Calculator

This is a simple java application written to evaluate mathematical expressions.

Documentation

The `docs/` folder in this repository contains an extensive documentation of the libraries used by Calculator. You can start at `docs/index.html`, or go to `http://htmlpreview.github.io/?http://github.com/sahasatvik/Calculator/master/docs/index.html` to view the one hosted online.

Usage

Execution

Navigate to `bin/`, or add it to your classpath, then simply enter :

```
java Calculator
```

Compiling

Compile all `.java` source files in `src/` in their directory, or use `javac -d bin/` in order to keep binaries and sources separate. Documentation can be regenerated by using `javadoc -d docs/ [src/source/files.java] -author -version`.

Alternatively, you can simply use the shell scripts `build` and `buildDocs` if you are running a UNIX-like OS.

Manual

Arithmetic Expressions

Calculator can evaluate simple arithmetic expressions, using the operators (`+`, `-`, `*`, `/`, `^`(power)), as well as parenthesis (`(`, `)`). *Calculator* follows the BODMAS rule.

Examples

```
1 + 1               =>          2.0
1 * (2 + 3)         =>          5.0
10 * (64 ^ -0.5)
                    =>         1.25
```

Variables

Calculator can also store user-defined *variables*. A total of 32 variables can be stored in one runtime.

Syntax

```
var = value        >    assign 'value' to 'var'
<var>              >    <var> will be replaced
                        by its value.
```

Uses

```
x = 3              =>         3.0
y = <x> + 1        =>         4.0
(<x>^2 + <y>^2)^0.5
                   =>         5.0
```

Miscellaneous features

Nesting of assignments is also supported, as follows :

```
x = 1 + (y = 1)
                   =>         2.0
<x>                =>         2.0
<y>                =>         1.0
```

A special variable `<ans>` stores the previous expression. Thus, the following is valid :

```
1 * 2 * 3 * 4      =>        24.0
<ans> * 5          =>       120.0
```

Functions

Calculator supports the use of some basic *functions*.

Syntax

```
fnc[ value ]       >    evaluate 'fnc' of 'value'
```

Uses

```
sin[<pi> / 2]     =>        1.0
1 + abs[2 - 3]    =>        2.0
log[<e> ^ 3]      =>        3.0
```

Function	Value returned
`abs[x]`	absolute value of **`<x>`**
`exp[x]`	exponent of **`<x>`** (**`<e> ^ <x>`**)
`log[x]`	logarithm of **`<x>`** (base **`<e>`**)
`fct[x]` or `x!`	factorial of **`<x>`**
`deg[x]`	convert **`<x>`** to degrees from radians
`rad[x]`	convert **`<x>`** to radians from degrees
`sin[x]`, `cos[x]`, `tan[x]`, `csc[x]`, `sec[x]`, `ctn[x]`	trigonometric functions (**`<x>`** in radians)

Commands

Calculator interprets expressions starting with `/` as *commands*. These are special expressions which are not parsed as mathematical expressions, but as instructions to the *Calculator*.

Command	Purpose
`/help`	general help
`/help vars`	help on *Variables*
`/help funcs`	help on *Functions*
`/help cmds`	help on *Commands*
`/list vars`	list *Variables*
`/list funcs`	list *Functions*
`/list cmds` or `/list`	list *Commands*
`/exit`	exit *Calculator*