != **null**) {

sum = sum + val.longValue();

}

}

**return** sum;

}

The thing is this is we are duplicating code which is bad thing. Now let us design a better solution.

We can leverage the concept of Java Generics and make sure that it works properly. Refer the below code.

**public** **static** **double** sum(List<? **extends** Number> list) {

BigDecimal sum = **new** BigDecimal("0.0");

Iterator<? **extends** Number> iterator = list.iterator();

**while**(iterator.hasNext()) {

Number val = iterator.next();

**if** (val != **null**) {

sum = sum.add(**new** BigDecimal(val.toString()));

}

}

**return** sum.doubleValue();

}

The above code will work for List<Integer>, List<Long>, List<Double>, List<Byte>, List<Short> and List<Float>. And why will it work? Because the wrapper classes “extends Number” class. This concept can be clearer after reading this [post](http://data-structure-learning.blogspot.com/2015/05/difference-between-arraylist-and-arrays.html).

List<Integer> ints = Arrays.*asList*(**null**, 2, 3, 4, 5);

List<Long> longs = Arrays.*asList*(1l, 2l, 6l, 7l);

List<Double> doubles = Arrays.*asList*(1d, 2d, 3d, 4d, 5d, 6d);

List<? **extends** Number> numbers = Arrays.*asList*(1, 4f, 5d);

System.***out***.println(*sum*(ints));

System.***out***.println(*sum*(longs));

System.***out***.println(*sum*(doubles));

System.***out***.println(*sum*(numbers));

Output

Integer List::14.0

Long List:: 16.0

Double List:: 21.0

? extends Number List:: 10.0