

spiegelwuerfel.py

```
001 # Überprüfung der Rechtwinkligkeit
002 # #####
003
004 # Authors:
005 # Joshua Wolf
006 # Silas Teske
007 # Lasse Zeh
008 # Christopher Mahn
009
010 # #####
011
012 # Import of Libraries
013 # -----
014
015 # import math as m
016 # import string as st
017 # import random as r
018 # import re
019 import numpy as np
020 import os
021
022
023 # -----
024 # Debugging-Settings
025
026 verbose = True # Shows more debugging information
027
028
029 # Functions
030 # -----
031
032
033 # Classes
034 # -----
035
036
037 # Beginning of the Programm
038 # -----
039
040 if __name__ == '__main__':
041
042     # Import der Messwerte des Neigungssensors
043     file = open(os.path.join("data", "imu_data_converted.txt"))
044     imu_data = file.readlines()
045     file.close()
046     for i, e in enumerate(imu_data):
047         imu_data[i] = e.strip().split(";")
048         temp = []
049         for j in imu_data[i]:
050             temp.append(float(j))
051         imu_data[i] = temp
052
053     # Import der Messwerte des Tachymeters
054     file = open(os.path.join("data", "totalstation_data_converted.txt"))
055     totalstation_data = file.readlines()
056     file.close()
057     for i, e in enumerate(totalstation_data):
058         totalstation_data[i] = e.strip().split(";")
059         temp = []
060         for j, f in enumerate(totalstation_data[i]):
061             if(j != 1):
062                 temp.append(float(f))
063             else:
064                 temp.append(f)
065         totalstation_data[i] = temp
066
067     # Berechnung der Orientierung der IMU
068     ori1IMU = []
069     ori1IMU.append(np.average([imu_data[0][8], imu_data[1][8], imu_data[2][8]]))
070     ori1IMU.append(np.average([imu_data[0][9], imu_data[1][9], imu_data[2][9]]))
071     ori1IMU.append(np.average([imu_data[0][10], imu_data[1][10], imu_data[2][10]]))
072
073     ori2IMU = []
074     ori2IMU.append(np.average([imu_data[7][8], imu_data[8][8], imu_data[9][8]]))
075     ori2IMU.append(np.average([imu_data[7][9], imu_data[8][9], imu_data[9][9]]))
076     ori2IMU.append(np.average([imu_data[7][10], imu_data[8][10], imu_data[9][10]]))
077
078     avg_imu1 = []
079     avg_imu1.append(np.average([totalstation_data[1][3], totalstation_data[3][3], totalstation_data[5][3]]))
```

```

080     avg_imu1.append(np.average([totalstation_data[1][4],totalstation_data[3][4],totalstation_data[5]
081                               [4]]))
082     avg_imu2 = []
083     avg_imu2.append(np.average([totalstation_data[7][3],totalstation_data[9][3],totalstation_data[11]
084                               [3]]))
085     avg_imu2.append(np.average([totalstation_data[7][4],totalstation_data[9][4],totalstation_data[11]
086                               [4]]))
087     avg_mirror1 = []
088     avg_mirror1.append(np.average([totalstation_data[0][3],totalstation_data[2][3],totalstation_data[4]
089                                   [3]]))
090     avg_mirror1.append(np.average([totalstation_data[0][4],totalstation_data[2][4],totalstation_data[4]
091                                   [4]]))
092     avg_mirror2 = []
093     avg_mirror2.append(np.average([totalstation_data[6][3],totalstation_data[8]
094                                   [3],totalstation_data[10][3]]))
095     avg_mirror2.append(np.average([totalstation_data[6][4],totalstation_data[8]
096                                   [4],totalstation_data[10][4]]))
097     orilmirror = []
098     orilmirror.append(orilIMU[0])
099     orilmirror.append((avg_imu1[1]-avg_mirror1[1])+orilIMU[1])
100     orilmirror.append((avg_imu1[0]-avg_mirror1[0])+orilIMU[2])
101     ori2mirror = []
102     ori2mirror.append(ori2IMU[0])
103     ori2mirror.append((avg_imu2[1]-avg_mirror2[1])+ori2IMU[1])
104     ori2mirror.append((avg_imu2[0]-avg_mirror2[0])+ori2IMU[2])
105     difference = []
106     difference.append(ori2mirror[0]-orilmirror[0])
107     difference.append(ori2mirror[1]-orilmirror[1])
108     difference.append(ori2mirror[2]-orilmirror[2])
109     differencegon = []
110     differencegon.append(difference[0]*(200/np.pi))
111     differencegon.append(difference[1]*(200/np.pi))
112     differencegon.append(difference[2]*(200/np.pi))
113     print(differencegon)
114
115     # Berechnungskontrolle
116     innenwinkel_imu = -(orilIMU[2] - ori2IMU[2])
117     print(f"Innenwinkel (IMU): {(innenwinkel_imu/np.pi*200):.5f} gon")
118     drehung_totalstation1 = avg_mirror1[0] - avg_imu1[0]
119     print(f"Innenwinkel (Totalstation1): {(drehung_totalstation1/np.pi*200):.5f} gon")
120     drehung_totalstation2 = avg_imu2[0] - avg_mirror2[0]
121     print(f"Innenwinkel (Totalstation2): {(drehung_totalstation2/np.pi*200):.5f} gon")
122     innenwinkel = 2*np.pi - innenwinkel_imu - drehung_totalstation1 - drehung_totalstation2
123     print(f"Innenwinkel: {(innenwinkel/np.pi*200):.5f} gon")
124

```