

PROGRAMARE PYTHON

PROIECT OOP

Proiect realizat de:
SUVAC ANDREI-SEBASTIAN
GRUPA 532 B

Cuprins

Introducere	pag. 3
Soluția proprie	pag. 8
Testarea soluției proprii	pag.8
Concluzii	pag.21
Bibliografie.....	pag.22



INTRODUCERE

ISTORIA PYTHON ȘI DATE GENERALE DESPRE PYTHON

Python este un limbaj de programare dinamic multi-paradigmă, creat în 1989 de programatorul olandez Guido van Rossum. Van Rossum este și în ziua de astăzi un lider al comunității de dezvoltatori de software care lucrează la perfecționarea limbajul **Python**.

Python este un limbaj de programare puternic, ușor de utilizat și de scris(se poate scrie ușor, de câteva ori mai rapid ca în Java sau C++). În momentul de față, **Python** este considerat de mulți începutul unei cariere de succes în programare datorită faptului că este ușor de reținut și de utilizat.

Python este independent de platformă de aceea permite rularea aceluiași program pe sisteme de operare precum Windows, Linux și chiar iOS.

Python este utilizat și de administratorii de sistem Windows sau Linux.

Python este OOP sau POO(programare orientată pe obiecte), deoarece informațiile rezultate în urma rulării unui program sunt obținute în urma interacțiunii unor entități numite obiecte și nu prin aplicarea unor algoritmi unor date de intrare. Limbajele OOP oferă claritate asupra programului și se apropie de modul în care oamenii percep obiectele care-i înconjoară.

Python e folosit de companii ca Google, Microsoft, Red Hat și Yahoo. Python poate realiza și pagini web cu ajutorul modulelor Python Django sau Flask. Este foarte des folosit datorită ușurinței de a fi învățat, dar în același timp, fiind OOP este utilizat de marile companii bursiere în analiza statistică (putem aminti aici câteva module folosite: pandas, numpy, scipy, matplotlib, tkinter).

Amazon folosește foarte mult **Python** intern pentru configurarea echipamentelor de rețea, a serverelor și altele. Pe lângă acest lucru, **Python** e folosit și pentru amazon cloud (AWS-CLI).

Python este un limbaj dinamic dar în același timp este strong type(dinamic înseamnă că nu trebuie să definim tipul variabilei iar strong înseamnă că nu amestecă stringul cu numărul).

Există mai multe implementări de Python:

1. **CPython** (**Python** scris în C)
2. **JPython** (**Python** care rulează în Java Virtual Machine)
3. **IronPython** (**Python** care rulează în .NET)

Etapa 1.1 – Java/Python

Fișierul “date.csv” conține date de la 10 senzori(temperatură, umiditate, etc), alături de data la care au fost făcute citirile(total 11 coloane). Parsați fișierul(prezent pe platforma Moodle) prin intermediul unei aplicații Java sau **Python**, astfel încât să obțineți:

- Un fișier “temperatura.csv”;
- Un fișier “umiditate.csv”;
- Un fișier “viteza.csv”;
- Un fișier “prezenta.csv”.

Fiecare fișier va conține, pe lângă datele din titlu, și coloana cu data la care au fost făcute citirile.

Etapa 1.2 - Python

Afișați datele din fișierele obținute în etapa 1.1 în grafice în ferestre diferite, în funcție de tipul datelor (un grafic cu toate citirile de temperatură în timp, unul pentru umiditate, etc.). În paralel, în fire de execuție separate, realizați conectarea la un microcontroller prin interfața RS232 și înregistrați datele primite într-un fișier csv nou(coloane, date și timp).

***Puncte bonus

- Realizați o interfață pentru aplicația **Python**
- Permiteți într-un fir separat de execuție conectarea prin port TCP la aplicația **Python**
- Adăugați opțiunea de transmitere a datelor către microcontroller

AutoSave

Off

Date.csv

Search (Alt+Q)

Andrei-Sebastian SUVAC (108303)

AS

Share

Comments

FileHomeInsertPage LayoutFormulasDataReviewViewHelp

Cut

Copy

Format Painter

Clipboard

Calibri

11

A

A

B

I

U

Font

Wrap Text

Merge & Center

Alignment

General

\$

%

0.00

0.00

Number

Normal

Bad

Good

Neutral

Calculation

Check cell

Explanatory...

Input

Styles

Insert

Delete

Format

Cells

AutoSum

Fill

Clear

Editing

Sort & Filter

Find & Select

Editing

Analyze Data

Analysis

A1

</

Fișierul date.csv de pe Moodle

În codul primit de noi la proiect erau introduse unele greșeli intenționate, greșeli pe care le-am modificat pentru a da o lizibilitate codului și pentru a întruni toate condițiile cerute, mai exact:

- Valorile de temperatură mai mici decât -5 vor fi transformate în -5
- Valorile de temperatură mai mari decât 5 vor fi transformate în 5
- Valorile de temperatură care sunt în intervalul [-5, 5] vor fi tratate ca atare și nu se vor modifica
- Valorile de umiditate care sunt egale sau depășesc 80% vor fi tratate ca atare și vor fi transformate în 80%
- Valorile de umiditate care sunt mai mici de 80% vor fi tratate ca atare și nu se vor modifica

Fişierele .csv la prima rulare a codului fără a modifica greşelile:

temperatura.csv																												
Andrei-Sebastian SUVAC (108303)																												
File Home Insert Page Layout Formulas Data Review View Help																												
Clipboard Font Alignment Number Styles Cells Editing Analysis																												
Temp 1																												
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
Temp 1	Temp 2	Temp 3	Data																									
8.41	5.4	1	#####																									
9.09	-4.16	9	#####																									
1.41	-5	20	#####																									
-5	-5	31	#####																									
-5	2.84	36	#####																									
-2.79	9.6	6	#####																									
6.57	7.54	9	#####																									
9.89	-1.46	6	#####																									
4.12	-5	25	#####																									
-5	-5	17	#####																									
-5	0.04	37	#####																									
-5	8.44	9	#####																									
4.2	9.07	15	#####																									
9.91	1.37	27	#####																									
6.5	-5	8	#####																									
-2.88	-5	38	#####																									
-5	-2.75	31	#####																									
-5	6.6	33	#####																									
1.5	9.89	26	#####																									
9.13	4.08	32	#####																									
8.37	-5	15	#####																									
-0.09	-5	6	#####																									
-5	-5	29	#####																									
-5	4.24	25	#####																									
-1.32	9.91	5	#####																									
7.63	6.47	8	#####																									
9.56	-2.92	15	#####																									
2.71	-5	33	#####																									
-5	-5	21	#####																									
-5	1.54	39	#####																									
-4.04	9.15	28	#####																									
5.51	8.34	13	#####																									
10	-0.13	4	#####																									
5.29	-5	20	#####																									
-4.28	-5	8	#####																									
-5	-1.28	19	#####																									
-5	7.65	10	#####																									

temperatura.csv

Microsoft Excel interface showing a spreadsheet named 'umiditae.csv'. The spreadsheet contains data in columns A through AC, with rows 1 through 38. The data is organized into four main sections: 'Umiditate' (A), 'Umiditate' (B), 'Umiditate' (C), and 'Data' (D). The 'Data' column contains numerical values and a series of asterisks (#####) indicating that the data is truncated or hidden. The interface includes the standard Excel ribbon (File, Home, Insert, Page Layout, Formulas, Data, Review, View, Help) and the status bar at the bottom.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	Umiditate	Umiditate	Umiditate	Data																									
2	15	80	55	#####																									
3	16	80	54	#####																									
4	17	80	54	#####																									
5	18	80	55	#####																									
6	19	80	51	#####																									
7	20	80	51	#####																									
8	21	80	50	#####																									
9	22	80	53	#####																									
10	23	80	54	#####																									
11	24	80	53	#####																									
12	25	80	51	#####																									
13	26	80	55	#####																									
14	27	80	55	#####																									
15	28	80	50	#####																									
16	29	80	55	#####																									
17	30	80	53	#####																									
18	31	80	54	#####																									
19	32	80	53	#####																									
20	33	80	52	#####																									
21	34	80	54	#####																									
22	35	80	54	#####																									
23	36	80	52	#####																									
24	37	80	54	#####																									
25	38	80	54	#####																									
26	39	80	55	#####																									
27	40	80	51	#####																									
28	41	80	53	#####																									
29	42	80	52	#####																									
30	43	80	55	#####																									
31	44	80	53	#####																									
32	45	80	53	#####																									
33	46	80	54	#####																									
34	47	80	54	#####																									
35	48	80	50	#####																									
36	49	80	52	#####																									
37	50	80	50	#####																									
38	51	80	53	#####																									

umiditate.csv

AutoSave On viteza.csv Search (Alt+Q) Andrei-Sebastian SUVAC (108303) AS

File Home Insert Page Layout Formulas Data Review View Help

Clipboard Font Alignment Number Styles

Normal Bad Good Neutral Calculation Check Cell Explanatory... Input

Work together faster and easier Share this document with others so you can work on it together. Show me Got it

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
Viteza 1	Data																											
1	0.01	#####																										
2	0.01	#####																										
3	0.01	#####																										
4	0.01	#####																										
5	0.01	#####																										
6	0.01	#####																										
7	0.01	#####																										
8	0.01	#####																										
9	0.03	#####																										
10	0.05	#####																										
11	0.07	#####																										
12	0.09	#####																										
13	0.11	#####																										
14	0.13	#####																										
15	0.15	#####																										
16	0.17	#####																										
17	0.19	#####																										
18	0.21	#####																										
19	0.23	#####																										
20	0.25	#####																										
21	0.27	#####																										
22	0.29	#####																										
23	0.31	#####																										
24	0.33	#####																										
25	0.35	#####																										
26	0.37	#####																										
27	0.39	#####																										
28	0.41	#####																										
29	0.43	#####																										
30	0.45	#####																										
31	0.47	#####																										
32	0.49	#####																										
33	0.51	#####																										
34	0.53	#####																										
35	0.55	#####																										
36	0.57	#####																										
37	0.59	#####																										
38	0.59	#####																										

Ready Type here to search 47°F Sunny 11:46 AM ENG US 25-Oct-21

viteza.csv

AutoSave On prezenta.csv Search (Alt+Q) Andrei-Sebastian SUVAC (108303)

File Home Insert Page Layout Formulas Data Review View Help

Clipboard Font Alignment Number Styles Editing Analysis

Normal Bad Good Neutral Calculation Check Cell Explanatory... Input

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
Prezenta	Prezenta	Data																										
1	0	1	#####																									
2	0	1	#####																									
3	0	1	#####																									
4	0	1	#####																									
5	0	1	#####																									
6	0	1	#####																									
7	0	1	#####																									
8	0	1	#####																									
9	0	1	#####																									
10	0	1	#####																									
11	0	1	#####																									
12	0	1	#####																									
13	0	0	#####																									
14	0	0	#####																									
15	0	0	#####																									
16	0	0	#####																									
17	0	0	#####																									
18	0	0	#####																									
19	0	0	#####																									
20	0	0	#####																									
21	0	0	#####																									
22	0	0	#####																									
23	0	0	#####																									
24	0	0	#####																									
25	0	0	#####																									
26	0	0	#####																									
27	0	0	#####																									
28	0	0	#####																									
29	0	0	#####																									
30	0	0	#####																									
31	0	0	#####																									
32	0	0	#####																									
33	0	0	#####																									
34	0	0	#####																									
35	0	0	#####																									
36	0	0	#####																									
37	0	0	#####																									
38	0	0	#####																									

prezenta.csv

SOLUȚIE PROPRIE ȘI TESTAREA SOLUȚIEI PROPRII

Codul modificat conform cerințelor:

```
partea1_proiect.py ×
1 import csv #import modulul csv implicit, care imi permite sa lucrez cu fisiere de tip csv prin care pot importa, salva sau modifica astfel de fisiere
2 nume_fisier="date.csv" #calea catre fisierul dorit
3
4 #temperatura
5 with open(nume_fisier, 'r', newline='') as csvfile: #deschide fisierul 'date.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv
6     catalog_read = csv.reader(csvfile, delimiter=',', quotechar='"') #instantiez un catalog_read folosind metoda reader a modulului csv cu argumentul csvfile delimitat
7     #prin virgule si caractere
8     for row in catalog_read: #pentru fiecare rand in parte din catalog_read
9         try: #se testeaza codul pentru erori
10             if float(row[0])<=-5: #daca numarul real de pe coloana 1 este mai mic decat -5
11                 row[0]=-5 #se inlocuieste numarul real de pe coloana 1 cu -5
12             except: #in momentul in care se intalneste o eroare, in cazul de fata intervalul [-5,5], daca valoarea din fisierul csv <=-5, imi inlocuieste valoarea cu -5
13                 pass #ce este mai mare decat -5, ramane ca valoare proprie, pass fiind trecerea peste eroare
14             try: #se testeaza codul pentru erori
15                 if float(row[0])>5: #daca numarul real de pe coloana 1 este mai mare decat 5
16                     row[0]=5 #se inlocuieste numarul real de pe coloana 1 cu 5
17             except: #in momentul in care se intalneste o eroare, in cazul de fata intervalul [-5,5], daca valoarea din fisierul csv >5, imi inlocuieste valoarea cu 5
18                 pass #ce este cuprins intre intervalul [-5, 5] este luat ca atare, pass fiind trecerea peste eroare, valoarea ramanand aceeaasi
19             try: #se testeaza codul pentru erori
20                 if float(row[1])<=-5: #daca numarul real de pe coloana 2 este mai mic decat -5
21                     row[1]=-5 #se inlocuieste numarul real de pe coloana 2 cu -5
22             except: #in momentul in care se intalneste o eroare, in cazul de fata intervalul [-5,5], daca valoarea din fisierul csv <=-5, imi inlocuieste valoarea cu -5
23                 pass #ce este mai mare decat -5, ramane ca valoare proprie, pass fiind trecerea peste eroare
24             try: #se testeaza codul pentru erori
25                 if float(row[1])>5: #daca numarul real de pe coloana 2 este mai mare decat 5
26                     row[1]=5 #se inlocuieste numarul real de pe coloana 2 cu 5
27             except: #in momentul in care se intalneste o eroare, in cazul de fata intervalul [-5,5], daca valoarea din fisierul csv >5, imi inlocuieste valoarea cu 5
28                 pass #ce este cuprins intre intervalul [-5, 5] este luat ca atare, pass fiind trecerea peste eroare, valoarea ramanand aceeaasi
29             try: #se testeaza codul pentru erori
30                 if float(row[2])<=-5: #daca numarul real de pe coloana 3 este mai mic decat -5
31                     row[2]=-5 #se inlocuieste numarul real de pe coloana 3 cu -5
32             except: #in momentul in care se intalneste o eroare, in cazul de fata intervalul [-5,5], daca valoarea din fisierul csv <=-5, imi inlocuieste valoarea cu -5
33                 pass #ce este mai mare decat -5, ramane ca valoare proprie, pass fiind trecerea peste eroare
34             try: #se testeaza codul pentru erori
35                 if float(row[2])>5: #daca numarul real de pe coloana 3 este mai mare decat 5
36                     row[2]=5 #se inlocuieste numarul real de pe coloana 3 cu 5
37             except: #in momentul in care se intalneste o eroare, in cazul de fata intervalul [-5,5], daca valoarea din fisierul csv >5, imi inlocuieste valoarea cu 5
38                 pass #ce este cuprins intre intervalul [-5, 5] este luat ca atare, pass fiind trecerea peste eroare, valoarea ramanand aceeaasi
39
```

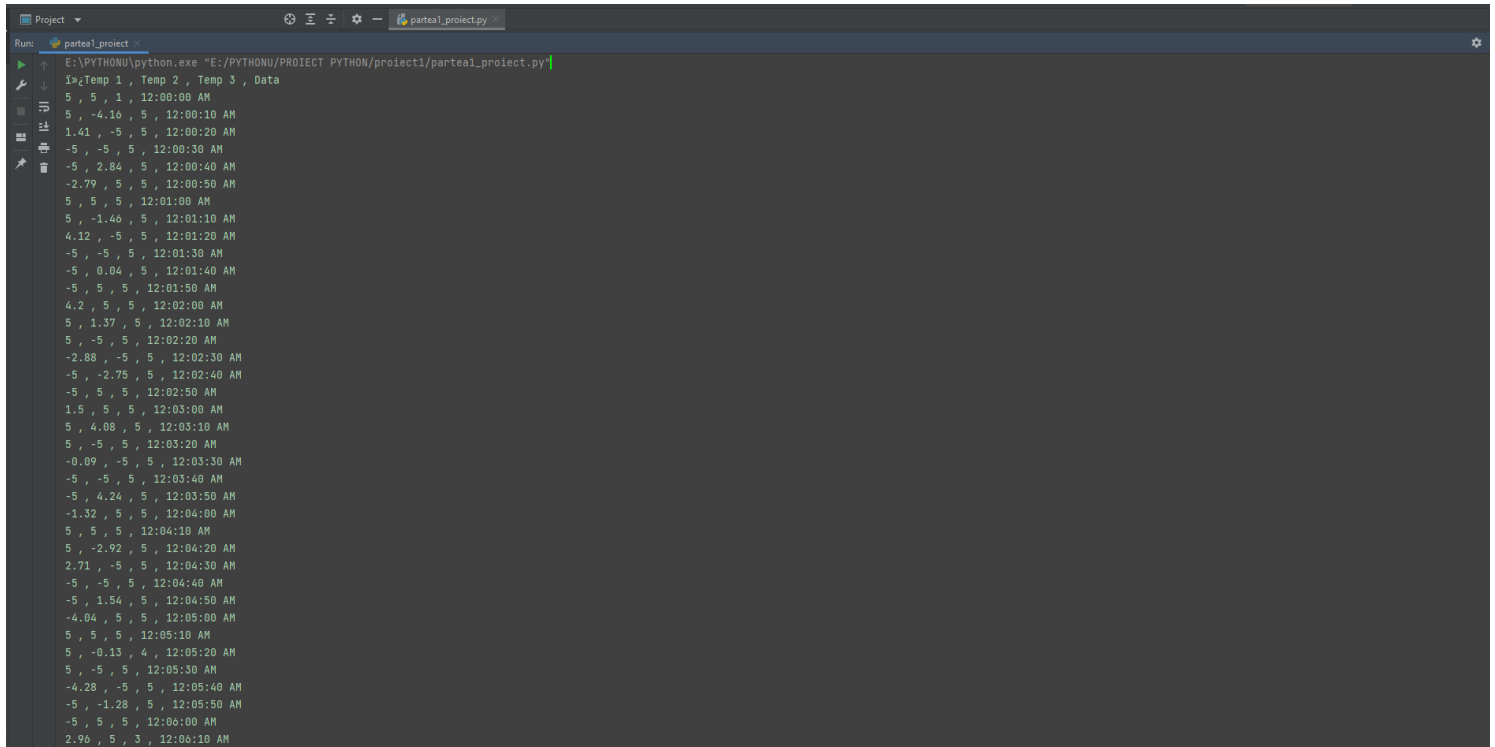


```

37         except: #in momentul in care se intalneste o eroare, in cazul de fata intervalul [-5,5], daca valoarea din fisierul csv >5, imi inlocuieste valoarea
38             pass #ce este cuprins intre intervalul [-5, 5] este luat ca atare, pass fiind trecerea peste eroare, valoarea ramanand aceeaasi
39
40         print(row[0],",",row[1],",",row[2],",",row[9]) #printam toate valorile de pe pozitia 0, 1, 2 si 9 din fisierul date.csv, respectand conditiile de mai sus
41         #scriem datele modificate in fisiere .csv
42         fisier_temp="temperatura.csv" #declar numele fisierului csv cu temperaturi
43         with open(fisier_temp, 'a', newline='') as csvfile_temp: #deschid fisierul "temperatura.csv" pentru adaugare(append) si adaug la un nou obiect csvfile_temp
44             catalog_write_temp = csv.writer(csvfile_temp, delimiter=',', quotechar='|', quoting=csv.QUOTE_MINIMAL) #afisarea datelor din csvfile_temp delimitate prin '|'
45             catalog_write_temp.writerow([row[0],row[1],row[2],row[9]]) #se acceseaza elementele din coloanele cu temperaturi si coloana cu datele la care au fost citite
46         print() #spatiu de tip line break
47
48         #umiditate
49         with open(nume_fisier, 'r', newline='') as csvfile: #deschide fisierul 'date.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv
50             catalog_read = csv.reader(csvfile, delimiter=',', quotechar='|') #instantiez un catalog_read folosind metoda reader a modului csv cu argumentul csvfile delimitat
51                 #prin virgule si caractere
52         for row in catalog_read: #pentru fiecare rand in parte din catalog_read
53             try: #se testeaza codul pentru erori
54                 if float(row[3])>80: #daca numarul real de pe coloana 4 este mai mare decat 80
55                     row[3]=80 #se inlocuieste numarul real de pe coloana 4 cu 80
56             except: #exceptia fiind ca in momentul in care se citeste o valoare mai mica sau egala cu 80 din fisierul csv
57                 pass #valoarea ramane neschimbata, pass fiind trecerea peste eroare
58             try: #se testeaza codul pentru erori
59                 if float(row[4])>80: #daca numarul real de pe coloana 5 este mai mare decat 80
60                     row[4]=80 #se inlocuieste numarul real de pe coloana 5 cu 80
61             except: #exceptia fiind ca in momentul in care se citeste o valoare mai mica sau egala cu 80 din fisierul csv
62                 pass #valoarea ramane neschimbata, pass fiind trecerea peste eroare
63             try: #se testeaza codul pentru erori
64                 if float(row[5])>80: #daca numarul real de pe coloana 6 este mai mare decat 80
65                     row[5]=80 #se inlocuieste numarul real de pe coloana 6 cu 80
66             except: #exceptia fiind ca in momentul in care se citeste o valoare mai mica sau egala cu 80 din fisierul csv
67                 pass #valoarea ramane neschimbata, pass fiind trecerea peste eroare
68
69             print(row[3],",",row[4],",",row[5],",",row[9]) #printam toate valorile de pe pozitia 3, 4, 5 si 9 din fisierul date.csv, respectand conditiile de mai sus
70
71             fisier_temp="umiditate.csv" #declar numele fisierului csv cu valorile pentru umiditate
72             with open(fisier_temp, 'a', newline='') as csvfile_temp: #deschid fisierul "umiditate.csv" pentru adaugare(append) si adaug la un nou obiect csvfile_temp
73                 catalog_write_temp = csv.writer(csvfile_temp, delimiter=',', quotechar='|', quoting=csv.QUOTE_MINIMAL) #afisarea datelor din csvfile_temp delimitate prin '|'
74                 catalog_write_temp.writerow([row[3],row[4],row[5],row[9]]) #se acceseaza elementele din coloanele cu umiditate si coloana cu datele la care au fost citite
75         print() #spatiu de tip line break
76
77         #viteza
78         with open(nume_fisier, 'r', newline='') as csvfile: #deschide fisierul 'date.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv
79             catalog_read = csv.reader(csvfile, delimiter=',', quotechar='|') #instantiez un catalog_read folosind metoda reader a modului csv cu argumentul csvfile delimitat
80                 #prin virgule si caractere
81         for row in catalog_read: #pentru fiecare rand in parte din catalog_read
82             print(row[0],",",row[9]) #printam toate valorile de pe pozitia 0 si 9 din fisierul date.csv
83
84             fisier_temp="viteza.csv" #declar numele fisierului csv cu valorile pentru viteza
85             with open(fisier_temp, 'a', newline='') as csvfile_temp: #deschid fisierul "viteza.csv" pentru adaugare(append) si adaug la un nou obiect csvfile_temp
86                 catalog_write_temp = csv.writer(csvfile_temp, delimiter=',', quotechar='|', quoting=csv.QUOTE_MINIMAL) #afisarea datelor din csvfile_temp delimitate prin '|'
87                 catalog_write_temp.writerow([row[0],row[9]]) #se acceseaza elementele din coloana cu viteza si din coloana cu datele la care au fost citite
88         print() #spatiu de tip line break
89
90         #prezenta
91         with open(nume_fisier, 'r', newline='') as csvfile: #deschide fisierul 'date.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv
92             catalog_read = csv.reader(csvfile, delimiter=',', quotechar='|') #instantiez un catalog_read folosind metoda reader a modului csv cu argumentul csvfile delimitat
93                 #prin virgule si caractere
94         for row in catalog_read: #pentru fiecare rand in parte din catalog_read
95             print(row[7],",",row[8],",",row[9]) #printam toate valorile de pe pozitiile 7, 8 si 9 din fisierul date.csv
96
97             fisier_temp="prezenta.csv" #declar numele fisierului csv cu valorile pentru prezenta
98             with open(fisier_temp, 'a', newline='') as csvfile_temp: #deschid fisierul "prezenta.csv" pentru adaugare(append) si adaug la un nou obiect csvfile_temp
99                 catalog_write_temp = csv.writer(csvfile_temp, delimiter=',', quotechar='|', quoting=csv.QUOTE_MINIMAL) #afisarea datelor din csvfile_temp delimitate prin '|'
100                 catalog_write_temp.writerow([row[7],row[8],row[9]]) #se acceseaza elementele din coloanele cu prezenta si din coloana cu datele la care au fost citite
101         print() #spatiu de tip line break

```

Putem observa că datele din fișierele .csv se modifică conform cerințelor de mai sus atunci când rulăm programul:



```
Run: partea1_project.py
E:\PYTHONU\python.exe "E:/PYTHONU/PROJECT PYTHON/project1/parteal_project.py"
1>Temp 1 , Temp 2 , Temp 3 , Data
5 , 5 , 1 , 12:00:00 AM
5 , -4.16 , 5 , 12:00:10 AM
1.41 , -5 , 5 , 12:00:20 AM
-5 , -5 , 5 , 12:00:30 AM
-5 , 2.84 , 5 , 12:00:40 AM
-2.79 , 5 , 5 , 12:00:50 AM
5 , 5 , 5 , 12:01:00 AM
5 , -1.46 , 5 , 12:01:10 AM
4.12 , -5 , 5 , 12:01:20 AM
-5 , -5 , 5 , 12:01:30 AM
-5 , 0.04 , 5 , 12:01:40 AM
-5 , 5 , 5 , 12:01:50 AM
4.2 , 5 , 5 , 12:02:00 AM
5 , 1.37 , 5 , 12:02:10 AM
5 , -5 , 5 , 12:02:20 AM
-2.88 , -5 , 5 , 12:02:30 AM
-5 , -2.75 , 5 , 12:02:40 AM
-5 , 5 , 5 , 12:02:50 AM
1.5 , 5 , 5 , 12:03:00 AM
5 , 4.08 , 5 , 12:03:10 AM
5 , -5 , 5 , 12:03:20 AM
-0.09 , -5 , 5 , 12:03:30 AM
-5 , -5 , 5 , 12:03:40 AM
-5 , 4.24 , 5 , 12:03:50 AM
-1.32 , 5 , 5 , 12:04:00 AM
5 , 5 , 5 , 12:04:10 AM
5 , -2.92 , 5 , 12:04:20 AM
2.71 , -5 , 5 , 12:04:30 AM
-5 , -5 , 5 , 12:04:40 AM
-5 , 1.54 , 5 , 12:04:50 AM
-4.04 , 5 , 5 , 12:05:00 AM
5 , 5 , 5 , 12:05:10 AM
5 , -0.13 , 4 , 12:05:20 AM
5 , -5 , 5 , 12:05:30 AM
-4.28 , -5 , 5 , 12:05:40 AM
-5 , -1.28 , 5 , 12:05:50 AM
-5 , 5 , 5 , 12:06:00 AM
2.96 , 5 , 3 , 12:06:10 AM
```

Temperatura

```
PROJECT PYTHON | project1 | partea1_project.py
Project
Run
partea1_project
5, 5, 5, 12:15:40 AM
5, -1.8, 5, 12:15:50 AM
3.8, -5, 5, 12:16:00 AM
-5, -5, 5, 12:16:10 AM
-5, 0.4, 5, 12:16:20 AM
-5, 5, 4, 12:16:30 AM

Umiditate 1, Umiditate 2, Umiditate 3, Data
15, 80, 55, 12:00:00 AM
16, 80, 54, 12:00:10 AM
17, 80, 54, 12:00:20 AM
18, 80, 55, 12:00:30 AM
19, 80, 51, 12:00:40 AM
20, 80, 51, 12:00:50 AM
21, 80, 50, 12:01:00 AM
22, 80, 53, 12:01:10 AM
23, 80, 54, 12:01:20 AM
24, 80, 53, 12:01:30 AM
25, 80, 51, 12:01:40 AM
26, 80, 55, 12:01:50 AM
27, 80, 55, 12:02:00 AM
28, 80, 50, 12:02:10 AM
29, 80, 55, 12:02:20 AM
30, 80, 53, 12:02:30 AM
31, 80, 54, 12:02:40 AM
32, 80, 53, 12:02:50 AM
33, 80, 52, 12:03:00 AM
34, 80, 54, 12:03:10 AM
35, 80, 54, 12:03:20 AM
36, 80, 52, 12:03:30 AM
37, 80, 54, 12:03:40 AM
38, 80, 54, 12:03:50 AM
39, 80, 55, 12:04:00 AM
40, 80, 51, 12:04:10 AM
41, 80, 53, 12:04:20 AM
42, 80, 52, 12:04:30 AM
43, 80, 55, 12:04:40 AM
44, 80, 53, 12:04:50 AM
45, 80, 53, 12:05:00 AM
46, 80, 54, 12:05:10 AM
```

Umiditate

```
PROJECT PYTHON | project1 | partea1_project.py
Project
Run
partea1_project
80, 80, 54, 12:15:40 AM
80, 80, 52, 12:15:50 AM
80, 80, 53, 12:16:00 AM
80, 80, 55, 12:16:10 AM
80, 80, 52, 12:16:20 AM
80, 80, 53, 12:16:30 AM

Viteza 1, Data
0.01, 12:00:00 AM
0.01, 12:00:10 AM
0.01, 12:00:20 AM
0.01, 12:00:30 AM
0.01, 12:00:40 AM
0.01, 12:00:50 AM
0.01, 12:01:00 AM
0.03, 12:01:10 AM
0.05, 12:01:20 AM
0.07, 12:01:30 AM
0.09, 12:01:40 AM
0.11, 12:01:50 AM
0.13, 12:02:00 AM
0.15, 12:02:10 AM
0.17, 12:02:20 AM
0.19, 12:02:30 AM
0.21, 12:02:40 AM
0.23, 12:02:50 AM
0.25, 12:03:00 AM
0.27, 12:03:10 AM
0.29, 12:03:20 AM
0.31, 12:03:30 AM
0.33, 12:03:40 AM
0.35, 12:03:50 AM
0.37, 12:04:00 AM
0.39, 12:04:10 AM
0.41, 12:04:20 AM
0.43, 12:04:30 AM
0.45, 12:04:40 AM
0.47, 12:04:50 AM
0.49, 12:05:00 AM
0.51, 12:05:10 AM
```

Viteza

```
PROJECT PYTHON | proiect1 | parte1_proiect.py
Run | parte1_proiect.py
0.2 , 12:13:10 AM
0.17 , 12:13:20 AM
0.14 , 12:13:30 AM
0.11 , 12:13:40 AM
0.08 , 12:13:50 AM
0.05 , 12:14:00 AM
0 , 12:14:10 AM
0 , 12:14:20 AM
0 , 12:14:30 AM
0 , 12:14:40 AM
0 , 12:14:50 AM
0 , 12:15:00 AM
0 , 12:15:10 AM
0 , 12:15:20 AM
0 , 12:15:30 AM
0 , 12:15:40 AM
0 , 12:15:50 AM
0 , 12:16:00 AM
0 , 12:16:10 AM
0 , 12:16:20 AM
0 , 12:16:30 AM

Prezenta 1 , Prezenta 2 , Data
0 , 1 , 12:00:00 AM
0 , 1 , 12:00:10 AM
0 , 1 , 12:00:20 AM
0 , 1 , 12:00:30 AM
0 , 1 , 12:00:40 AM
0 , 1 , 12:00:50 AM
0 , 1 , 12:01:00 AM
0 , 1 , 12:01:10 AM
0 , 1 , 12:01:20 AM
0 , 1 , 12:01:30 AM
0 , 1 , 12:01:40 AM
0 , 0 , 12:01:50 AM
0 , 0 , 12:02:00 AM
0 , 0 , 12:02:10 AM
0 , 0 , 12:02:20 AM
0 , 0 , 12:02:30 AM
0 , 0 , 12:02:40 AM
```

Prezenta

În continuare, am modificat codul primit la ședința de proiect, codul care ne permite să realizăm graficele aferente, de temperatură, de umiditate, de viteză și de prezență. De fiecare dată când rulăm acest cod trebuie să fim atenți deoarece primul program realizat mai sus rescrie fișierele .csv în continuare, iar graficele se modifică întocmai, deci trebuie mereu să ștergem fișierele .csv când rulăm primul sau al doilea program, pentru a fi sigur că nu avem suprapuneri. Acest cod conținea de asemenea greșeli, dar erau foarte minore, greșeli de label sau greșeli de scriere a coloanelor din fișierele .csv în vederea realizării corecte a graficului.

Codul modificat conform cerințelor:

```
parte1_proiect.py | grafice_proiect.py
1 | import csv #importam modulul csv care ne permite sa lucram cu fisierele de tip csv prin care se pot importa, salva sau modifica astfel de fisiere
2 | import matplotlib.pyplot as plt #importam matplotlib.pyplot cu eticheta scurta plt (scriere mai facila)
3 |
4 | #Vom stoca individual in listele de mai jos citirile de pe coloanele de interes
5 |
6 | class grafice_statice(): #Instantiem o clasa cu denumirea grafice_statice
7 |     def temperatura(self): #Definim metoda temperatura, prin primul argument al tuturor metodelor(self), care specifică ca metoda apartine clasei
8 |         x = [] # timp
9 |         y = [] # temperatura 1
10 |         z = [] # temperatura 2
11 |         w = [] # temperatura 3
12 |
13 |         with open('temperatura.csv', 'r') as csvfile: #deschide fisierul 'temperatura.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv
14 |             date = csv.reader(csvfile, delimiter=',') #instantiez date folosind metoda reader a modulului csv delimitat prin ','
15 |             for row in date: #pentru fiecare rand in parte din date
16 |                 try: #se testeaza codul pentru erori
17 |                     y.append(float(row[0])) #la lista y adaug valorile reale din coloana 1, valorile pentru temperatura 1, din fisierul 'temperatura.csv'
18 |                     z.append(float(row[1])) #la lista z adaug valorile reale din coloana 2, valorile pentru temperatura 2, din fisierul 'temperatura.csv'
19 |                     w.append(float(row[2])) #la lista w adaug valorile reale din coloana 3, valorile pentru temperatura 3, din fisierul 'temperatura.csv'
20 |                     x.append(row[3]) #la lista x adaug valorile pentru timp din coloana 4, din fisierul 'temperatura.csv'
21 |                 except Exception as e: #in cazul unei erori - cum va aparea la transformarea primului rand, programul trece la pasul urmator
22 |                     pass #trecerea peste eroare
23 |             with plt.style.context("fivethirtyeight"): #stilul graficului
24 |
25 |
```

```

26 #Identificam in mod unic graficul curent (figura 1)
27
28 plt.figure(1)
29 #Realizam plot-uri intre perechile (Timp, Temperatura 1), (Timp, Temperatura 2) si (Timp, Temperatura 3)
30 plt.plot(x,y, color="khaki", label='Temperatura 1') #(Timp, Temperatura 1)
31 plt.plot(x,z, color="steelblue", label='Temperatura 2') #(Timp, Temperatura 2)
32 plt.plot(x,w, color="crimson", label='Temperatura 3') #(Timp, Temperatura 3)
33 #Afisam etichetele axelor, titlul si legenda graficului
34 plt.xlabel('Timp') #eticheta axei OX
35 plt.ylabel('Temperatura') #eticheta axei OY
36 plt.title('Temperatura (Timp) ') #titlul graficului
37 plt.legend() #legenda graficului
38 #Specificam diviziunile axelor OX si OY
39 plt.xticks([0, 9, 18, 27, 36, 45, 54, 63, 72, 81, 90, 99]) #OX
40 plt.yticks([-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6]) #OY
41
42
43 #Definim mai jos listele si setarile pentru graficul de umiditate
44
45 def umiditate(self): #Definim metoda umiditate, prin primul argument al tuturor metodelor(self), care specifică ca metoda apartine clasei
46     a = [] # timp
47     b = [] # umiditate 1
48     c = [] # umiditate 2
49     d = [] # umiditate 3
50     with open('umiditate.csv', 'r') as csvfile: #deschide fisierul 'umiditate.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv

```

```

50 with open('umiditate.csv', 'r') as csvfile: #deschide fisierul 'umiditate.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv
51     date = csv.reader(csvfile, delimiter=',') #instantiez date folosind metoda reader a modului csv delimitat prin ','
52     for row in date: #pentru fiecare rand in parte din date
53         try: #se testeaza codul pentru erori
54             b.append(float(row[0])) #la lista b adaug valorile reale din coloana 1, valorile pentru umiditate 1, din fisierul 'umiditate.csv'
55             c.append(float(row[1])) #la lista c adaug valorile reale din coloana 2, valorile pentru umiditate 2, din fisierul 'umiditate.csv'
56             d.append(float(row[2])) #la lista d adaug valorile reale din coloana 3, valorile pentru umiditate 3, din fisierul 'umiditate.csv'
57             a.append(row[3]) #la lista a adaug valorile pentru timp din coloana 4, din fisierul 'umiditate.csv'
58         except Exception as e: # in cazul unei erori - cum va aparea la transformarea primului rand, programul trece la pasul urmator
59             #print ("Warning, error found: ", row)
60         pass #trecerea peste eroare
61     with plt.style.context("fivethirtyeight"):
62         plt.figure(2) #Identificam in mod unic graficul curent (figura 2)
63         plt.plot(a,b, color="olive", label='Umiditate 1') #(Timp, Umiditate 1)
64         plt.plot(a,c, color="yellow", label='Umiditate 2') #(Timp, Umiditate 2)
65         plt.plot(a,d, color="violet", label='Umiditate 3') #(Timp, Umiditate 3)
66         plt.xlabel('Timp') #eticheta axei OX
67         plt.ylabel('Umiditate %') #eticheta axei OY
68         plt.title('Umiditate (Timp) ') #titlul graficului
69         plt.legend() #legenda graficului
70         plt.xticks([0, 10, 20, 30, 40, 50, 60, 70, 80]) #OX
71         plt.yticks([0, 10, 20, 30, 40, 50, 60, 70, 80]) #OY
72
73 #Definim mai jos listele si setarile pentru graficul de viteza
74

```

```

73 #Definim mai jos listele si setarile pentru graficul de viteza
74
75 def viteza(self): #Definim metoda viteza, prin primul argument al tuturor metodelor(self), care specifică ca metoda apartine clasei
76     i = [] # viteza 1
77     j = [] # timp
78     plt.figure(3) # Identificam in mod unic graficul curent (figura 3)
79     with open('viteza.csv', 'r') as csvfile: #deschide fisierul 'viteza.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv
80         date = csv.reader(csvfile, delimiter=',') #instantiez date folosind metoda reader a modului csv delimitat prin ','
81         for row in date: #pentru fiecare rand in parte din date
82             try: #se testeaza codul pentru erori
83                 i.append(float(row[0]))
84                 j.append(row[1])
85             except Exception as e: # in cazul unei erori - cum va aparea la transformarea primului rand, programul trece la pasul urmator
86                 #print ("Warning, error found: ", row)
87             pass #trecerea peste eroare
88
89     with plt.style.context("fivethirtyeight"): #stilul graficului
90         plt.plot(j,i, color="lightcoral", marker="o", markerfacecolor="black", markersize="4", label='Viteza 1') #(Timp, Viteza 1)
91         plt.xlabel('Timp') #eticheta axei OX
92         plt.ylabel('Viteza') #eticheta axei OY
93         plt.title('Viteza (Timp) ') #titlul graficului
94         plt.legend() #legenda graficului
95         plt.xticks([0, 9, 18, 27, 36, 45, 54, 63, 72, 81, 90, 99]) #OX
96         plt.yticks([0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]) # OY
97

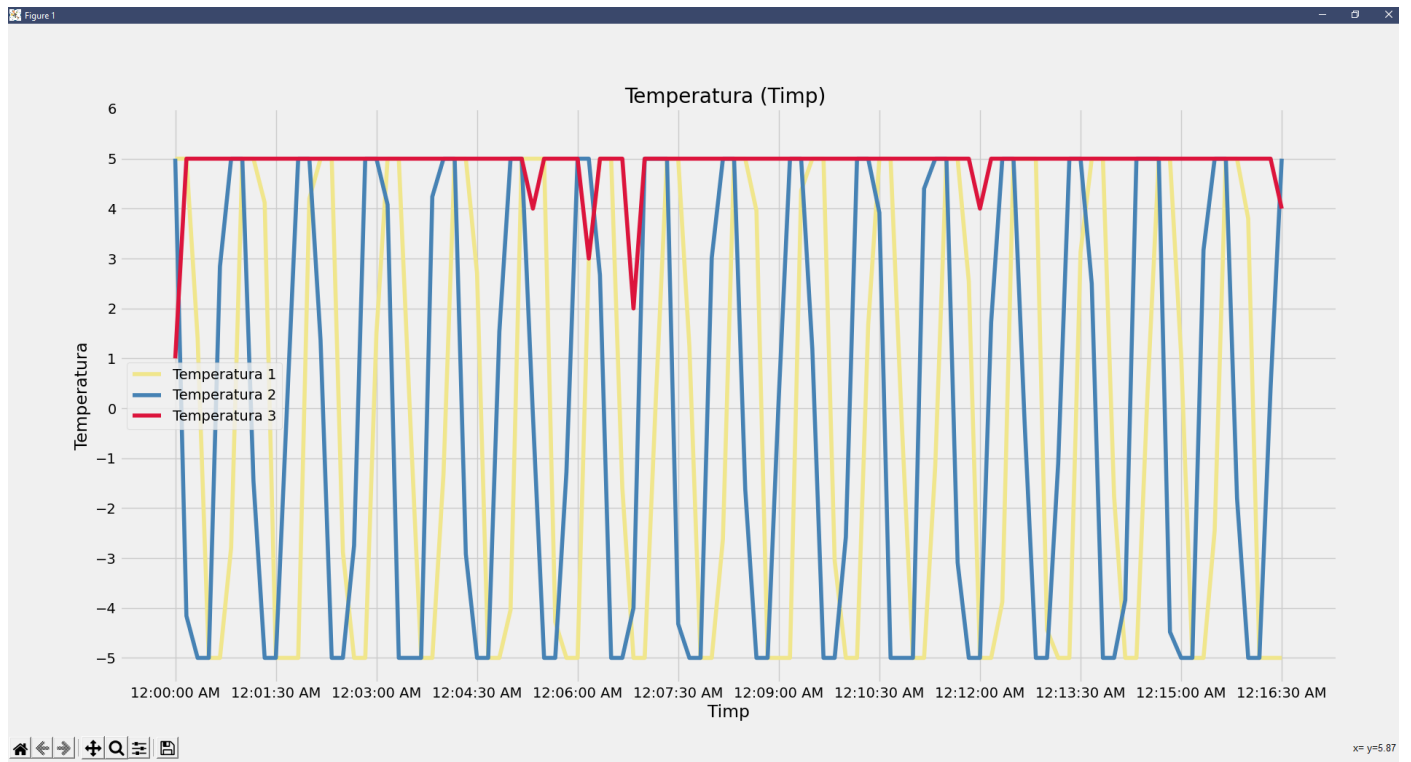
```

```

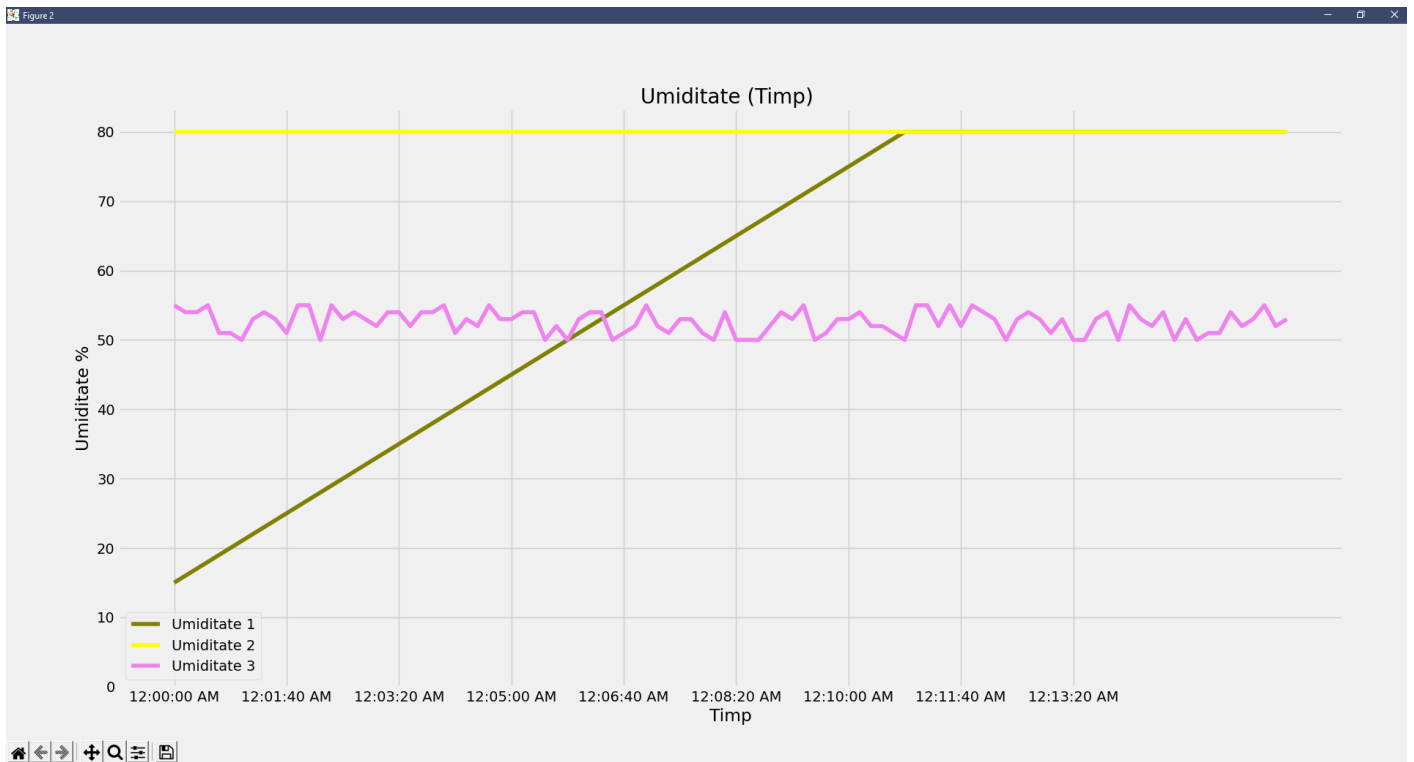
partea1_proiect.py x grafice_proiect.py
98 def prezenta(self): #Definim metoda prezenta, prin primul argument al tuturor metodelor(self), care specifică ca metoda apartine clasei
99     e = [] # timp
100     f = [] # prezenta 1
101     g = [] # prezenta 2
102     plt.figure(4) # Identificam in mod unic graficul curent(figura 3)
103     with open('prezenta.csv',
104             'r') as csvfile: # deschide fisierul 'prezenta.csv' pentru citire('r') si se instantiaza un obiect csvfile de tip csv
105         date = csv.reader(csvfile,
106                           delimiter=',') # instantiez date folosind metoda reader a modulului csv delimitat prin ','
107         for row in date: # pentru fiecare rand in parte din date
108             try: # se testeaza codul pentru erori
109                 f.append(float(row[
110                             0])) # la lista f adaug valorile din coloana 1, valorile pentru prezenta 1, din fisierul 'prezenta.csv'
111                 g.append(float(row[
112                             1])) # la lista g adaug valorile din coloana 2, valorile pentru prezenta 2, din fisierul 'prezenta.csv'
113                 e.append(
114                     row[2]) # la lista e adaug valorile pentru timp din coloana 3, din fisierul 'prezenta.csv'
115             except Exception as error: # in cazul unei erori - cum va aparea la transformarea primului rand, programul trece la pasul urmator
116                 # print ("Warning, error found: ")
117                 pass # trecerea peste eroare
118
119     with plt.style.context("fivethirtyeight"): # stilul graficului
120
121         plt.suptitle('Grafice Prezenta') # titlul graficelor
122
123         plt.subplot(2, 1, 1) # creare subgrafice(nr linii, nr coloane, pozitie index)
124         plt.plot(e, f, color="blue", linewidth=1) # (Prezenta 1, Timp)
125         plt.title('Grafic Prezenta 1') # titlul graficului
126         plt.xlabel('Timp') # eticheta axei OX
127         plt.ylabel('Prezenta1') # eticheta axei OY
128         plt.xticks([0, 20, 40, 60, 80, 99]) # subdiviziunile de pe axa OX, valorile timpului
129
130         plt.subplot(2, 1, 2) # creare subgrafice(nr linii, nr coloane, pozitie index)
131         plt.plot(e, g, color="green", linewidth=1) # (Prezenta 2, Timp)
132         plt.title('Grafic Prezenta 2') # titlul graficului
133         plt.xlabel('Timp') # eticheta axei OX
134         plt.ylabel('Prezenta2') # eticheta axei OY
135         plt.xticks([0, 20, 40, 60, 80, 99]) # subdiviziunile de pe axa OX, valorile timpului
136         plt.tight_layout() # aranjarea proportionala a graficelor
137         plt.show() #afisarea graficelor
138
139     grafice_stat=grafice_statice() #redenumim clasa pentru a initializa apelarea graficelor
140     grafice_stat.temperatura() #graficul de temperatura
141     grafice_stat.umiditate() #graficul de umiditate
142     grafice_stat.viteza() #graficul de viteza

```

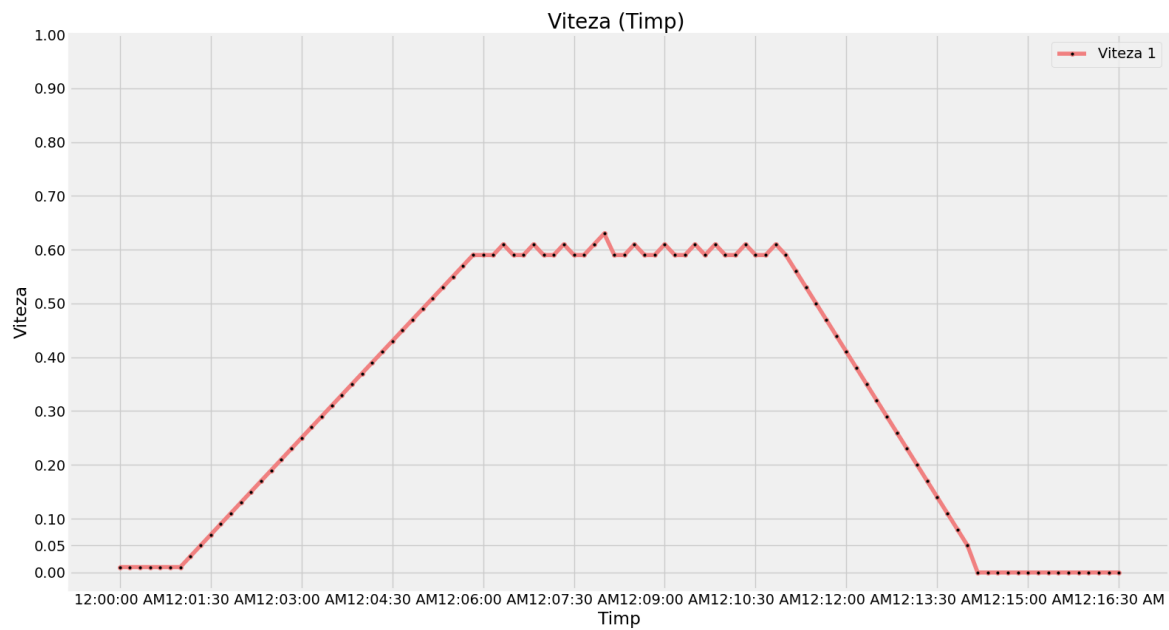
La rularea programului obțin următoarele grafice:



Grafic temperatură

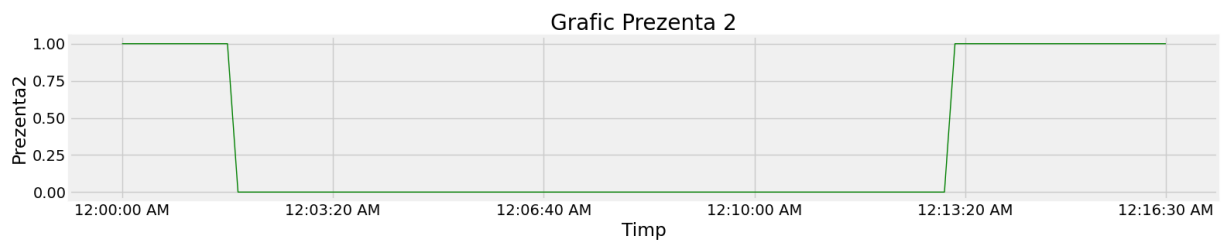
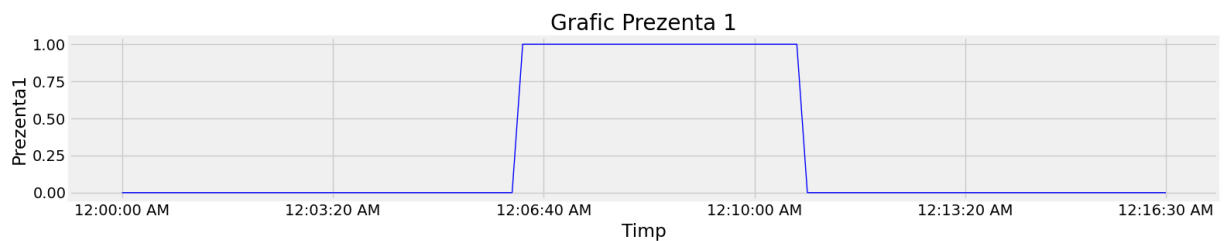


Grafic umiditate



Grafic viteză

Grafice Prezenta



Grafic prezență

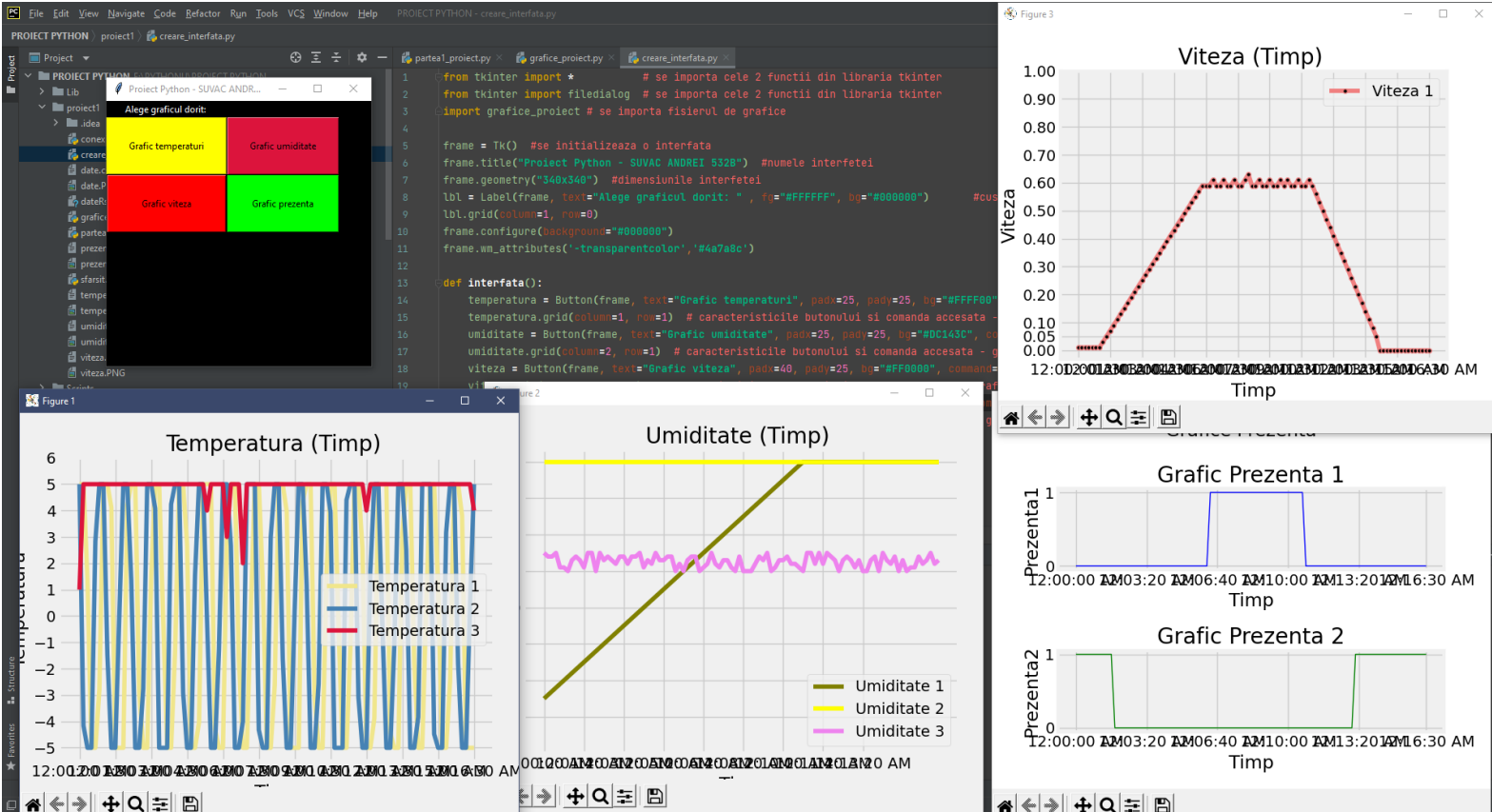
Pentru a aduce o contribuție acestui proiect, m-am gândit să realizez o interfață(GUI) cu ajutorul librăriei tkinter, interfață ce îmi va permite să afișez graficele aferente proiectului la apăsarea unui simplu buton, așa că am încercat să învăț să realizez această interfață cu ajutorul link-urilor din bibliografie, iar acesta este rezultatul final:

```

1  from tkinter import *           # se importa cele 2 functii din libraria tkinter
2  from tkinter import filedialog # se importa cele 2 functii din libraria tkinter
3  import grafice_proiect # se importa fisierul de grafice
4
5  frame = Tk() #se initializeaza o interfata
6  frame.title("Proiect Python - SUVAC ANDREI 532B") #numele interfetei
7  frame.geometry("340x340") #dimensiunile interfetei
8  lbl = Label(frame, text="Alegeți ce grafic doriți să afișați: ", fg="#FFFFFF", bg="#000000") #customizari
9  lbl.grid(column=1, row=0) #customizari
10 frame.configure(background="#000000") #customizari
11 frame.wm_attributes('-transparentcolor', '#4a7a8c') #customizari
12
13 def interfata():
14     temperatura = Button(frame, text="Grafic temperaturi", padx=25, pady=25, bg="#FFFF00", command=grafice_proiect.grafice_stat.temperatura)
15     temperatura.grid(column=1, row=1) # caracteristicile butonului si comanda accesata - grafic temperaturi
16     umiditate = Button(frame, text="Grafic umiditate", padx=25, pady=25, bg="#DC143C", command=grafice_proiect.grafice_stat.umiditate)
17     umiditate.grid(column=2, row=1) # caracteristicile butonului si comanda accesata - grafic umiditate
18     viteza = Button(frame, text="Grafic viteza", padx=40, pady=25, bg="#FF0000", command=grafice_proiect.grafice_stat.viteza)
19     viteza.grid(column=1, row=2) # caracteristicile butonului si comanda accesata - grafic viteza
20     prezenta = Button(frame, text="Grafic prezenta", padx=28, pady=25, bg="#00FF00", command=grafice_proiect.grafice_stat.prezenta)
21     prezenta.grid(column=2, row=2) # caracteristicile butonului si comanda accesata - grafic prezenta
22
23     interfata() #ruleaza programul
24     frame.mainloop() #interfata va rula intr-un loop infinit pana cand o inchidem noi
25

```

Codul pentru creerea interfeței



Interfața în sine și graficele afișate după apăsarea butoanelor

În continuare, am realizat o conexiune prin intermediul protocolului TCP și am reușit prin intermediul unor aplicații să realizăm și conexiunea la interfața RS232(HW Virtual Serial Port&Hercules). Mai jos avem niște informații despre TCP și RS232:

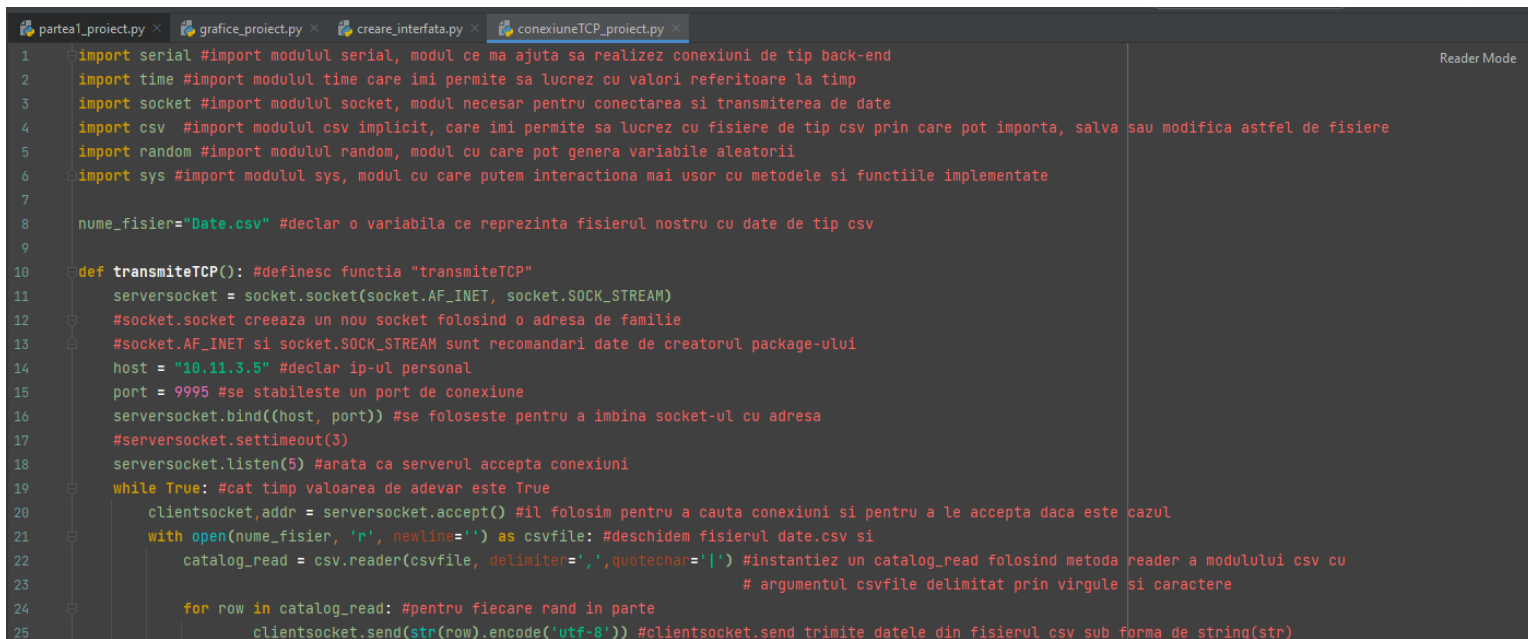
Transmission Control Protocol (sau TCP, în traducere liberă din engleză Protocolul de Control al Transmisiei) este un protocol folosit de obicei de aplicații care au nevoie de confirmare de primire a datelor. Efectuează o conectare virtuală full duplex între două puncte terminale, fiecare punct fiind definit de către o adresă IP și de către un port TCP.

Transmission Control Protocol (TCP) este unul dintre protocoalele de bază ale suitei de protocoale Internet. TCP este unul dintre cele două componente originale ale suitei (celalalt fiind Protocolul Internet, sau IP), astfel încât întreaga suită este frecvent menționată ca stiva TCP/IP. În special, TCP oferă încredere, asigură livrarea ordonată a unui flux de octeți de la un program de pe un computer la alt program de pe un alt computer aflat în rețea. Pe lângă sarcinile sale de gestionare a traficului, TCP controlează mărimea segmentului de date, debitul de informație, rata la care se face schimbul de date, precum și evitarea congestionării traficului de rețea. Printre aplicațiile cele mai uzuale ce utilizează TCP putem enumera World Wide Web (WWW), posta electronică (e-mail) și transferul de fișiere (FTP).

Portul serial sau interfață serială, cunoscut și sub numele de RS-232, este un port de comunicare al calculatoarelor folosit pentru conectarea unor periferice cum ar fi mouse, modem, imprimantă, scanner, microcontroler, și alte echipamente hardware. Porturile seriale sunt folosite și pentru conectarea a două sisteme de calcul într-o rețea peer-to-peer.

Spre deosebire de portul paralel, în cazul portului serial există o singură linie de date, prin care se poate transmite un singur bit la un moment dat. Din acest motiv cât și modulului mai complex de gestiune a comunicației, viteza interfeței seriale este sensibil mai mică decât cea a interfeței paralele. În schimb, portul serial a fost proiectat de la început pentru comunicații bidirecționale.

Realizarea conexiunii:



```
1 import serial #import modulul serial, modul ce ma ajuta sa realizez conexiuni de tip back-end
2 import time #import modulul time care imi permite sa lucrez cu valori referitoare la timp
3 import socket #import modulul socket, modul necesar pentru conectarea si transmiterea de date
4 import csv #import modulul csv implicit, care imi permite sa lucrez cu fisiere de tip csv prin care pot importa, salva sau modifica astfel de fisiere
5 import random #import modulul random, modul cu care pot genera variabile aleatorii
6 import sys #import modulul sys, modul cu care putem interactiona mai usor cu metodele si functiile implementate
7
8 nume_fisier="Date.csv" #declar o variabila ce reprezinta fisierul nostru cu date de tip csv
9
10 def transmiteTCP(): #definesc functia "transmiteTCP"
11     serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12     #socket.socket creeaza un nou socket folosind o adresa de familie
13     #socket.AF_INET si socket.SOCK_STREAM sunt recomandari date de creatorul package-ului
14     host = "10.11.3.5" #declar ip-ul personal
15     port = 9995 #se stabileste un port de conexiune
16     serversocket.bind((host, port)) #se foloseste pentru a imbina socket-ul cu adresa
17     #serversocket.settimeout(3)
18     serversocket.listen(5) #arata ca serverul accepta conexiuni
19     while True: #cat timp valoarea de adevar este True
20         clientsocket,addr = serversocket.accept() #il folosim pentru a cauta conexiuni si pentru a le accepta daca este cazul
21         with open(nume_fisier, 'r', newline='') as csvfile: #deschidem fisierul date.csv si
22             catalog_read = csv.reader(csvfile, delimiter=',', quotechar='"') #instantiez un catalog_read folosind metoda reader a modulului csv cu
23                                     # argumentul csvfile delimitat prin virgule si caractere
24             for row in catalog_read: #pentru fiecare rand in parte
25                 clientsocket.send(str(row).encode('utf-8')) #clientsocket.send trimite datele din fisierul csv sub forma de string(str)
```



```

Wireless LAN adapter Wi-Fi:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : p11.eregie.pub.ro
    Link-local IPv6 Address . . . . . : fe80::65e3:1596:b9d7:b3f6%16
    IPv4 Address. . . . . : 10.11.3.5
    Subnet Mask . . . . . : 255.255.252.0
    Default Gateway . . . . . : 10.11.0.1

PPP adapter Broadband Connection 62:

    Connection-specific DNS Suffix  . :
    IPv4 Address. . . . . : 10.11.4.28
    Subnet Mask . . . . . : 255.255.255.255
    Default Gateway . . . . . : 0.0.0.0

C:\Users\40731>mode

Status for device COM10:
-----
    Baud:             1200
    Parity:            Even
    Data Bits:         7
    Stop Bits:         1
    Timeout:           OFF
    XON/XOFF:          OFF
    CTS handshaking:   OFF
    DSR handshaking:   OFF
    DSR sensitivity:   OFF
    DTR circuit:       ON
    RTS circuit:       ON

Status for device COM7:
-----
    Baud:             1200
    Parity:            Even
    Data Bits:         7
    Stop Bits:         1
    Timeout:           OFF
    XON/XOFF:          OFF
    CTS handshaking:   OFF
    DSR handshaking:   OFF
    DSR sensitivity:   OFF
    DTR circuit:       ON
    RTS circuit:       ON

```

Concluzii

În urma acestui proiect am învățat la un nivel de începător ceea ce presupune limbajul de programare Python și asta îmi dă încredere că pot pe viitor să continui să învăț acest limbaj de programare care este unul de viitor și care cu siguranță este pe placul meu. Am învățat să lucrez cu fișiere de tip .csv, am învățat cum să creez grafice cu ajutorul librăriei matplotlib, am învățat cum să fac o interfață pentru graficele mele cu ajutorul librăriei tkinter, am învățat să realizez o conexiune prin intermediul Python și deasupra tuturor, mi-am dezvoltat o gândire analitică și critică, ceea ce îmi dă speranțe în continuare pentru a lucra în acest program. 14 săptămâni nu sunt suficiente pentru a acumula foarte multă informație, dar consider că este un start bun pentru a învăța Python.

BIBLIOGRAFIE

DOCUMENTAȚIE PROPRIE:

<https://www.w3schools.com/python/default.asp>
<https://www.programiz.com/python-programming>
<https://docs.python.org/3.9/contents.html>
<https://www.udemy.com/>
<https://ro.wikipedia.org/>

CARTE: PYTHON CRASH COURSE(2ND EDITION) DE **ERIC MATTHES**:

<https://www.amazon.com/Python-Crash-Course-Hands-Project-Based/dp/1593276036>

DESPRE FIȘIERE .CSV:

<https://realpython.com/python-csv/>

DOCUMENTAȚIE CREARE GRAFICE :

<https://matplotlib.org/stable/index.html>

DOCUMENTAȚIE CREARE INTERFAȚĂ PENTRU GRAFICE:

<https://pythonguides.com/python-tkinter-colors/>

```
SFÂRȘIT DE PROIECT!  
SUVAC ANDREI-SEBASTIAN  
532B
```

```
Process finished with exit code 0
```