tVar

# Init

require("../tVar.lua")

matrix = require("../matrix")

# Global tVar.

numFormat = "%.3f"

mathEnviroment = "align"

debugMode = "off"

outputMode = "RES" --RES, RES\_EQ, RES\_EQ\_N

numeration = true

# New

tVar:New(0.04,"r\_{se}")

tVec:New({10,2,7},"v\_{1}")

tMat:New({{10,2,5},{2,4,3},{7,4,3}},"a\_{2}")

# Output

:print() -–abh. v OutputMode

:outRES\_EQ\_N(number[bool],enviroment[bool])

:outRES\_EQ([bool],[bool])

:outREs([bool],[bool])

:out() –-nur Wert

# Set [tVar]

:setName([string])

:setUnit([string])

:clean(name[string]) –-berechn. Schritte entf.

Tipp: können verkettet werden

# Misc

[tVar]:bracR() –-Runde Klammern

[tVar]:CRLF([string]) –-neuwline, [string] wird vor und nach Umbruch eingefügt

[tVar]:CRLFb([string]) –-Umbruch vor [tVar]

[tVar]:copy()

# Math

tVar.sqrt([tVar],[number])

tVar.PI

tVar.min(...)

tVar.max(...)

[tMat]:T() -–Transponieren

[tMat]:Det()

[tMat]:Inv()

[tVec]:crossP()

# Example

\begin{luacode\*}

require("../tVar.lua")

matrix = require("../matrix")

numFormat = "%.2f"

outputMode = "RES\_EQ\_N”

numeration = false

N\_D\_2=((-V\_LF \* (0.4\*d) + M\_LF + H\_LF \* H\_RB\_PL)/(R\_Hebel\_2)):CRLFb("="):setName("N\_{D,2}"):setUnit("\\kNpm"):print()

\end{luacode\*}