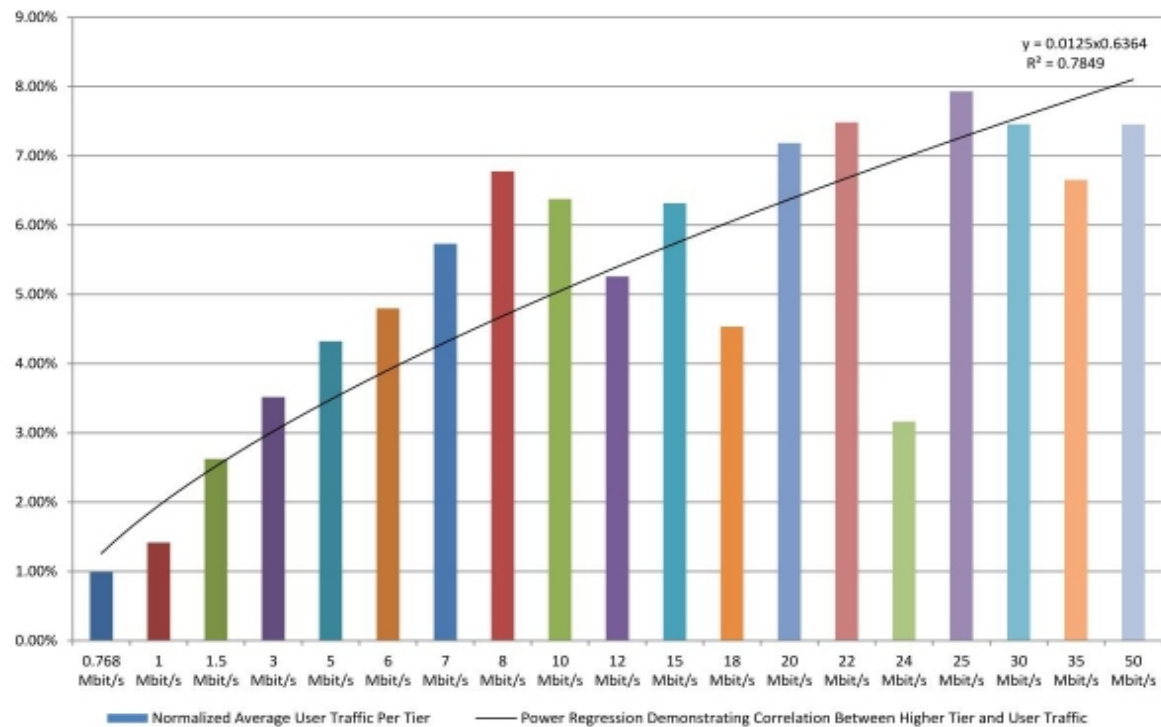


Long-term real growth in US Stocks (Log Scale)

Annual price index adjusted for inflation 1871–2010



Normalized Average User Traffic Per Tier



Average

When your 'average' joe says "average" they generally don't give any thought to there being several types of "averages".

That is because your average person is not a Mathematician or a Computer Scientist.

But alas... you are!!!



The Arithmetic Mean

When people say 'average', they are typically referring to the **Arithmetic Mean**... how do we find this?

$$\text{Mean} = \text{Sum} / \text{Number of Values}$$



Mean Mr. Mustard

vector artwork ©Hank Grebe

The Geometric Mean

The n th root (where n is the number of values) of the product of the numbers.

In other words... multiply them all together and take the n th root:

Geometric Mean (Three or More Numbers)

$$\text{Mean} = (a_1 \times a_2 \times \cdots a_n)^{1/n}$$

a = a number in the group

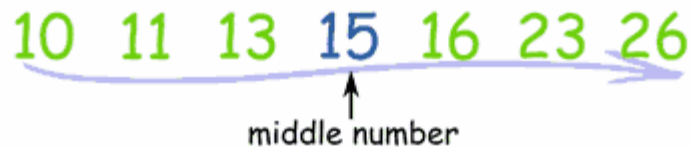
n = quantity of numbers in the group

Example:

$$\begin{aligned} a_1 &= 10 & M &= (a_1 \times a_2 \times \cdots a_n)^{1/n} \\ a_2 &= 15 & &= (10 \times 15 \times 20)^{1/3} \\ a_3 &= 20 & &= (3000)^{1/3} \\ & & &= \sqrt[3]{3000} \\ & & &= 14.42 \end{aligned}$$

Median

The median is the middle number of a set of numbers... the set of numbers must be sorted (in order):



When N is even (an even number of values) there will be two 'middle numbers'... in that case you take the average of the two 'middle numbers'.

Mode

Mr. A's super secret mnemonic for Mode:

MOde = Most Often

The mode is simply the Most Often occurring value in a set of numbers.

What if no value appears more than once?

What if there is more than one mode?

Max

You guessed it... the largest of a set of data.

Our approach:

- Set Max to the first value
- Iterate (loop) over the set of values
 - if the next value is larger
then it is the new max
- When the loop completes...
you've got your Max!
- Remember to return the value
(why not just print it out?)

Min

You guessed it... the smallest of a set of data.

Our approach again:

- Set Min to the first value
- Iterate (loop) over the set of values
 - if the next value is smaller
then it is the new min
- When the loop completes...
you've got your Min!
- Remember to return the value
(why not just print it out?)

Range

$$\text{Range} = \text{Max} - \text{Min}$$

Standard Deviation

Standard Deviation is the average amount that a number differs from the mean for a set of values.

SD = Square root of the sum of the square differences from the mean divided by the number of values.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

X	X - X'	(X-X') ²
10	5	25
2	-3	9
4	-1	1
5	0	0
6	1	1
7	2	4
3	-2	4
3	-2	4
Average	S (X-X')² =	48
5	SD = Sqrt (S (X-X')² /n) =	2.44949
N		
8		

$$48 / 8 = 6$$

$$\text{sqrt}(6) = 2.44949$$

Some Principles to Code By

You should not even think about writing code until you **THINK** about what you are going to do.

The more time you spend **planning** the better and more efficient your code will be.

Planning is not just a cute thing to do when you have the time or 'get around to it'.

My Gift to You...

NOW YOU HAVE NO EXCUSE!

TUIT

Some things to think about...

- How do we get user input for an unknown number of values?
- How would we know how many variables or what size to make the array?
- Would a String work for user input?
- Can we convert data from a String into several int's?
- Are there some built-in tools in Java that might help us?

parse

/pärs/ 🔊

verb

1. analyze (a sentence) into its parts and describe their syntactic roles.

noun

COMPUTING

1. an act of or the result obtained by parsing a string or a text.
-

parseInt

```
public static int parseInt(String s)
    throws NumberFormatException
```

Parses the string argument as a signed decimal integer. The characters in the string must all be decimal digits, except that the first character may be an ASCII minus sign '-' ('\u002D') to indicate a negative value or an ASCII plus sign '+' ('\u002B') to indicate a positive value. The resulting integer value is returned, exactly as if the argument and the radix 10 were given as arguments to the `parseInt(java.lang.String, int)` method.

Parameters:

`s` - a `String` containing the `int` representation to be parsed

Returns:

the integer value represented by the argument in decimal.

From the API... split!

Return value:

```
String[]
```

Method Header:

```
split(String regex)
```

Splits this string around matches of the given regular expression.

String Object -> split();

Integer Object -> parseInt();

```
public class SplitStringExample
{
    public static void main(String[] args)
    {
        String input = "34,55,66,35,67,98,77";

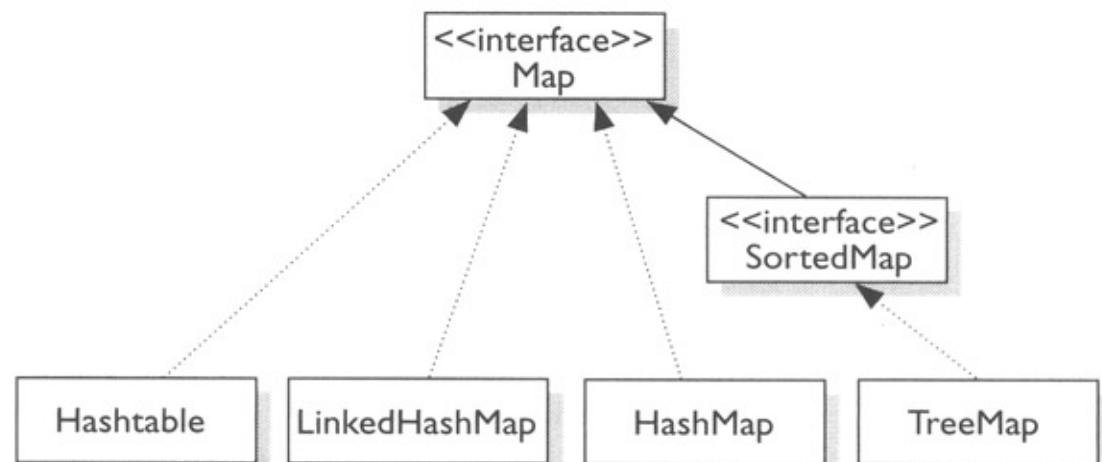
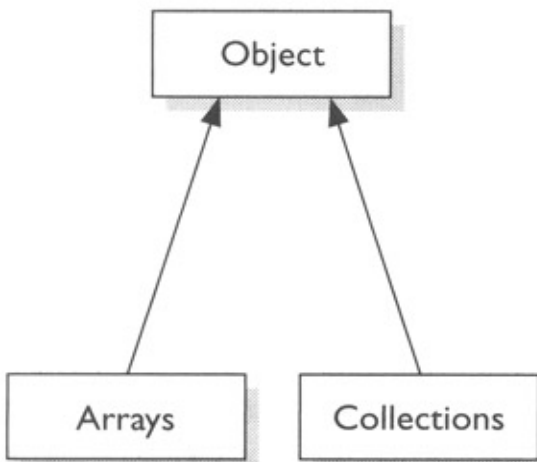
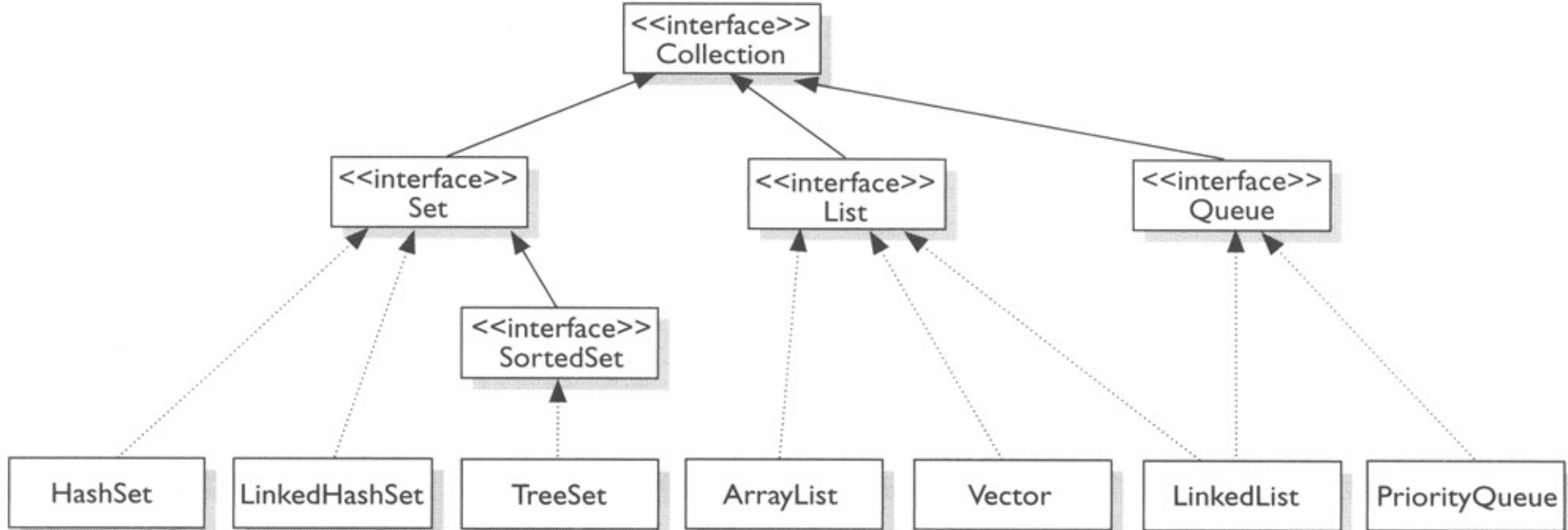
        String[] strValues = input.split(",");
        int[] intValues = new int[strValues.length];

        for(int i=0; i<strValues.length; i++)
        {
            intValues[i] = Integer.parseInt(strValues[i]);
            System.out.println(intValues[i]);
        }
    }
}
```

The Collections Framework

From the API:

The **collections framework** is a unified architecture for representing and manipulating collections, enabling them to be manipulated independently of the details of their representation. It reduces programming effort while increasing performance. It enables interoperability among unrelated APIs, reduces effort in designing and learning new APIs, and fosters software reuse. The framework is based on more than a dozen collection interfaces. It includes implementations of these interfaces and algorithms to manipulate them.



.....>
implements

————>
extends

Built In Sorting

```
public int calcMode(int[] nums)
{
    Arrays.sort(nums);
    int mode = -999, testMode;
    int occurs = 0, testOccur = 0;
```

Procedural Design

```
public class StatLibrary
{
    public static int getAvg(int[] nums)
    { // implementation not shown
    }
    public static int getMax(int[] nums)
    { // implementation not shown
    }
    public static int getMin(int[] nums)
    { // implementation not shown
    }
    // etc
}
```

```
public static void main(String[] args)
{
    String originalList = "99,33,53,46,74,78,53";
    int[] intNums = SuperStatLibrary.convertStringToIntArray(originalList);
    int min, max;
    double avg, stdDev;
    String mode;

    avg = SuperStatLibrary.getAvg(intNums);
    System.out.println("Average = " + avg);
    min = SuperStatLibrary.getMin(intNums);
    System.out.println("Min = " + min);
    max = SuperStatLibrary.getMax(intNums);
    System.out.println("Max = " + max);
    mode = SuperStatLibrary.getMode(intNums);
    System.out.println("Mode = " + mode);
}
```

Object Oriented Design

```
public class SuperStat
{
    private String[] strNums;
    private int[] intNums;

    public SuperStat()
    { // implementation not shown
    }
    public SuperStat(String originalNums)
    { // implementation not shown
    }
    public double getAvg()
    { // implementation not shown
    }
    public double getMax()
    { // implementation not shown
    }
    // etc
}
```

```
public class SuperStatClient
{
    public static void main(String[] args)
    {
        SuperStat nums = new SuperStat("12,40,50,55,33,23");

        System.out.println("Average = " + nums.getAvg());
        System.out.println("Max = " + nums.getMax());
        System.out.println("Min = " + nums.getMin());
        System.out.println("Range = " + nums.getRange());
        System.out.println("Mode = " + nums.getMode());
        System.out.println("Standard Deviation = " + nums.getStdDev());
    }
}
```

Teamwork

