

```

1 # class for importing data from csv
2 import os
3
4 class DataToImport:
5     def __init__(self):
6         self.data_path = None
7         self.file_content = [None]
8         self.x = []
9         self.y = [] # []
10        self.Anzahl_Spalten = 0 # Zaehler fuer die
gesamte Anzahl der Spalten
11
12        def extract_tablename(self, path):
13            """
14            Funktion extrahiert den Dateinamen aus
demDateipfad
15            :param path: Dateipfad
16            :return: result: Dateiname
17            """
18            result = os.path.basename(path)
19            result = result[:-4]
20            return result
21
22        def separate_lines(self, f_content, spalten):
23            """
24            function seperates the file content in two
variables.
25            :param f_content: is in two columns, e.g. ['4
.234', '3.12']
26            :spalten: Anzahl der gesamten Spalten
27            :return: two seperate variables
28            """
29
30            sp = 1 # Spaltenindex, kann bei 1 beginnen,
da Spaltenindex die x-Werte enthaelt
31            b_x_valid = False # konnte x gelesen werden
?
32            x = []
33
34            # Liste fuer y-Werte erzeugen
35            y = []

```

```

36         for i in range(spalten):
37             y.append([])
38
39             # print("Laenge file_content: ", len(
f_content))
40             i = 0      # Laufindex muss zurueck gesetzt
werden, von for Schleife
41             # solange das Ende des Inhalts nicht erreicht
ist lese ein
42             while i < len(f_content):
43                 # x einlesen, erster Wert muss 'x' sein
44                 try:
45                     if i > 0:
46                         x.append(float(f_content[i][0]))
# erste Stelle ist ein string, kein float
47                     else:
48                         x.append(f_content[i][0])
49                         b_x_valid = True
50                 except:
51                     print(f"Punkt {i} kann nicht
verarbeitet werden")
52
53                 # y einlesen, wenn der x-Wert eingelesen
werden konnte
54
55                 if b_x_valid:
56                     for sp in range(spalten-1):
57                         try:
58
59                             if i > 0:
60                                 y[sp+1].append(float(
f_content[i][sp+1]))
61                             else:
62                                 y[sp+1].append(f_content[
i][sp+1])
63                         except:
64                             print(f"Punkt {i} kann nicht
verarbeitet werden. Zugehoeriges x wird gelöscht")
65                             del x[i]
66
67                     for j in range(1, spalten):

```

```

68             try:
69                 del y[j][i]
70             except:
71                 print(f"Index {j} |
{i} nicht zu loeschen!")
72                 sp = sp+1          #
Index ans Ende der Spalten setzen, damit diese nicht
importiert werden
73                 break          # notwendig um die
restlichen y-Werte nicht zu lesen
74                 sp += 1
75                 sp = 1          # Ruecksetzen fuer
naechsten Durchlauf fuer x
76                 b_x_valid = False    # Gueltigkeitswert
ruecksetzen
77                 i += 1
78
79         return x, y
80
81     def importieren(self, data_path):
82         """
83         function to import a file content
84         :param data_path: file to import
85         :return: nothing, x and y are self
86         """
87         try:
88             with open(data_path, "r") as
csv_import_file:
89                 for line in csv_import_file.read().
split("\n"):
90                     self.file_content.append(line.
split(",") # = csv_import_file.read()
91                     self.file_content.__delitem__(0)
# None in erster Zeile löschen
92
93                     # Anzahl Spalten der Daten zaehlen
94                     for self.Anzahl_Spalten in range(len(
self.file_content[0])):
95                         self.Anzahl_Spalten += 1
96         except:
97             print("EXCEPTION importieren, Datei kann

```

```

97  nicht geöffnet werden")
98
99      # test Daten nach x sortieren
100     Tablename = self.extract_tablename(data_path
    )
101     # Tabelle test.csv muss gesondert behandelt
    werden. Daten werden nach x sortiert.
102     if Tablename == 'test':
103         i = 0
104         while i < len(self.file_content)-1:
105             try:
106                 self.file_content[i][0] = float(
    self.file_content[i][0])
107                 self.file_content[i][1] = float(
    self.file_content[i][1])
108             except:
109                 print(self.file_content[i])
110                 del self.file_content[i]
111                 print("konnte string nicht in
    float konvertieren -> wurde gelöscht")
112                 i -= 1
113
114                 i += 1
115                 del self.file_content[i]
116                 # Daten werden nach x sortiert.
117                 self.file_content.sort()
118
119         if self.Anzahl_Spalten > 0:
120             self.x, self.y = self.separate_lines(
    self.file_content, self.Anzahl_Spalten)
121         else:
122             self.x = []
123             self.y = []
124
125
126     print("importieren DONE")
127

```