# Instructions

## Installation

The calculator has been implemented as a static library, so you only need to download MatrixCalc.java and import it into your project. It is not necessary, or possible, to create an instance of the class, as all methods can be called directly.

## Available operations

Publicly availabe methods are add, subtract, scale, multiply, determinant, invert, setStrassenCutoff and getStrassenCutoff. Two-dimensional double arrays are used to represent matrices. All operations will throw an IllegalArgumentException when a jagged or empty matrix is passed as a parameter. There might also be other requirements depending on the method. These are detailed below.

The add and subtract methods take two matrices of identical size and return a matrix where each cell contains the result of adding or subtracting the corresponding cells in the matrices given as parameters.

The scale method takes one matrix and a scalar of double-type and returns a matrix where each cell of the matrix has been multiplied by the scalar.

The multiply method takes two matrices and returns a matrix where each cell contains the dot-product of values taken from the corresponding row of the first matrix and the corresponding column of the second matrix. Because of this, the row length of the first matrix and the column length of the second matrix must be equal.

The multiply method uses both naive matrix multiplication and the Strassen-method to calculate the result. While the Strassen-method is faster multiplying large matrices, it's implementation requires some overhead, which makes it costlier for multiplying small matrices. Because of this, it is faster to switch to the naive method when multiplying smaller matrices.

The strassenCutoff variable is used to define when the program switches multiplication methods, and can be set using the setStrassenCutoff method. By default this is set to 257, which should be fine for most situations, but if you're doing many calculations and want optimal performance, it might be worth testing other values. The current strassenCutoff value can be checked with the getStrassenCutoff method.

The determinant method takes a single square matrix as its parameter and returns the determinant of said matrix as a double. If the given matrix is non-invertible, the method will not produce a result, and will return NaN instead.

The invert method takes a single square matrix as its parameter and returns the inverse of that matrix. Note that the method does not check whether the given matrix is invertible or not, so if presented with a non-invertible matrix, the result may contain NaN-values. To check if a matrix is invertible, you can calculate it's determinant. If the determinant does not equal 0, the matrix is invertible.