PyVerm Entwurf

Version 0

Station

+ standpoint: Point
+ orientation: float
+ targetlist: list

+ __init__(self, standpoint=None)
  Initialisiert die Station

+ add_target(self, point, richtung, distance)
  fügt einen Anschluss hinzu

+ abriss(self, report=True):
  berechnet den Abriss (Ori)
  für die Station

+ free_station(self, report=True):
  berechnet Abriss und Standpunkt
  der Station

weitere notwendige Methoden:

+ aufnahme(self)
  um neue Punkte aufzunehmen

+ stakeout(self)
  um Punkte abzustecken

+ schnurgeruest(self)
  um relativ zu Achse Abzustecken

Point

+ y: float
+ x: float
+ z: float
+ number: string

+ __init__(y,x,z):
  erstellt das Punkt Objekt

+ __getitem__(self, key):
  wenn key = 0 get self.x etc,
  imitiert ein tuple

PointDatabase

+ $number: Point
  werden Situativ erstellt

+ __init__()

+ add_point(number, y, x, z)
  erstellt Punkt Objekt in
  der Variable self.$number

+ __getitem__(self, number):
  gibt das Punktobjekt in
  self.$number zurück

+ to_csv(self, filename)
  exportiert Punkte in csv
+ to_koo(self, filename)
  exportiert Punkte in koo

+ from_csv(self, filename)
  importier Punkte aus csv
+ from_koo(self, filename)
  importier Punkte aus koo

Report

+ report

+ __init__(self):
  Initialisiert den Report

+ clear(self):
  leert den Report

+ __str__(self)
  gibt den report als String zurück

+ save(self, filename)
  speichert den Report in txt-Datei

Orthogonal (evtl. keine Klasse)

weitere notwendige Methoden:

+ absteckung()

+ aufnahme()

+ helmert_transformation()

+ lot()

Schnitte (keine Klasse)

weitere notwendige Methoden:

+ gerade_gerade()

+ gerade_kreis()

+ kreis_kreis()

evtl noch Klotoide

Diverses (keine Klasse)

+ distance(point_1, point_2)
  berechnet Distanz zwischen
  zwei Punkten

+ azimut(point_1, point_2)
  berechnet Azimut zwischen
  zwei Punkten

weitere notwendige Methoden:

+ distanzreduktion()
  Reduktion einer Distanz ins
  Koordinatensystem

+ flaechenberechnung()
  Berechnen der Fläche aus
  Punkten inkl. Bogensegmenten

+ Polygonzug()
  berechnen eines Polygonzuges

Kreis (keine Klasse)

weitere notwendige Methoden:

+ 3points()

+ 2points_1line()

+ 1point_1line()

+ pfeilhoehe()

+ radius()

+ tangenten()

+ bogenteile()

+ elemente()