# Documentatie Tema 1 si Tema 2

## Introducere Teoretica

Aceasta documentatie descrie implementarea a doua teme in Python: gestionarea angajatilor si vizualizarea datelor.  
In dezvoltarea celor doua proiecte am folosit principii esentiale ale limbajului Python si biblioteci populare pentru analiza si afisarea datelor.

### Principii Folosite

1. Programare Orientata pe Obiecte (OOP):  
- Mostenire: O clasa copil poate extinde functionalitatea unei clase parinte.  
- Variabile de clasa: Variabile care apartin clasei si sunt comune tuturor obiectelor.  
- Suprascrierea metodelor: O metoda din clasa parinte poate fi redefinita intr-o clasa copil.  
  
2. Gestionarea datelor cu Pandas:  
- Manipularea fisierelor CSV.  
- Selectarea si filtrarea datelor in coloane si randuri.  
  
3. Vizualizarea datelor cu Matplotlib:  
- Generarea de grafice liniare.  
- Personalizarea graficelor cu titluri, etichete si legende.

## Tema 1: Gestionarea Angajatilor si Managerilor

### Teorie Folosita

In Tema 1, am implementat doua clase folosind principii OOP:  
- Clasa Employee: gestioneaza angajatii, contorizeaza numarul acestora si permite adaugarea de task-uri.  
- Clasa Manager: mosteneste Employee, adauga departamentul si suprascrie metoda de afisare in functie de o valoare X.

### Pasi de Implementare

1. Clasele si variabilele:  
- Am creat clasa Employee cu atributul de clasa empCount pentru a numara obiectele.  
- Am adaugat metode pentru afisarea detaliilor si gestionarea task-urilor.  
  
2. Extinderea clasei prin mostenire:  
- Am creat clasa Manager care mosteneste Employee.  
- Am adaugat variabila de clasa mgr\_count pentru contorizarea managerilor.  
- Am implementat suprascrierea metodei display\_employee pentru a afisa informatii diferite in functie de X % 3.  
  
3. Valorile X si Y:  
- X = numarul de litere din numele studentului.  
- Y = numarul de litere din toate prenumele studentului.  
  
4. Crearea si testarea obiectelor:  
- Am creat obiecte Employee si Manager pentru a testa metodele si variabilele de clasa.

### Rezultate Obtinute

Am obtinut o implementare functionala care contorizeaza angajatii si managerii si afiseaza informatii diferite in functie de X:  
- X % 3 == 0: Afiseaza numele.  
- X % 3 == 1: Afiseaza salariul.  
- X % 3 == 2: Afiseaza departamentul.  
  
Numarul total de angajati si manageri este afisat corect.

## Tema 2: Import si Vizualizare Date din CSV

### Teorie Folosita

In Tema 2 am utilizat bibliotecile Pandas si Matplotlib pentru a importa si afisa date dintr-un fisier CSV:  
- Pandas: citeste si filtreaza datele.  
- Matplotlib: genereaza grafice pentru vizualizarea informatiilor.

* extensia Rainbow CSV in editorul Visual Studio Code.  
  Aceasta extensie ofera urmatoarele beneficii:  
  - Evidentiere automata a coloanelor din fisierul CSV, ceea ce faciliteaza citirea si intelegerea datelor.  
  - Detectarea automata a separatorului de campuri (ex: virgula, tab).  
  - Validarea fisierelor CSV pentru a identifica erorile in structura datelor.

### Pasi de Implementare

1. Fisierului CSV:  
- Am primit fisierul data.csv care contine valorile pentru Durata si Puls.  
  
2. Citirea datelor cu Pandas:  
- Am folosit pd.read\_csv() pentru a incarca datele.  
- Am selectat primele 6 valori, ultimele 9 valori si toate valorile folosind slicing.  
  
3. Generarea graficelor cu Matplotlib:  
- Am folosit plt.plot() pentru a afisa datele in grafice liniare.  
- Am personalizat graficele cu titluri, etichete si legende.  
  
4. Crearea unui meniu in consola:  
- Am implementat un meniu pentru a selecta ce date sa fie afisate.  
- Optiunile includ: toate valorile, primele 6 valori si ultimele 9 valori.

### Rezultate Obtinute

Programul permite vizualizarea datelor din fisierul data.csv intr-un mod intuitiv:  
- Graficul pentru toate valorile.  
- Graficul pentru primele 6 valori.  
- Graficul pentru ultimele 9 valori.  
  
Meniul ofera utilizatorului o interfata simpla pentru alegerea optiunilor.

## Concluzie

Prin aceste doua proiecte am invatat sa utilizez conceptele de programare orientata pe obiecte si sa lucrez cu biblioteci externe precum Pandas si Matplotlib. Tema 1 a consolidat cunostintele despre mostenire, suprascriere si variabile de clasa, iar Tema 2 mi-a oferit experienta practica in gestionarea si vizualizarea datelor.

## Bibliografie

1*. Documentatia oficiala Python* - https://docs.python.org/3/   
2. *Pandas User Guide* - https://pandas.pydata.org/docs/   
3. *Matplotlib Documentation* - https://matplotlib.org/stable/contents.html   
4. *Stack Overflow* - Solutii la probleme specifice: https://stackoverflow.com/   
5. *W3Schools Python Tutorial* - https://www.w3schools.com/python/   
6. *Real Python - Tutoriale avansate Python* - https://realpython.com/