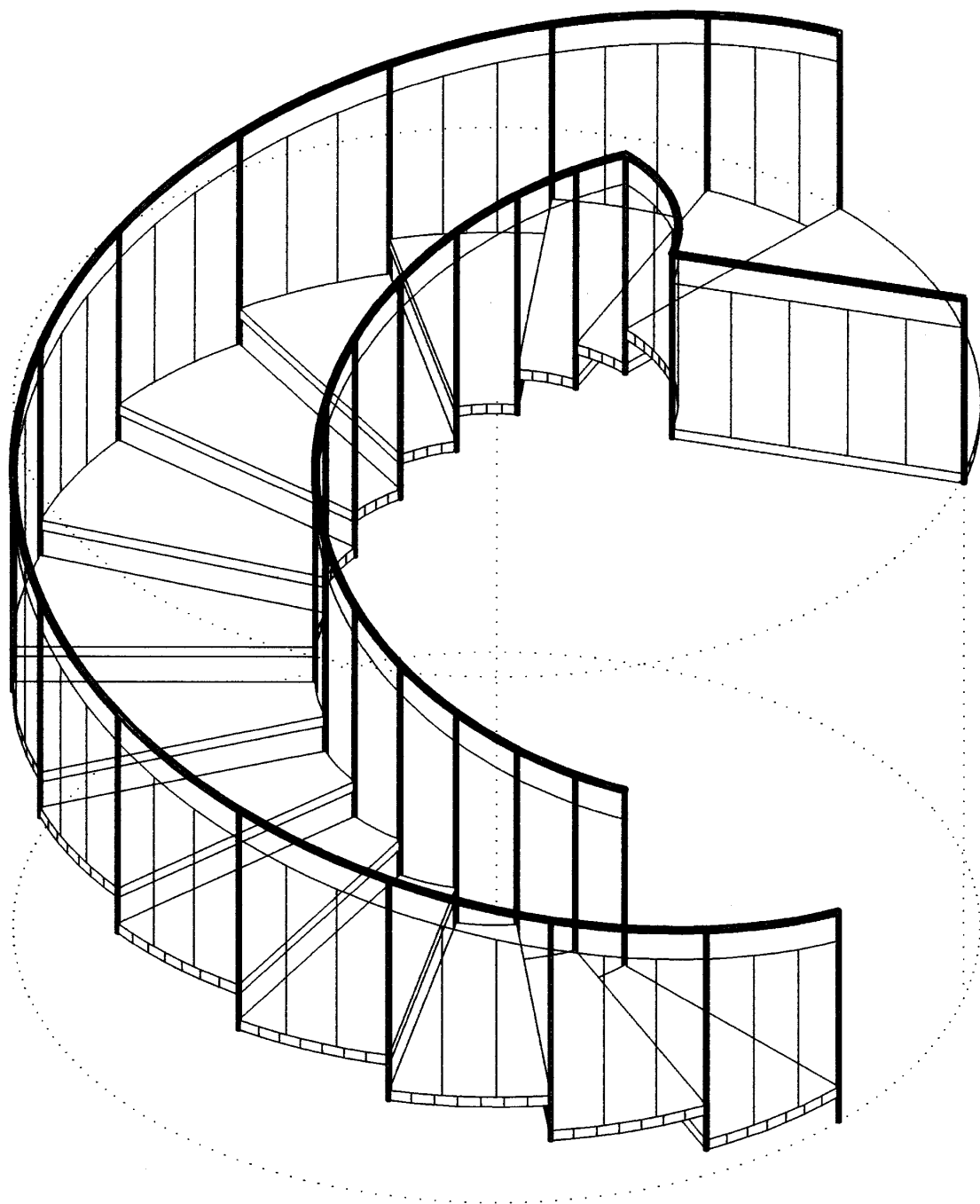


WENDELTREPPENMODUL für AutoCAD

programmiert in Autolisp

Rainer Kuhn Juni 1997



Datei: wt_info.txt

INFOS ZU WTREPPE.LSP (WENDELTRIPPENMODUL (c) 1997 Rainer Kuhn)

Das Programm WTREPPE erstellt nach Einstellungen und Eingabe von Vorgabewerten in einem eigenen Dialogfenster die Zeichnung einer Wendel- oder einer Spindeltreppe. Das Programm wurde in AUTOLISP als Zusatzfunktion fuer AUTOCAD Release 12 geschrieben. Weitere Erlaeuterungen zum Programm WTREPPE befinden sich in der Datei WT_HELP.HLP und koennen nach Programmaufruf ueber das "Hilfe"-Feld der Dialogbox aufgerufen werden.

Laden des Programms von der ACAD-Befehlszeile aus:

```
(load "wtreppe") <RETURN>
```

Starten des Programms von der ACAD-Befehlszeile aus:

```
wtreppe      <RETURN>
```

Die Programm benoetigt eine Bildschirmaufloesung von mindestens 1024x768 Pixel. Sollte der verwendete Bildschirm bzw. die Grafikkarte nur eine Aufloesung von 800x600 erreichen, so kann WTREPPE mit einer "komprimierten" Dialogbox folgendermassen gestartet werden:

Laden des Programms von der ACAD-Befehlszeile aus:

```
(load "800x600") <RETURN>
```

Starten des Programms von der ACAD-Befehlszeile aus:

```
wtreppe      <RETURN>
```

LISTE DER ZUGEHORIGEN DATEIEN:

wtreppe .lsp	= [AutoLISP-] Funktion WTREPPE : Wendeltreppenmodul
wt_dbox .dcl	= Dialogbox zu WTREPPE [Dialogue Control Language]
wt_help .hlp	= Hilfedatei zu WTREPPE [ASCII-Textdatei]
wt_help .hdx	= Hilfeindexdatei zu wt_help.hlp
wt_info .txt	= diese Infotextdatei zu WTREPPE [ASCII-Textdatei]
800x600 .lsp	= Funktion WTREPPE fuer Bildschirmaufloesung 800x600
800x600 .dcl	= Dialogbox zu WTREPPE fuer Bildschirmaufloesung 800x600

LISTE DER IN 'WTREPPE.LSP' VERWENDETEN VARIABLEN:

1) Zahlenwerte (Vorgaben/Berechnungen):

a	= Auftrittstiefe in der Lauflinie
anz	= Anzahl Steigungen (Treppe gesamt)
anz0	= vorlaeufige Anzahl Steigungen fuer Anpassung
anz1	= Anzahl Stufen (Treppe gesamt)
anz2	= Anzahl Steigungen zur Berechnung der Durchgangshoehe
d	= Aussendurchmesser Treppe
di	= Innendurchmesser Treppe (Spindel/Treppenaue)
gt	= Grundrissteilung
gt0	= vorlaeufige Grundrissteilung fuer Anpassung
h	= zu ueberwindende Hoehe (Gesamttreppe)
hd	= Durchgangshoehe unter Hinterkante Podest
hg	= Gelaenderhoehe
hm	= Hilfwert zur Berechnung des Stufenwinkels (geometrische Hoehe auf Mittelpunkt M)
hs	= Hoehe (Dicke) Stufen/Podest
hsp	= Hoehe Spindel
ls	= Stufenlaenge
ll	= Abstand Lauflinie zu AK Treppe
r	= Aussenradius Treppe
rg	= Radius Gelaender fuer Blockdefinition/Gelaender
ri	= Innenradius Treppe (Spindel/Treppenaue)
rl	= Radius Lauflinie
s	= Steigungshoehe
s0	= vorlaeufige Steigungshoehe fuer Anpassung
trm	= Treppenmass ($2 s + a$)

2) Zaehlvariablen und Status-/Kennwerte:

dcldat	= Kennwert fuer DCL-Datei [bei Ladefehler negativ]
ko	= Statuswert fuer Fehlerpruefung zur DBoxeingabe [nil oder 1]
nf	= Zaehlwert fuer Schleife Blockdefinition/3D-Flaechenbelegung
ng	= Zaehlwert fuer Schleife Blockdefinition/Gelaender
nt	= Zaehlwert fuer Schleife Treppenzeichnung/Blockeinfuegung
nvs	= Statuswert fuer DBox-Vorschau [nil oder 1]
ok	= Statuswert fuer DBox-Ausstieg [0, 1, 2 oder 3]

3) Koordinatenwerte:

x	= x-Wert Treppenmittelpunkt
y	= y-Wert Treppenmittelpunkt
z	= z-Wert Treppenmittelpunkt
zef	= z-Wert Einfuegepunkt fuer Treppenzeichnung/Blockeinfuegung
zgp	= z-Wert Gelaenderpunkt fuer Blockdefinition/Gelaender

4) Winkel:

wan	= Antrittswinkel [Bogenmass]
wang	= Antrittswinkel [Grad] (relativ, bezogen auf x-Achse / = VK erste Stufe)
waug	= Austrittswinkel [Grad] (relativ, bezogen auf x-Achse / = VK Podest)
wdhg	= Winkel zur Berechnung der Durchgangshoehe [Grad]
wef	= Einfuegewinkel fuer Blocks [Bogenmass]
wefg	= Einfuegewinkel fuer Blocks [Grad]

weig = eingeschlossener Winkel (absolut, positiv) [Grad]
 wgp = Winkel Gelaenderpunkt fuer Blockdefinition/Gelaender
 wp = Podestwinkel (absolut, positiv) [Bogenmass]
 wpg = Podestwinkel (absolut, positiv) [Grad]
 ws = Stufenwinkel (absolut, positiv) [Bogenmass]
 wsg = Stufenwinkel (absolut, positiv) [Grad]
 wsg0 = vorlaeufiger Stufenwinkel fuer Anpassung [Grad]
 w1,w2 = Winkel fuer Blockdefinition/Rahmenkonstruktion
 wf1,wf2 = Winkel fuer Blockdefinition/3D-Flaechenbelegung

5) Textvariablen/Einstellungen:

anp1 = Anpassungsoption 1 (Steigung ggf. steiler/flacher)
 anp2 = Anpassungsoption 2 (eing.Winkel fix/ggf.steiler/ggf.flacher)
 blnst = Blockname fuer Stufe
 blnpo = Blockname fuer Podest
 blname = Blockname fuer Blockdefinition (blnst/blnpo)
 blvar = Zustandsvariable fuer Blockdefinition (Stufe/Podest)
 lname = Layername fuer Treppenlayer
 lfarbe = Layerfarbe fuer Treppenlayer
 richt = Drehrichtung Treppe (linksdr./rechtsdr.)
 trtyp = Treppentyp (Wendeltr./Spindeltr.)
 vorg1 = Vorgabeoption 1 (eing.Wink./Gr.-Teilung/Treppenmass)
 vorg2 = Vorgabeoption 2 (Antritts-/Austritts-Winkel)
 warn = Inhalt des Warnmeldungsfeldes in DBox
 znart = Zeichnungsart (mit/ohne Gelaender)

6) Koordinatenpunkte:

EA1,EA2 = Eckpunkte aussen fuer Blockdefinition/Rahmenkonstruktion
 EAF1,EAF2 = Eckpunkte aussen fuer Blockdefinition/3D-Flaechenbelegung
 EI1,EI2 = Eckpunkte innen fuer Blockdefinition/Rahmenkonstruktion
 EIF1,EIF2 = Eckpunkte innen fuer Blockdefinition/3D-Flaechenbelegung
 GP1,GP2 = Gelaenderpunkte fuer Blockdefinition/Gelaender
 M = Mittelpunkt Treppe Basisgeschoss
 MGP = Mittelpunkt Gelaenderpunkte fuer Blockdefinition/Gelaender
 MO = Mittelpunkt Treppe Obergeschoss
 MS = Mittelpunkt Stufe fuer Treppenzeichnung/Blockeinfuegung
 MS0 = Mittelpunkt Stufe fuer Blockdefinition
 PD1,PD2 = Punkte fuer Durchgangshoehen-Konstruktionslinie

7) Auswahlsaetze:

awsr = Auswahlsatz fuer Blockdefinition/Rahmenkonstruktion
 awsf = Auswahlsatz fuer Blockdefinition/3D-Flaechenbelegung
 awsg = Auswahlsatz fuer Blockdefinition/Gelaender

8) Variablen der Fehlerbehandlungsfunktion ('WEND_FEHL'):

angbase0 = urspruengl. Inhalt der ACAD-Systemvariablen 'angbase'
 blipmode0 = urspruengl. Inhalt der ACAD-Systemvariablen 'blipmode'
 clayer0 = urspruengl. Inhalt der ACAD-Systemvariablen 'clayer'
 cmdecho0 = urspruengl. Inhalt der ACAD-Systemvariablen 'cmdecho'
 error0 = urspruengl. Inhalt der Fehlerbehandlungsfunktion '*error*'
 fmeld = beinhaltet die von ACAD/AutoLISP uebergebene Fehlermeldung
 [globale Variable der Fehlerbehandlungsfunktion]
 osmode0 = urspruengl. Inhalt der ACAD-Systemvariablen 'osmode'
 wendfehl = gibt bei 'exit' od. Tastatur-Abbruch einen leeren String aus

;Datei: wtreppe.lsp

;WENDELTREPPENMODUL (c) 1997 Rainer Kuhn

;Dokumentation zum Programm in den Dateien 'wt_info.txt' und 'wt_help.hlp'

; TEIL 1 *****DIALOGBOXSTEUERUNG****

```
(defun dbox ()
  (setq ok nil)
  (while (/= ok 1)
    (setq dcldat (load_dialog "wt_dbox.dcl"))
    (if (not (new_dialog "wtreppe" dcldat)) (exit))
    (WERTVORG)
    (action_tile "vs"
      "(PRUEF1) (if (/= ko 1) (progn (WERTBER) (VORSCH)))")
    (action_tile "accept"
      "(PRUEF2) (if (/= ko 1) (progn (WERTBER) (done_dialog 1)))" );[ggf.ok=1]
    (action_tile "pw" "(WERTSP) (done_dialog 2)" );[ok=2]
    (action_tile "help" "(WERTSP) (done_dialog 3)" );[ok=3]
    (action_tile "cancel" "(done_dialog 0)" );[ok=0]
    (setq ok (start_dialog))
    (unload_dialog dcldat)
    (DBOXOK)
  )
);end
```

;------DBOX-AUSSTIEGSABFRAGE /PUNKTWAHL****

```
(defun dboxok ()
  (cond ((= ok 0) ;[-Abbruch-]
    (progn (setq wendfehl "") (exit))
  )
    ((= ok 2) ;[-Punktwahl-]
    (setq M (getpoint "\n>Mittelpunkt wählen: ")
      x (car M) y (cadr M) z (caddr M)
    )
  )
    ((= ok 3) ;[-Hilfe-]
    (acad_helpdlg "wt_help.hlp" "")
  )
)
);end
```

;------DBOX-FEHLEINGABEPRUEFUNG****

```
(defun pruef1 ()
  (WERTSP)
  (setq ko nil)
  (if (and (= vorg1 "vew") (= weig 0))
    (progn
      (alert "Eingeschlossener Winkel muß >0 sein !")
      (mode_tile "weig" 2)
      (setq ko 1)
    )
  )
  (if (and (= vorg1 "vgt") (= gt 0))
    (progn
      (alert "Grundrißteilung muß >0 sein !")
      (mode_tile "gt" 2)
      (setq ko 1)
    )
  )
)
);end
```

;------DBOX-BLOCKNAMENSPRUEFUNG****

```
(defun pruef2 ()
  (PRUEF1)
  (if (and (equal "" blnst) (/= ko 1))
    (progn
      (alert "Angabe des Blocknamens für Stufe fehlt !")
      (mode_tile "blnst" 2)
      (setq ko 1)
    )
  )
  (if (and (equal "" blnpo) (/= ko 1))
    (progn

```

```

        (alert "Angabe des Blocknamens für Podest fehlt!")
        (mode_tile "blnp" 2)
        (setq ko 1)
    )
)

(if (and (/= nil (tblsearch "BLOCK" (strcase blnst))) (/= ko 1))
    (progn
        (alert (strcat "Blockname für Stufe existiert bereits."
            "\nBitte anderen Blocknamen angeben.))
        (mode_tile "blnst" 2)
        (setq ko 1)
    )
)
(if (and (/= nil (tblsearch "BLOCK" (strcase blnp))) (/= ko 1))
    (progn
        (alert (strcat "Blockname für Podest existiert bereits."
            "\nBitte anderen Blocknamen angeben.))
        (mode_tile "blnp" 2)
        (setq ko 1)
    )
)
(if (and (equal blnst blnp) (/= ko 1))
    (progn
        (alert (strcat "Blocknamen für Stufe und Podest sind gleich."
            "\nBitte unterschiedliche Blocknamen angeben.))
        (mode_tile "blnp" 2)
        (setq ko 1)
    )
)
);end

;-----**DBOX-DESAKTIVIERUNGSFUNKTIONEN**

(defun dview ()
  (mode_tile "weig" 0) (mode_tile "weig" 2)
  (set_tile "gt" "") (mode_tile "gt" 1)
  (mode_tile "anp2" 0)
)
(defun dvgt ()
  (mode_tile "gt" 0) (mode_tile "gt" 2)
  (set_tile "weig" "") (mode_tile "weig" 1)
  (mode_tile "anp2" 1)
)
(defun dvtm ()
  (mode_tile "s0" 2)
  (set_tile "gt" "") (mode_tile "gt" 1)
  (set_tile "weig" "") (mode_tile "weig" 1)
  (mode_tile "anp2" 1)
)
(defun danw ()
  (mode_tile "wang" 0) (mode_tile "wang" 2)
  (set_tile "waug" "") (mode_tile "waug" 1)
)
(defun dauw ()
  (mode_tile "waug" 0) (mode_tile "waug" 2)
  (set_tile "wang" "") (mode_tile "wang" 1)
)

;-----**VORH.WERTE DBOX ZUWEISEN**

(defun wertvorg ()
  (if (/= nil richt) (set_tile "richt" richt))
  (if (/= nil trtyp) (set_tile "trtyp" trtyp))
  (if (/= nil znart) (set_tile "znart" znart))
  (if (/= nil h) (set_tile "h" (rtos h)))
  (if (/= nil d) (set_tile "d" (rtos d)))
  (if (/= nil hs) (set_tile "hs" (rtos hs)))
  (if (/= nil s0) (set_tile "s0" (rtos s0)))
  (if (/= nil di) (set_tile "di" (rtos di)))
  (if (/= nil hg) (set_tile "hg" (rtos hg)))
  (if (/= nil wpg) (set_tile "wpg" (rtos wpg)))
  (if (/= nil x) (set_tile "x" (rtos x)))
  (if (/= nil y) (set_tile "y" (rtos y)))
  (if (/= nil z) (set_tile "z" (rtos z)))
)

```

```

(if (/= nil lname) (set_tile "lname" lname))
(if (/= nil lfarb) (set_tile "lfarb" lfarb))
(if (/= nil blnst) (set_tile "blnst" blnst))
(if (/= nil blnpo) (set_tile "blnpo" blnpo))
(if (/= nil gt) (set_tile "gt" (rtos gt)))
(if (/= nil wang) (set_tile "wang" (rtos wang)))
(if (/= nil waug) (set_tile "waug" (rtos waug)))
(if (/= nil weig) (set_tile "weig" (rtos weig)))
(if (/= nil vorg2) (set_tile "vorg2" vorg2))
(if (equal vorg2 "auw") (DAUW) (DANW))
(if (/= nil vorg1) (set_tile "vorg1" vorg1))
(cond ((equal vorg1 "vgt") (DVGTT))
      ((equal vorg1 "vtm") (DVTM))
      (t (DVEW)))
)

(if (/= nil anp1) (set_tile "anp1" anp1))
(if (/= nil anp2) (set_tile "anp2" anp2))
(if (and (= ok 2) (/= nil nvs)) (VORSCH))
);end

```

;------**DBOX-WERTE SPEICHERN**

```

(defun wertsp ()
  (setq richt (get_tile "richt")
        trtyp (get_tile "trtyp")
        znart (get_tile "znart")
        h (atof (get_tile "h"))
        d (atof (get_tile "d"))
        hs (atof (get_tile "hs"))
        s0 (atof (get_tile "s0"))
        di (atof (get_tile "di"))
        hg (atof (get_tile "hg"))
        wpg (atof (get_tile "wpg"))
        x (atof (get_tile "x"))
        y (atof (get_tile "y"))
        z (atof (get_tile "z"))
        lname (get_tile "lname")
        lfarb (get_tile "lfarb")
        blnst (get_tile "blnst")
        blnpo (get_tile "blnpo")
        vorg1 (get_tile "vorg1")
        vorg2 (get_tile "vorg2")
        gt (atof (get_tile "gt"))
        wang (atof (get_tile "wang"))
        waug (atof (get_tile "waug"))
        weig (atof (get_tile "weig"))
        anp1 (get_tile "anp1")
        anp2 (get_tile "anp2"))
  )
);end

```

;------**DBOX-VORSCHAU**

```

(defun vorsch ()
  (setq nvs 1)
  (set_tile "anz" (itoa anz))
  (set_tile "anz1" (itoa anz1))
  (set_tile "s" (rtos s))
  (set_tile "a" (rtos a))
  (set_tile "wsg" (rtos wsg))
  (set_tile "hd" (rtos hd))
  (set_tile "ls" (rtos ls))
  (if (/= nil gt) (set_tile "gt_vs" (rtos gt)))
  (if (/= nil wang) (set_tile "wang_vs" (rtos wang)))
  (if (/= nil waug) (set_tile "waug_vs" (rtos waug)))
  (if (/= nil weig) (set_tile "weig_vs" (rtos weig)))
  (if (/= nil trm) (set_tile "trm" (rtos trm)))
  (set_tile "warn" warn))
);end

```

;**** TEIL 2 *****WERTEBERECHNUNG****

```

(defun wertber ()
  (WERTSP)
  (WINKUMR)
)

```

```

(setq weig (abs weig))
(setq ri (/ di 2) ;[-Vorab-Wertzuweisungen-]
  r (/ d 2)
  ll 25.0 ;[*Vorgabe: Abstand Lauflinie-AK Treppe 25cm*]
  rl (- r ll)
)
(setq anz0 (/ h s0)) ;[-Anpassung1: Steigung-]
(if (equal anz0 (fix anz0))
  (setq anz (fix anz0))
  (cond ((equal anp1 "sgr") (setq anz (fix anz0)))
        ((equal anp1 "ski") (setq anz (+ (fix anz0) 1))))
)
)
(setq anz1 (- anz 1)
  s (/ h anz)
)
(cond ((equal vorg1 "vew") (BEWEVEW)) ;[-optionale Berechnungswege-]
      ((equal vorg1 "vgt") (BEWEVGT))
      ((equal vorg1 "vtm") (BEWEVTM))
)
)

```

```

(setq wdhg (- 360 wpg)) ;[-Durchgangshöhe-]
(if (> wdhg weig)
  (setq hd (- h hs))
  (progn (setq anz2 (/ wdhg wsg))
         (if (/= anz2 (fix anz2)) (setq anz2 (+ anz2 1)))
         (setq anz2 (fix anz2) hd (- (* anz2 s) hs))
  )
)
)
(setq trm (+ (* 2 s) a) ;[-sonst.Werte/Zuweisungen-]
  ls (- r ri)
  wan (/ (* wang 2 pi) 360)
  wp (/ (* wpg 2 pi) 360)
  M (list x y z)
  MO (list x y (+ z h))
)
(if (equal richt "re") (setq ws (- 0 ws) wp (- 0 wp)))
(WARNMELD)
);end wertber

```

-----**WARNMELDUNGEN FÜR DBOX-VORSCHAU**

```

(defun warnmeld ()
  (setq warn nil)
  (if (or (> 14 s) (> s 21)) (setq warn "**Steigungshöhe nicht 14-21 cm !**"))
  (if (or (> 23 a) (> a 37)) (setq warn "**Auftritt nicht 23-37 cm !**"))
  (if (/= s0 s) (setq warn "**Steigungsvorgabe wurde verändert**"))
  (if (or (> 59 trm) (> trm 65)) (setq warn "**Treppenmaß nicht 59-65 cm !**"))
  (if (< hd 210) (setq warn "**Durchgangshöhe < 210 cm !**"))
  (if (= nil warn) (setq warn "keine"))
);end

```

-----**WINKELUMRECHNUNG 0 <= w < 360**

```

(defun winkumr ()
  (while (< wang 0) (setq wang (+ wang 360)))
  (while (> wang 359.99) (setq wang (- wang 360)))
  (while (< waug 0) (setq waug (+ waug 360)))
  (while (> waug 359.99) (setq waug (- waug 360)))
);end

```

-----**BERECHNUNGSWEG VORG.1= EING.WINKEL**

```

(defun beweview ()
  (setq wsg0 (/ weig anz1) ;[-Anpassung2: eing.Winkel-]
        gt0 (/ 360 wsg0)
  )
  (if (or (equal gt0 (fix gt0)) (equal anp2 "ewfi"))
    (setq gt gt0)
    (cond ((equal anp2 "agr") (setq gt (fix gt0)))
          ((equal anp2 "aki") (setq gt (+ (fix gt0) 1)))
    )
  )
)
(BEWEVGT)

```



```

);end

;-----**BERECHNUNGSWEG VORG.1= GR-TEILUNG**

(defun bewevg ()
  (setq wsg (/ 360.0 gt)
        weig (* anz1 wsg)
  )
  (BEWEVORG2)
  (setq ws (/ (* wsg 2 pi) 360)
        a (* 2 (* (sin (/ ws 2)) rl))
  )
);end

;-----**BERECHNUNGSWEG VORG.1= TREPPENMASS 63**

(defun bewevtm ()
  (setq a (- 63 (* 2 s))
        hm (sqrt (- (expt rl 2) (expt (/ a 2) 2)))
        ws (* 2 (atan (/ (/ a 2) hm)))
        wsg (/ (* ws 360) (* 2 pi))
  )
  (setq gt (/ 360.0 wsg)
        weig (* anz1 wsg)
  )
  (BEWEVORG2)
);end

;-----**BERECHNUNGSWEG VORG.2 (AN-/AUSTRITTSWINKEL)**

(defun bewevorg2 ()
  (cond ((equal vorg2 "anw") ;[-Vorgabe: Antrittswinkel-]
        (cond ((equal richt "li") (setq wang (+ wang weig)))
              ((equal richt "re") (setq wang (- wang weig)))
        )
        (WINKUMR)
  )
  ((equal vorg2 "auw") ;[-Vorgabe: Austrittswinkel-]
   (cond ((equal richt "li") (setq wang (- wang weig)))
         ((equal richt "re") (setq wang (+ wang weig)))
   )
   (WINKUMR)
  )
)
);end

;***TEIL 3*****ZEICHNUNGS-STEUERUNG**

(defun zeichnung ()
  (command "apunkt" "(1 -1 1) ;[-Vorbereitung-]
          "kreis" M r "kreis" (list x y (+ z h (/ hg 2))) r
          "zoom" "grenzen"
  )
  (entdel (entlast)) (entdel (entlast))
  (command "layer" "n" lname "fa" lfarb lname ""
          "layer" "n" "trp_konst"
          "fa" 61 "trp_konst" "lt" "punkt" "trp_konst" ""
  )
  (princ "\nDefiniere Blocks...") ;[-Blockdefinition-]
  (setvar "clayer" "0")
  (setq blvar "stufe")
  (BLOCKDEF) ;[1. Blockdef.= Stufe]
  (setq blvar "podest")
  (BLOCKDEF) ;[2. Blockdef.= Podest]
  (setvar "clayer" lname) ;[-Zeichnungslayer-]
  (if (equal trtyp "sp") ;[-Spindel (ggf.)-]
      (progn
        (if (equal znart "mg") (setq hsp (+ h hg)) (setq hsp (+ h 1)))
        (command "kreis" M (/ di 2) "eigändr" (entlast) "" "o" hsp "")
      )
  )
);end

```

```
(TREPPE) ;[-Treppe-]
(KONSTR) ;[-Konstrukt.linien-]
);end
```

```
;------**BLOCKDEFINITION STUFE/PODEST**
```

```
(defun blockdef ()
  (if (equal blvar "stufe")
      (setq w2 ws nf 6) ;[Stufe]
      (setq w2 wp nf 12) ;[Podest]
  )
  (setq MS0 M w1 0 awsr nil awsf nil awsg nil)
  (if (equal richt "re") ;[-"Rahmen"-]
      (setq EI2 (polar MS0 w1 ri) EI1 (polar MS0 w2 ri)
          EA2 (polar MS0 w1 r) EA1 (polar MS0 w2 r)
      )
      (setq EI1 (polar MS0 w1 ri) EI2 (polar MS0 w2 ri)
          EA1 (polar MS0 w1 r) EA2 (polar MS0 w2 r)
      )
  )
  (command "linie" EI1 EA1 "") (setq awsr (ssget "l"))
  (command "linie" EI2 EA2 "") (ssadd (entlast) awsr)
  (command "bogen" "m" MS0 EI1 EI2) (ssadd (entlast) awsr)
  (command "bogen" "m" MS0 EA1 EA2) (ssadd (entlast) awsr)
  (command "eigändr" awsr "" "o" (- 0 hs) "")
  (setq awsf (ssadd) wf1 w1) ;[-3DFlaechen-]
  (repeat nf
    (setq wf2 (+ wf1 (/ w2 nf))
        EIF1 (polar MS0 wf1 ri) EAF1 (polar MS0 wf1 r)
        EIF2 (polar MS0 wf2 ri) EAF2 (polar MS0 wf2 r)
    )
    (command "3dfläche" "u" EIF1 "u" EAF1 "u" EAF2 "u" EIF2 "")
    ; (command "3dfläche" EIF1 EAF1 EAF2 EIF2 "") ;[*Darst.-Option*]
    (ssadd (entlast) awsf)
    (command "kopieren" (entlast) "" MS0 (list x y (- z hs)))
    (ssadd (entlast) awsf)
    (setq wf1 wf2)
  )
  (setq awsg (ssadd)) ;[-Gelaender-]
  (if (equal znart "mg")
      (if (equal blvar "stufe")
          (progn
            (setq rg r) (GELSTU)
            (if (equal trtyp "wt") (progn (setq rg ri) (GELSTU)))
          )
          (progn
            (GELPODR)
            (if (equal trtyp "wt") (GELPODI))
          )
      )
  )
  (if (equal blvar "stufe") ;[-Def. Block-]
      (setq bname blnst) ;[Stufe]
      (setq bname blnpo) ;[Podest]
  )
  (command "block" bname MS0 awsr awsf awsg "")
);end blockdef
```

```
;------**BLOCKDEF.: GELAENDER STUFE**
```

```
(defun gelstu ()
  (setq wgp w1 zgp (+ z hg) ng 1.0
      MGP (list x y zgp) GP2 (polar MGP wgp rg)
  )
  (repeat nf
    (setq GP1 GP2
        wgp (+ wgp (/ ws nf)) zgp (+ zgp (/ s nf))
        MGP (list x y zgp) GP2 (polar MGP wgp rg)
    )
    (GELHANDL) ;[-Handlauf-]
    (if (and (= rg r) (= (/ ng 2) (fix (/ ng 2)))) ;[-Gelaenderstaebe-]
        (progn (command
            "punkt" (list (car GP2) (cadr GP2) (- (caddr GP2) 15))
            "eigändr" (entlast) "" "o" (- (- zgp z 15)) ""
        )
    )
  )
)
```

```

        (ssadd (entlast) awsg)
      )
    )
  (setq ng (+ ng 1))
)
;[-Gelaenderpfosten-]
(command "kreis" GP2 1.0 "eigändr" (entlast) "" "o" (- (+ hg s hs)) "")
(ssadd (entlast) awsg)
);end

```

```

;-----**BLOCKDEF.:RADIAL-GELAENDER PODEST**

```

```

(defun gelpodr ()
  (setq wgp w2 zgp (+ z hg) MGP (list x y zgp) ng 1.0 rg r
        GP1 (polar MGP wgp ri) GP2 (polar MGP wgp r)
  )
  (GELHANDL) ;[-Handlauf-]
              ;[-Gelaenderpfosten-]
  (command "kreis" GP2 1.0 "eigändr" (entlast) "" "o" (- (+ hg hs)) "")
  (ssadd (entlast) awsg)
  (repeat (fix ls) ;[-Gelaenderstaebe-]
    (if (= (/ ng 25) (fix (/ ng 25)))
      (progn (setq rg (- rg 25) GP2 (polar MGP wgp rg))
              (command
                "punkt" (list (car GP2) (cadr GP2) (- (caddr GP2) 15))
                "eigändr" (entlast) "" "o" (- (- hg 15)) ""
              )
              (ssadd (entlast) awsg)
            )
      )
    )
  (setq ng (+ ng 1))
)
);end

```

```

;-----**BLOCKDEF.:INNEN-GELAENDER PODEST**

```

```

(defun gelpodi ()
  (setq wgp w1 ng 1.0 GP2 (polar MGP wgp ri)
  )
  (repeat nf
    (setq GP1 GP2 wgp (+ wgp (/ wp nf))
          GP2 (polar MGP wgp ri)
    )
    (GELHANDL) ;[-Handlauf-]
    (if (= (/ ng 3) (fix (/ ng 3))) ;[-Gelaenderstaebe-]
      (progn (command
                "punkt" (list (car GP2) (cadr GP2) (- (caddr GP2) 15))
                "eigändr" (entlast) "" "o" (- (- zgp z 15)) ""
              )
              (ssadd (entlast) awsg)
            )
      )
    )
  (setq ng (+ ng 1))
)
;[-Gelaenderpfosten-]
(command "kreis" GP2 1.0 "eigändr" (entlast) "" "o" (- (+ hg hs)) "")
(ssadd (entlast) awsg)
);end

```

```

;-----**BLOCKDEF.:GELAENDER HANDLAUF**

```

```

(defun gelhandl ()
  (command "linie" GP1 GP2 "") ;[-Handlauflinien-]
  (ssadd (entlast) awsg)
  (command "kopieren" (entlast) "" MGP (list x y (- zgp 15)))
)

```

```

(ssadd (entlast) awsg)
(command "bks" "zachse" GP1 GP2                ;[-Handlauf als Rohr-]
      "kreis" (trans GP1 0 1) 1.5
      "eigändr" (entlast) "" "o" (distance GP1 GP2) ""
      "bks" "welt"
)
(ssadd (entlast) awsg)
);end

```

-----**TREPPE ZEICHNEN**

```

(defun treppe ()
  (setq wef wan zef z nt 1)
  (repeat anz
    (setq zef (+ zef s) MS (list x y zef)
          wefg (/ (* wef 360) (* 2 pi)) ;[Einfügewinkel in Grad]
    )
    (if (and (equal znart "mg") (= nt 1)) ;[-Gelaenderanfang-]
      (progn
        (command "kreis" (polar (list x y (+ zef hg)) wef r) 1.0
                  "eigändr" (entlast) "" "o" (- (+ hg s) ""))
        (if (equal trtyp "wt")
            (command "kreis" (polar (list x y (+ zef hg)) wef ri) 1.0
                    "eigändr" (entlast) "" "o" (- (+ hg s) ""))
          )
        )
      )
    (if (not (= nt anz)) ;[-Stufen/Pod.-]
        (command "einfüge" blnst MS "1" "1" wefg) ;[Stufe]
        (command "einfüge" blnpo MS "1" "1" wefg) ;[Podest]
      )
    (setq wef (+ wef ws) nt (+ nt 1))
  )
  (princ "Verdeckte Linien...")
  (command "verdeckt")
);end

```

-----**KONSTRUKTIONSLINIEN**

```

(defun konstr ()
  (setq PD1 (polar (list x y (- zef hs)) (+ (- wef ws) wp) r)
        PD2 (polar (list x y (- zef hd hs)) (+ (- wef ws) wp) r)
  )
  (setvar "clayer" "trp_konst")
  (command
    "linie" PD1 PD2 "" ;[-Durchg.höhe-]
    "bogen" "m" M (polar M wef r) (polar M (- wef 0.01) r) ;[-Basiskreis-]
    "bogen" "m" MO (polar M wef r) (polar MO (- wef 0.01) r) ;[-Treppenloch-]
  ) ;[*Anm.:Bogen wird im Gegensatz zum Kreis nicht verdeckt*]
  (if (equal trtyp "wt") (command "linie" M MO "")) ;[-Mittelachse-]
  (setvar "clayer" lname)
);end

```

TEIL 4**FEHLERBEHANDLUNGSFUNKTION**

```

(defun WEND_FEHL (fmeld)
  (setq *error* error0)
  : (command "apunkt" viewdir0) ;[*Option*]
  (if (= nil wendfehl)
    (princ (strcat "\nFehler bei Wendeltreppenfunktion: " fmeld " "))
    (princ wendfehl)
  )
  (setvar "cmdecho" cmdecho0)
  (setvar "blipmode" blipmode0)
  (setvar "angbase" angbase0)
  (setvar "osmode" osmode0)
  (setvar "clayer" clayer0)
  (princ)
);end

```

TEIL 5**HAUPTPROGRAMM**

```

(princ "Wendeltreppenmodul WTREPPE geladen.\n")

```

```

(defun c:wtrappe (/ anz0 anz2 gt0 hm hsp r rg ri rl
                 dcldat ko nf ng nt nvs ok
                 zef zgp
                 wan wdhg wef wefg wgp wp ws wsg0 w1 w2 wf1 wf2
                 blname blvar
                 E11 EI2 EA1 EA2 EIF1 EIF2 EAF1 EAF2
                 GP1 GP2 M MGP MO MS MS0 PD1 PD2
                 awsr awsf awsg
                 ) ;[*Anm.:sonstige Variablen nicht lokal definiert,
                 ; da für erneuten DBox-Aufruf benoetigt*]
  (princ "\nWENDELTREPPENMODUL\n")
  (setq error0 *error*
        *error* WEND_FEHL ;[*zum Abschalten der Fehlerfkt.];davor*]
        wendfehl nil
  )
  (setq cmdecho0 (getvar "cmdecho")
        blipmode0 (getvar "blipmode")
        angbase0 (getvar "angbase")
        osmode0 (getvar "osmode")
        clayer0 (getvar "clayer")
        viewdir0 (getvar "viewdir")
  )
  (setvar "cmdecho" 0)
  (setvar "blipmode" 0)
  (setvar "angbase" 0)
  (setvar "osmode" 0)
  (DBOX)
  (ZEICHNUNG)
  ; (command "apunkt" viewdir0 "verdeckt") ;[*Option*]
  (setvar "cmdecho" cmdecho0)
  (setvar "blipmode" blipmode0)
  (setvar "angbase" angbase0)
  (setvar "osmode" osmode0)
  (setvar "clayer" clayer0)
  (setq *error* error0)
  (princ)
);end wtrappe

```

//Datei: wt_dbox.dcl

//Dialogbox zum Programm 'wtreppe.lsp'

```
wtreppe : dialog {
    label = "WENDELTREPPENMODUL (c) 1997 Rainer Kuhn";
: column {
: row {
: boxed_column {
    label = "Einstellungen";
: boxed_radio_column {
    key = "richt"; label = "Treppenrichtung:";
: radio_button {
    key = "re"; label = "rechtsdrehend";
}
: radio_button {
    key = "li"; label = "linksdrehend"; value = "1";
}
}
: boxed_radio_column {
    key = "trtyp"; label = "Treppentyp:";
: radio_button {
    key = "sp"; label = "Spindeltreppe"; value = "1";
}
: radio_button {
    key = "wt"; label = "Wendeltreppe";
}
}
: boxed_radio_column {
    key = "znart"; label = "Zeichnungsart:";
: radio_button {
    key = "mg"; label = "mit Geländer"; value = "1";
}
: radio_button {
    key = "og"; label = "ohne Geländer";
}
}
}
: boxed_column {
    label = "Vorgabewerte";
: edit_box {
    key = "h"; label = "Treppe Gesamthöhe [cm].:"; value = "270";
}
: edit_box {
    key = "d"; label = "Treppe Durchmesser [cm].:"; value = "200";
}
: edit_box {
    key = "di"; label = "Spindel/TrAuge Durchm.[cm].:"; value = "10";
}
: edit_box {
    key = "hs"; label = "Stufen-/Podestdicke [cm].:"; value = "5";
}
: edit_box {
    key = "wpg"; label = "Podestwinkel [Grad].:"; value = "60";
}
: edit_box {
    key = "hg"; label = "Geländerhöhe [cm].:"; value = "90";
}
: edit_box {
    key = "s0"; label = "Steigungshöhe (*) [cm].:"; value = "18";
}
}
: boxed_column {
    label = "Mittelpunkt";
: button {
    key = "pw"; label = "Punktwahl <";
}
: spacer_1 {}
: text {
    label = "oder";
}
: text {
    label = "Koordinaten:";
}
: edit_box {
    key = "x"; label = "X .:"; value = "0";
}
}
```

```

        : edit_box {
            key = "y"; label = "Y :"; value = "0";
        }
        : edit_box {
            key = "z"; label = "Z :"; value = "0";
        }
    }

: boxed_column {
    label = "Zeichnungsdaten ";
    : boxed_column {
        label = "Treppenlayer";
        : edit_box {
            key = "lname"; label = "Name :"; value = "treppe";
        }
        : edit_box {
            key = "lfarb"; label = "Farbe :"; value = "40";
        }
    }
    : spacer_0 {}
    : boxed_column {
        label = "Blockname für...";
        : edit_box {
            key = "blnst"; label = "Stufen:"; value = "stufe";
        }
        : edit_box {
            key = "blnpo"; label = "Podest:"; value = "podest";
        }
    }
}
}
}
: spacer_1 {}
: row {
: boxed_column {
    label = "Berechnungsweg (optional)";
    : row {
        : boxed_radio_column {
            key = "vorg1"; label = "Vorgabeoption 1";
            : radio_button {
                key = "vew"; label = "eingeschlossener Winkel"; value = "1";
                action = "(DVEW)";
            }
            : radio_button {
                key = "vgt"; label = "Grundrißteilung";
                action = "(DVGT)";
            }
            : radio_button {
                key = "vtm"; label = "Treppenmaß (2s+a=) 63cm";
                action = "(DVTM)";
            }
        }
        : boxed_radio_column {
            key = "vorg2"; label = "Vorgabeoption 2";
            : text { label = "vorgegebener Winkel:"; }
            : radio_button {
                key = "anw"; label = "Antrittswinkel";
                action = "(DANW)";
            }
            : radio_button {
                key = "auw"; label = "Austrittswinkel"; value = "1";
                action = "(DAUW)";
            }
        }
    }
}
: boxed_column {
    label = "Optionale Vorgabewerte";
    : edit_box {
        key = "weig"; value = "300";
        label = "eingeschlossener Winkel (abs. / w > 0) [Grad]:";
    }
    : edit_box {
        key = "gt"; value = "";
        label = "Grundrißteilung (Stufen pro voller Wendelung):";
    }
    : edit_box {
        key = "wang"; value = "0";
        label = "Antrittswinkel (rel. / Bezug x-Achse) [Grad]:";
    }
}

```

```

    }
    : edit_box {
    key = "waug"; value = "";
    label = "Austrittswinkel (rel. / Bezug x-Achse) [Grad]:";
    }
}
: row {
: boxed_radio_column {
key = "anp1"; label = "Anpassungsoption 1";
: text { label = "(*) Steigung:"; }
: radio_button {
key = "sgr"; label = "ggf. steiler"; value = "1";
}
: radio_button {
key = "skl"; label = "ggf. flacher";
}
}
: boxed_radio_column {
key = "anp2"; label = "Anpassungsoption 2";
: radio_button {
key = "ewfi"; label = "eingeschlossener Winkel fix"; value = "1";
}
: radio_button {
key = "agr"; label = "eing.Winkel/Auftritt ggf. größer";
}
: radio_button {
key = "akl"; label = "eing.Winkel/Auftritt ggf. kleiner";
}
}
}
}
: boxed_column {
label = "Treppenvorschau mit Anpassungen";
: button {
key = "vs"; label = "** Werte neu berechnen **";
}
: concatenation {
: text_part { label = "Steigungshöhe [cm]: "; }
: text_part { key = "s"; label = " "; }
}
: concatenation {
: text_part { label = "Auftritt (Lauflinie) [cm]: "; }
: text_part { key = "a"; label = " "; }
}
: concatenation {
: text_part { label = "Treppenmaß (2s+a) [cm]: "; }
: text_part { key = "trm"; label = " "; }
}
: concatenation {
: text_part { label = "eing. Winkel (abs.) [Grad]: "; }
: text_part { key = "weig_vs"; label = " "; }
}
: concatenation {
: text_part { label = "Grundrissteilung : "; }
: text_part { key = "gt_vs"; label = " "; }
}
: concatenation {
: text_part { label = "Antrittswinkel (->x) [Grad]: "; }
: text_part { key = "wang_vs"; label = " "; }
}
: concatenation {
: text_part { label = "Austrittswinkel(->x) [Grad]: "; }
: text_part { key = "waug_vs"; label = " "; }
}
: concatenation {
: text_part { label = "Anzahl Steigungen : "; }
: text_part { key = "anz"; label = " "; }
}
: concatenation {
: text_part { label = "Anzahl Auftritte (Stufen) : "; }
: text_part { key = "anz1"; label = " "; }
}
: concatenation {
: text_part { label = "Stufenwinkel [Grad]: "; }
: text_part { key = "wsg"; label = " "; }
}
}
: concatenation {

```



```
        : text_part { label = "Stufenlänge" [cm]: ";}
        : text_part { key = "ls"; label = " ";}
    }
    : concatenation {
        : text_part { label = "Durchgangshöhe" [cm]: ";}
        : text_part { key = "hd"; label = " ";}
    }
    : boxed_row {
        label = "WARNMELDUNGEN";
        : text { key = "warn"; label = " ";}
    }
}
}
}
: spacer_1 {}
ok_cancel_help;
}
```

\ Datei: wt_help.hlp
\ Hilfetext zum Programm 'wtreppe.lsp'

WTREPPE (WENDEL TREPPENMODUL)

Das Programm WTREPPE erstellt nach Benutzervorgaben in einem Dialogfenster die Zeichnung einer Wendel- oder einer Spindeltreppe.

Neben der Möglichkeit, verschiedene Grundeinstellungen und Vorgabewerte einzugeben, kann durch Wahl der Drehrichtung und diverse Optionen zwischen 40 verschiedenen Berechnungswegen für die Treppe gewählt werden, um nach ggf. notwendigen Anpassungen der Werte das gewünschte Ergebnis zu erzielen.

Die jeweiligen Berechnungswerte können vor der Zeichnungserstellung in einer Datenvorschau überprüft werden (siehe Hilfetext "Treppenvorschau").

Die Treppe wird auf einem Zeichnungslayer, dessen Name und Farbe man in der Dialogbox bestimmen kann, erzeugt. Sie kann durch Wahl des Mittelpunktes über Koordinaten bzw. durch Punktwahl mit der Maus direkt in eine bestehende Zeichnung eingefügt werden (in diesem Fall kann durch Eingabe des entsprechenden Layernamens auch ein bereits existierender Layer verwendet werden) oder in einer eigenen Zeichnung gespeichert werden, die dann ggf. als Block komplett in eine andere Zeichnung eingefügt werden kann.

\KONSTRUKTIONSLINIEN

\TRP_KONST

KONSTRUKTIONSLINIEN (AUF LAYER "TRP_KONST")

Während der Zeichnungserstellung legt WTREPPE einen neuen Layer "TRP_KONST" an. Auf diesem Layer werden nach Erstellung der eigentlichen Treppenzeichnung folgende Hilfslinien (gepunktete Linien) für den Benutzer gezeichnet:

- Basiskreis: Platzbedarf der Treppe (Durchmesser) im Basisgeschoß
 - Treppenloch: Platzbedarf für ggf. erforderliches Treppenloch im Obergeschoß (Durchmesser)
 - Durchgangshöhenlinie: stellt die in der Vorschau berechnete Durchgangshöhe grafisch dar (= Hinterkante Podestunterseite bis Oberkante Stufe bzw. Basisgeschoß)
 - Mittelachse der Treppe (wird nur bei Wendeltreppen dargestellt)
- Durch Ausschalten oder Frieren des Layers "TRP_KONST" kann die Treppe ohne Hilfslinien dargestellt werden.

\2D-DARSTELLUNG

\GRUNDRISS

\ANSICHT

2D-DARSTELLUNG

Die Zeichnung der Treppe wird von WTREPPE in dreidimensionaler Darstellung erzeugt (Programmvorgabe: Ansichtspunkt 1,-1,1).

Eine zweidimensionale Darstellung kann nach Beendigung der Zeichnung durch den ACAD-Befehl "APUNKT" (Ansichtspunkt) mit Eingabe entsprechender Werte (ggf. anschließend Befehl "VERDECKT") erzeugt werden.

Bsp.:

Grundrißdarstellung	mit APUNKT "0,0,1"
Ansichtsdarstellung 1	mit APUNKT "1,0,0"
Ansichtsdarstellung 2	mit APUNKT "0,1,0"

\LAUFLINIE **LAUFLINIE**

Der Abstand der Lauflinie von der Außenkante der Stufen ist im Programm mit 25 cm vorgegeben. Durch ihn bestimmt sich die Auftrittstiefe der Stufen (mit evtl. Konsequenzen für Stufenwinkel, eingeschlossenen Winkel usw.).

\VORGABEOPTIONEN \OPTIONALE VORGABEWERTE **VORGABEOPTIONEN / OPTIONALE VORGABEWERTE**

Bei den Vorgabeoptionen kann der Benutzer wahlweise festlegen, ob für die Berechnung der Treppe neben den sonstigen Vorgabewerten der eingeschlossene Winkel, die Grundrißteilung oder das Treppenmaß $2s+a=63$ cm (Schrittmaß) vorgegeben wird. In allen drei Fällen kann zusätzlich festgelegt werden, ob der An- oder der Austrittswinkel der Treppe vorgegeben ist.

Die je nach Einstellung nicht benötigten Eingabefelder ("optionale Vorgabewerte") und Anpassungsoptionsfelder werden in der Dialogbox automatisch deaktiviert.

(Mehr darüber im Hilfetext zu den jeweiligen optionalen Vorgabewerten und im Hilfetext "Anpassungsoptionen").

\EINGESCHLOSSENER WINKEL **EINGESCHLOSSENER WINKEL**

Im Eingabefeld "eingeschlossener Winkel" wird der zwischen Antrittswinkel (= Vorderkante erste Stufe) und Austrittswinkel (= Vorderkante Podest) von der Treppe eingeschlossene Winkel als absoluter Winkel in Grad angegeben, d.h. der Wert muß größer Null (positiv) sein und kann 360 Grad überschreiten. 360 Grad entsprechen einer vollen Wendelung. Werden negative Werte eingegeben, so werden diese automatisch in deren Betrag umgewandelt.

Bsp.:

Dreiviertel-Wendelung zwischen An- und Austritt --> Eingabe "270"

Eineinhalb-fache Wendelung zw. An- und Austritt --> Eingabe "540"

\GRUNDRISSTEILUNG **GRUNDRISSTEILUNG**

Im Eingabefeld "Grundrißteilung" wird die Anzahl der Stufen bei einer angenommenen vollen Wendelung der Treppe (d.h. die Unterteilung eines Vollkreises) angegeben. Auch wenn der tatsächlich eingeschlossene Winkel keiner vollen Wendelung (360 Grad) entspricht bzw. mehr als eine Wendelung beschreibt, bleibt die Berechnungsgrundlage des Wertes Grundrißteilung die volle Wendelung. Der Wert muß keine Ganzzahl sein.

\ANTRITTSWINKEL **ANTRITTSWINKEL**

Im Eingabefeld "Antrittswinkel" wird der der Winkel des Treppenantritts (= Vorderkante erste Stufe) als relativer, auf die x-Achse des Koordinatensystems bezogener Winkel in Grad angegeben. Bei Eingabe von negativen Werten oder Werten größer oder gleich 360 Grad wird der Wert von WTREPPE automatisch in einen positiven Winkel zwischen 0 und 359.99 Grad umgerechnet.

VAUSTRITTSWINKEL AUSTRITTSWINKEL

Im Eingabefeld "Austrittswinkel" wird der der Winkel des Treppenaustritts (= Vorderkante Podest) als relativer, auf die x-Achse des Koordinatensystems bezogener Winkel in Grad angegeben. Bei Eingabe von negativen Werten oder Werten größer oder gleich 360 Grad wird der Wert von WTREPPE automatisch in einen positiven Winkel zwischen 0 und 359.99 Grad umgerechnet.

VANPASSUNGSOPTIONEN ANPASSUNGSOPTIONEN

Durch Eingabe bestimmter Vorgaben kann es notwendig werden, daß Vorgabewerte verändert und an die Berechnung angepaßt werden müssen. So kann es passieren, daß die vom Benutzer vorgegebenen Werte nicht mit den in der "Treppenvorschau" ausgegeben Berechnungswerten übereinstimmen (hierbei handelt es sich um den Vorgabewert "Steigungshöhe" oder den optionalen Vorgabewert "eingeschlossener Winkel", falls diese Variante bei "Vorgabeoption 1" gewählt wurde). Durch Einstellung der Anpassungsoptionen (auch in Kombination mit den Vorgabeoptionen) kann der Benutzer versuchen, die Berechnungswerte den gewünschten Vorgabewerten anzunähern bzw. diese anzugleichen.

* Anpassungsoption 1 (Steigung):

Die Anzahl der Stufen bzw. der Steigungen muß ein ganzzahliger Wert sein. Diese bestimmt sich durch die Vorgabewerte "Gesamthöhe" der Treppe (i.A. die Geschoßhöhe) und "Steigungshöhe". Ergibt sich hierbei kein ganzzahliger Wert für die Anzahl der Stufen/Steigungen, so muß -unter der Voraussetzung, daß die Geschoßhöhe ein fester Wert ist- die vorgegebene Steigungshöhe in der "Treppenvorschau" verändert werden. Der Benutzer kann nun wählen, ob die Steigung im Fall einer notwendigen Anpassung entweder steiler oder flacher als der vorgegebene Wert errechnet wird (mit entsprechender Veränderung der Anzahl der Stufen/Steigungen und anderer Werte). Diese Option ist für alle Arten der Einstellung von Vorgabeoptionen relevant.

* Anpassungsoption 2 (eingeschlossener Winkel):

Diese Option ist nur bei entsprechender Einstellung der Vorgabeoption 1 relevant (bei anderer Einstellung ist Anpassungsoption 2 deaktiviert). Hier kann der Benutzer wählen, ob der eingeschlossene Winkel im Fall einer notwendigen Anpassung entweder fix bleibt, also nicht verändert wird (dadurch kann es dazu kommen, daß die Grundrißteilung kein ganzzahliger Wert ist), oder er bei ganzzahliger Grundrißteilung größer oder kleiner als der vorgegebene Wert errechnet wird (mit entsprechender Konsequenz für Grundrißteilung, Auftrittstiefe und andere Werte).

\TREPPENVORSCHAU

\WERTE NEU BERECHNEN

TREPPENVORSCHAU NACH ANPASSUNGEN / WERTE NEU BERECHNEN

Im Fenster "Treppenvorschau" werden nach Anlicken des "* Werte neu berechnen *-Feldes die (nach ggf. erforderlichen automatischen Vorgabewertanpassungen) aktuellen Berechnungswerte und Treppendaten entsprechend den Einstellungen und Vorgabewerten angezeigt.

Bei Änderung von Einstellungen oder Vorgabewerten werden die aktuellen Berechnungswerte nicht automatisch, sondern erst nach Anlicken des "*Werte neu berechnen*-Feldes angezeigt.

Automatischen Vorgabewertanpassungen können z.B. notwendig werden, um eine ganzzahlige Stufenanzahl oder eine ganzzahlige Grundrißteilung zu erreichen.

Vorgabewertanpassungen können beim Vorgabewert "Steigungshöhe" und beim optionalen Vorgabewert "eingeschlossener Winkel" erfolgen; in diesem Fall müssen die Vorschauwerte nicht mit den Vorgabewerten übereinstimmen.

Durch die Anpassungsoptionen kann versucht werden, die tatsächlichen Werte (Vorschau) an die Vorgabewerte anzunähern bzw. diese anzugleichen.

(Mehr dazu im Hilfeindex "Anpassungsoptionen".)

\OK

OK

Durch Anlicken des "OK"-Feldes wird die Treppenzeichnung entsprechend Einstellungen und Werten in der Dialogbox erstellt.

Die Einstellungen und Werte der Dialogbox werden gespeichert und bei erneutem Aufruf von WTREPPE in der Dialogbox wiedergegeben.

\ABBRECHEN

ABBRECHEN

Durch Anlicken des "Abbrechen"-Feldes wird das WTREPPE-Programm abgebrochen und man kehrt zur ACAD-Befehlszeile zurück.

\HILFE

HILFE

Durch Anlicken des "Hilfe..."-Feldes wird die WTREPPE-Hilfefunktion aufgerufen. Die Einstellungen und Werte der Dialogbox werden gespeichert und bei Beendigung der Hilfefunktion in der Dialogbox wiedergegeben.