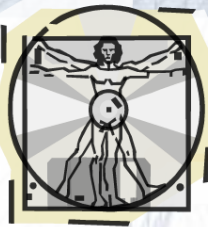


# Assignment 2

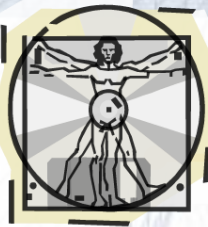




# Assignment 2

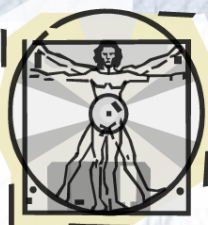
- Requirement-based testing
  - SUT - JFreeChart
  - Requirements
  - Test design
  - Test execution





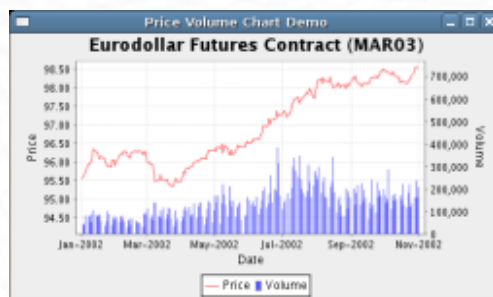
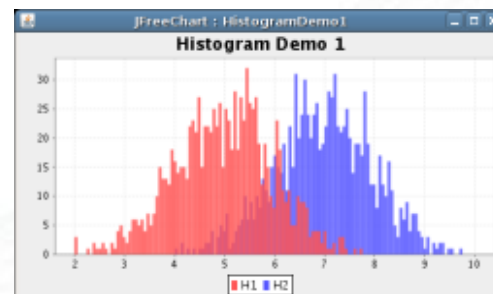
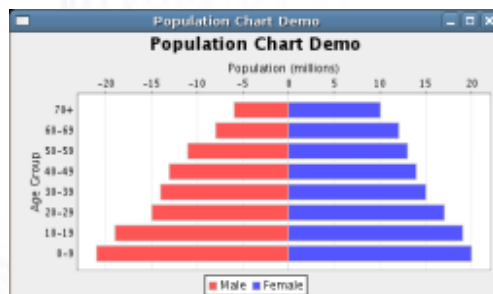
# Activities

- **1<sup>st</sup> week:** Familiarization (setup Eclipse, Junit, mock framework, ensure they work together)
- **1<sup>st</sup> week:** Writing tests (write test cases and suite for Junit, prepare stubs or mocks)
- **2<sup>nd</sup> week:** Executing unit test using Junit and demo
- **2<sup>nd</sup> week:** Reporting the results

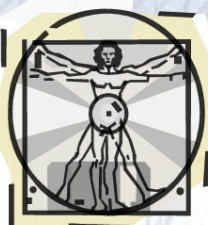


# SUT: JFreeChart

- The JFreeChart framework is a free/open-source chart library intended to be integrated into other systems as a quick and simple way to add charting functionality to Java applications



<http://www.jfree.org/jfreechart>

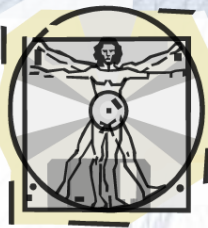


# JFreeChart – Requirements

- **Reminder:** SW Requirements can come in many different formats, e.g., plain English, UML models, Javadoc, ...
- In Assignment 2, we will use the Javadoc documentation of JFreeChart as its requirements
  - Online now

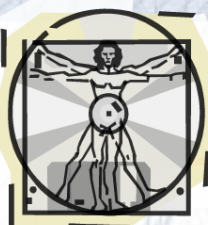
The screenshot shows the JFreeChart Javadoc documentation in a Mozilla Firefox browser window. The address bar shows the URL <http://136.159.106.66/JFreeChart/ModifiedJavadoc/>. The page title is "Overview - Mozilla Firefox". The browser window has several tabs open, including "Overview", "myUoFC", and "(0 unread) Rogers Yahoo! Mail, garusi". The page content includes a navigation menu on the left with links to "All Classes", "Packages", and "All Classes". The main content area shows the "Overview" page with a table of packages and their descriptions.

Package	Description
<a href="#">org.jfree.chart</a>	Core classes, including <a href="#">JFreeChart</a> and <a href="#">ChartPanel</a> .
<a href="#">org.jfree.chart.annotations</a>	A framework for adding annotations to charts.
<a href="#">org.jfree.chart.annotations.junit</a>	Tests for the classes in the <a href="#">org.jfree.chart.annotations</a> package.
<a href="#">org.jfree.chart.axis</a>	Axis classes and interfaces.
<a href="#">org.jfree.chart.axis.junit</a>	Tests for the axis classes and interfaces.
<a href="#">org.jfree.chart.block</a>	Blocks and layout classes used extensively by the <a href="#">LegendTitle</a> class.
<a href="#">org.jfree.chart.block.junit</a>	
<a href="#">org.jfree.chart.demo</a>	Some basic demos to get you started.



# Test Requirements

- JFreeChart classes under test
- `org.jfree.data.DataUtilities`
  - Has 9 methods
- `org.jfree.data.Range`
  - Has 19 methods
- You have to write tests for 5 methods from each class



# Test Design

- Equivalence class testing
- Boundary value analysis
- Robustness testing
- Worst case testing

## org.jfree.data Class DataUtilities

java.lang.Object  
└─org.jfree.data.DataUtilities

public abstract class **DataUtilities**  
extends java.lang.Object

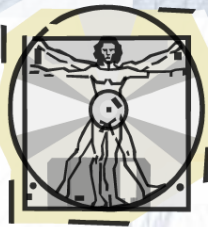
Utility methods for use with some of the data classes (but not the datasets, see [DatasetUtilities](#)).

### Constructor Summary

[DataUtilities](#)()

### Method Summary

static double	<a href="#">calculateColumnTotal</a> ( <a href="#">Values2D</a> data, int column)	Returns the sum of the values in one column of the supplied data table.
static double	<a href="#">calculateRowTotal</a> ( <a href="#">Values2D</a> data, int row)	Returns the sum of the values in one row of the supplied data table.
static java.lang.Number[]	<a href="#">createNumberArray</a> (double[] data)	Constructs an array of Number objects from an array of double primitives.
static java.lang.Number[][]	<a href="#">createNumberArray2D</a> (double[][] data)	Constructs an array of arrays of Number objects from a corresponding structure.
static <a href="#">KeyedValues</a>	<a href="#">getCumulativePercentages</a> ( <a href="#">KeyedValues</a> data)	Returns a <a href="#">KeyedValues</a> instance that contains the cumulative percentage values.



# Test Execution

- Junit 4 or 5
- You need to use **mocking**
  - For “Values2D” and “KeyedValues”
- You are free to choose your mocking framework
- Example of jMock tests for the SUT of the Assignment 2 will be given
  - jMock or any other mocking framework