spiegelwuerfel.py

```python
# Überprürfung der Rechtwinkligkeit
# ###########################################################################

# Authors:
# Joshua Wolf
# Silas Teske
# Lasse Zeh
# Christopher Mahn

# ###########################################################################

# Import of Libraries
# ---------------------------------------------------------------------------

# import math as m
# import string as st
# import random as r
# import re
import numpy as np
import os


# ---------------------------------------------------------------------------
# Debugging-Settings

verbose = True  # Shows more debugging information


# Functions
# ---------------------------------------------------------------------------


# Classes
# ---------------------------------------------------------------------------


# Beginning of the Programm
# ---------------------------------------------------------------------------

if __name__ == '__main__':

    # Import der Messwerte des Neigungssensors
    file = open(os.path.join("data","imu_data_converted.txt"))
    imu_data = file.readlines()
    file.close()
    for i, e in enumerate(imu_data):
        imu_data[i] = e.strip().split(";")
        temp = []
        for j in imu_data[i]:
            temp.append(float(j))
        imu_data[i] = temp

    # Import der Messwerte des Tachymeters
    file = open(os.path.join("data","totalstation_data_converted.txt"))
    totalstation_data = file.readlines()
    file.close()
    for i, e in enumerate(totalstation_data):
        totalstation_data[i] = e.strip().split(";")
        temp = []
        for j, f in enumerate(totalstation_data[i]):
            if(j != 1):
                temp.append(float(f))
            else:
                temp.append(f)
        totalstation_data[i] = temp

    # Berechnung der Orientierung der IMU
    ori1IMU = []
    ori1IMU.append(np.average([imu_data[0][8], imu_data[1][8], imu_data[2][8]]))
    ori1IMU.append(np.average([imu_data[0][9], imu_data[1][9], imu_data[2][9]]))
    ori1IMU.append(np.average([imu_data[0][10], imu_data[1][10], imu_data[2][10]]))
    if(verbose):
        print(f"[Debug] ori1IMU: {ori1IMU}")

    ori2IMU = []
    ori2IMU.append(np.average([imu_data[7][8], imu_data[8][8], imu_data[9][8]]))
    ori2IMU.append(np.average([imu_data[7][9], imu_data[8][9], imu_data[9][9]]))
    ori2IMU.append(np.average([imu_data[7][10], imu_data[8][10], imu_data[9][10]]))
    if(verbose):
        print(f"[Debug] ori2IMU: {ori2IMU}")

    avg_imu1 = []
    avg_imu1.append(np.average([totalstation_data[1][3],totalstation_data[3][3],totalstation_data[5][3]]))
    avg_imu1.append(np.average([totalstation_data[1][4],totalstation_data[3][4],totalstation_data[5][4]]))
    if(verbose):
        print(f"[Debug] avg_imu1: {avg_imu1}")

    avg_imu2 = []
    avg_imu2.append(np.average([totalstation_data[7][3],totalstation_data[9][3],totalstation_data[11][3]]))
    avg_imu2.append(np.average([totalstation_data[7][4],totalstation_data[9][4],totalstation_data[11][4]]))
    if(verbose):
        print(f"[Debug] avg_imu2: {avg_imu2}")

    avg_mirror1 = []
    avg_mirror1.append(np.average([totalstation_data[0][3],totalstation_data[2][3],totalstation_data[4][3]]))
    avg_mirror1.append(np.average([totalstation_data[0][4],totalstation_data[2][4],totalstation_data[4][4]]))
    if(verbose):
        print(f"[Debug] avg_mirror1: {avg_mirror1}")

    avg_mirror2 = []
    avg_mirror2.append(np.average([totalstation_data[6][3],totalstation_data[8][3],totalstation_data[10][3]]))
    avg_mirror2.append(np.average([totalstation_data[6][4],totalstation_data[8][4],totalstation_data[10][4]]))
    if(verbose):
        print(f"[Debug] avg_mirror2: {avg_mirror2}")

    ori1mirror = []
    ori1mirror.append(ori1IMU[0])                              # roll
    ori1mirror.append((avg_imu1[1]-avg_mirror1[1])+ori1IMU[1])  # pitch
    ori1mirror.append((avg_imu1[0]-avg_mirror1[0])+ori1IMU[2])  # yaw
    if(verbose):
        print(f"[Debug] ori1mirror: {ori1mirror}")

    ori2mirror = []
    ori2mirror.append(ori2IMU[0])                              # roll
    ori2mirror.append((avg_imu2[1]-avg_mirror2[1])+ori2IMU[1])  # pitch
    ori2mirror.append((avg_imu2[0]-avg_mirror2[0])+ori2IMU[2])  # yaw
    if(verbose):
        print(f"[Debug] ori2mirror: {ori2mirror}")

    difference = []
    difference.append(ori2mirror[0]-ori1mirror[0])
    difference.append(ori2mirror[1]-ori1mirror[1])
    difference.append(ori2mirror[2]-ori1mirror[2])
    if(verbose):
        print(f"[Debug] difference: {difference}")

    differencegon = []
    differencegon.append(difference[0]*(200/np.pi))
    differencegon.append(difference[1]*(200/np.pi))
    differencegon.append(difference[2]*(200/np.pi))
    if(verbose):
        print(f"[Debug] differencegon: {differencegon}")

    # Berechnungskontrolle
    innenwinkel_imu = -(ori1IMU[2] - ori2IMU[2])
    drehung_totalstation1 = avg_mirror1[0] - avg_imu1[0]
    drehung_totalstation2 = avg_imu2[0] - avg_mirror2[0]
    innenwinkel = 2*np.pi - innenwinkel_imu - drehung_totalstation1 - drehung_totalstation2
    if(verbose):
        print(f"[Debug] Innenwinkel (IMU): {(innenwinkel_imu/np.pi*200):.5f} gon")
        print(f"[Debug] Innenwinkel (Totalstation1): {(drehung_totalstation1/np.pi*200):.5f} gon")
        print(f"[Debug] Innenwinkel (Totalstation2): {(drehung_totalstation2/np.pi*200):.5f} gon")
        print(f"[Debug] Innenwinkel: {(innenwinkel/np.pi*200):.5f} gon")

    # Export results
    file = open(os.path.join("data","results.txt"),f"w")
    file.writelines(f"Result\n")
    file.writelines(f"------\n\n")
    file.writelines(f"Rotation of the mirror (Roll, Pitch, Yaw): {differencegon}\n\n")
    file.writelines(f"Other Values:\n")
    file.writelines(f"ori1IMU: {ori1IMU}\n")
    file.writelines(f"ori2IMU: {ori2IMU}\n")
    file.writelines(f"avg_imu1: {avg_imu1}\n")
    file.writelines(f"avg_imu2: {avg_imu2}\n")
    file.writelines(f"avg_mirror1: {avg_mirror1}\n")
    file.writelines(f"avg_mirror2: {avg_mirror2}\n")
    file.writelines(f"ori1mirror: {ori1mirror}\n")
    file.writelines(f"ori2mirror: {ori2mirror}\n")
    file.close()
```