Calculate average of Sequence.

To calculate average we take Summation or Mean of the sequence. I have discussed the topic of Summation in previous post. Click [here](http://data-structure-learning.blogspot.com/2015/05/summation-by-2-ways.html) to see how Summation is calculated. We will call that method directly so as to get Summation. For convenience I will Paste entire code for Summation class in this post too.

Let us work on this problem.

For given sequence of numbers we define Arithmetic Mean (average) as summation of all numbers in list divided by total number of elements.

In simple terms it means that add all elements (numbers) in array. Once you have sum divide it by length of array.

/\*\*

\* Takes sum for int[].

\* \*/

**public** **static** **int** sum(**int**[] a) {

**int** sum = 0;

**for** (**int** i = 0; i < a.length; i++) {

sum = sum + a[i];

}

**return** sum;

}

/\*\*

\* Takes mean for int[].

\* we return double as division may lead to double value.

\* \*/

**public** **static** **double** mean(**int**[] a) {

**double** length = a.length;

**return** *sum*(a) / length;

}

Now I will first write the code for average then at the end of the blog I will paste code for Summation class. You can also click [here](http://data-structure-learning.blogspot.com/2015/05/summation-by-2-ways.html) to see post on [Summation](http://data-structure-learning.blogspot.com/2015/05/summation-by-2-ways.html).

Below is the code for Mean class which calculates mean or Arithmetic mean or average of int[], long[] and double[]. Also, I have written code of how to calculate average in Java 8 by streams.

Below is code for Mean class.

**package** arrays;

**import** java.lang.reflect.Array;

**import** java.util.Arrays;

**import** java.util.OptionalDouble;

**public** **class** Mean {

/\*\*

\* Takes mean for int[].

\* we return double as division may lead to double value.

\* \*/

**public** **static** **double** mean(**int**[] a) {

*validate*(a);

**double** length = a.length;

**return** Summation.*sum*(a) / length;

}

/\*\*

\* Validate the array.

\* Array must not be null not size must not be 0

\* \*/

**public** **static** **void** validate(Object obj) {

**if** (obj == **null**) {

**throw** **new** IllegalArgumentException("Array should not be null");

} **else** **if** (Array.*getLength*(obj) == 0) {

**throw** **new** IllegalArgumentException("Array should not be empty");

}

}

/\*\*

\* takes mean for long[].

\* we return double as division may lead to double value.

\* \*/

**public** **static** **double** mean(**long**[] a) {

*validate*(a);

**double** length = a.length;

**return** Summation.*sum*(a) / length;

}

/\*\*

\* takes mean for double[].

\* we return double as division may lead to double value.

\* \*/

**public** **static** **double** mean(**double**[] a) {

*validate*(a);

**double** length = a.length;

**return** Summation.*sum*(a) / length;

}

**public** **static** **void** main(String[] args) {

**int**[] i = { 1, 2, 3, 4 };

**long**[] l = { 0, 1, 2, 3, 4 };

**double**[] d = { 0, 0, 0, 1, 2, 3, 4 };

System.***out***.println("mean int[] "+*mean*(i));

System.***out***.println("mean long[] "+*mean*(l));

System.***out***.println("mean double[] "+*mean*(d));

/\*\*

\* Java 8 average demo.

\* \*/

OptionalDouble average = Arrays.*stream*(i).average();

System.***out***.println("mean as stream int[] "+average.getAsDouble());

average = Arrays.*stream*(l).average();

System.***out***.println("mean as stream long[] "+average.getAsDouble());

average = Arrays.*stream*(d).average();

System.***out***.println("mean as stream double[] "+average.getAsDouble());

}

}

Output Mean class is as follows:

mean int[] 2.5

mean long[] 2.0

mean double[] 1.4285714285714286

mean as stream int[] 2.5

mean as stream long[] 2.0

mean as stream double[] 1.4285714285714286

//---------------------------------------------------------------------------

Below is program for Summation class.

**package** arrays;

**import** java.util.Arrays;

**public** **class** Summation {

**public** **static** **int** sum(**int**[] a) {

**int** sum = 0;

**for** (**int** i = 0; i < a.length; i++) {

sum = sum + a[i];

}

**return** sum;

}

**public** **static** **long** sum(**long**[] a) {

**long** sum = 0;

**for** (**int** i = 0; i < a.length; i++) {

sum = sum + a[i];

}

**return** sum;

}

**public** **static** **double** sum(**double**[] a) {

**double** sum = 0;

**for** (**int** i = 0; i < a.length; i++) {

sum = sum + a[i];

}

**return** sum;

}

**public** **static** **void** main(String[] args) {

**int**[] a = { 3, 2, 1, 4, 3, 2 };

**long**[] l = { 3, 2, 1, 4, 3, 2 };

**double**[] d = { 3, 2, 1, 4, 3, 2 };

System.***out***.println("int[] "+*sum*(a));

System.***out***.println("float[] "+*sum*(l));

System.***out***.println("double[] "+*sum*(d));

**int** sumInt=Arrays.*stream*(a).sum();

System.***out***.println("Stream int[] sum "+sumInt);

**long** sumLong=Arrays.*stream*(l).sum();

System.***out***.println("Stream long[] sum "+sumLong);

**double** sumDouble=Arrays.*stream*(d).sum();

System.***out***.println("Stream double[] sum "+sumDouble);

}

}

Output for Summation class is as follows:

int[] 15

float[] 15

double[] 15.0

Stream int[] sum 15

Stream long[] sum 15

Stream double[] sum 15.0