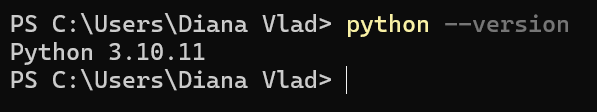
**DOCUMENTAȚIE**

Nume: Vlad Diana-Elena => X=4(număr litere nume), Y=10(număr litere prenume);

**Exercițiul 1: Crearea clasei Manager moștenită din clasa Employee**

**I. Prerequisites & Run**

1. Este necesar ca Python să fie instalat pe calculator. Acest proiect a fost realizat în versiunea 3.10.11.



1. Este necesar un editor de cod precum VSCode.
2. Pentru a rula soluția:
   * Deschideți un terminal în directorul unde se află ex1.py.
   * Rulați comanda: python ex1.py



**II. Explicație cod:**

1. Fișierul ex1.py include două clase:
   * Employee: Definește funcționalități de bază pentru angajați.
   * Manager: Moștenește Employee și adaugă funcționalități suplimentare.
2. Clasa Employee:
   * Atribut de clasă: empCount -> Contorizează numărul de obiecte Employee create.
   * Constructorul \_\_init\_\_: Primește numele (name) și salariul (salary) și inițializează atributele din clasă.
   * Metode:
     + display\_emp\_count: Afișează numărul total de angajați.
     + display\_employee: Afișează numele și salariul unui angajat.

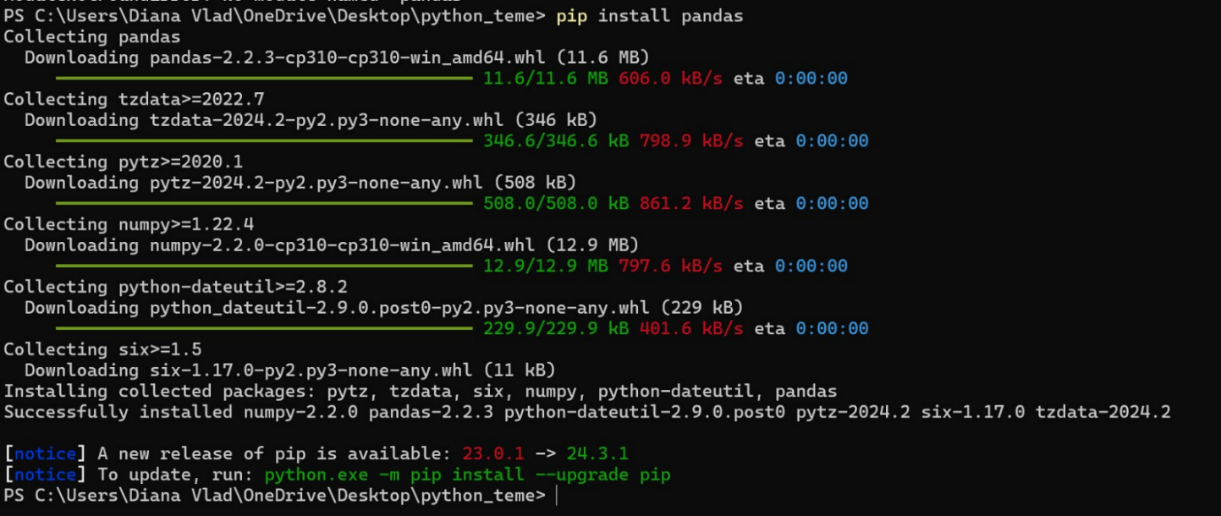
* Destructor \_\_del\_\_: Scade contorul empCount când un obiect Employee este distrus.

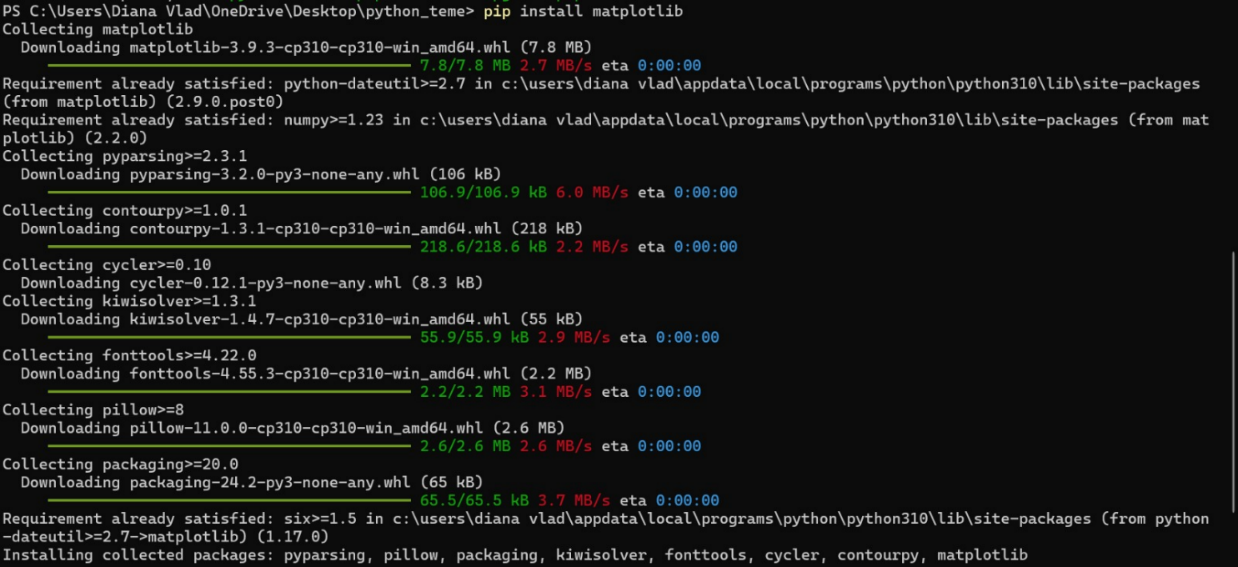
1. Clasa Manager:
   * Atribut de clasă**:** mgr\_count -> Numără obiectele Manager create.
   * Constructorul \_\_init\_\_:
     + Primește ca parametrii name, salary și department.
     + Suprascrie constructorul din clasa Employee (metoda super() permite apelarea constructorului din clasa de bază).
     + Prefixează numele departamentului cu F33-(numele echipei de la disciplină).
     + Incrementează mgr\_count.
   * Suprascriere metodă display\_employee: Afișează doar salariul managerului în implementarea curentă (4%3=1 => am afișat doar salariul).
2. Codul principal (if \_\_name\_\_ == "\_\_main\_\_"):
   * Creează trei obiecte Manager (10/3=3 => 3 obiecte create).
   * Apelează metoda display\_employee pentru fiecare obiect.
   * Creează un obiect Employee.
   * Afișează numărul de manageri și angajați folosind contorii mgr\_count și empCount, evidențiind faptul că numărul acestora diferă din cauza creării obiectului de tip Employee.

**Exercițiul 2: Manipulare și vizualizare date CSV**

**I. Prerequisites & Run**

1. Instalare modulele necesare folosind comanda:

* pip install pandas
* pip install matplotlib



1. Este necesar ca fișierul data.csv să se află în același director cu ex2.py.
2. Pentru rulare:
   * Deschideți terminalul în directorul unde se află ex2.py.
   * Rulați comanda: python ex2.py

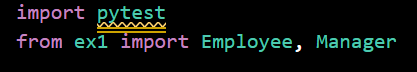


**II. Explicație cod**

1. Importuri:
   * pandas: Pentru manipularea fișierului CSV.
   * matplotlib: Pentru generarea graficelor.
2. Citirea datelor:
   * data = pd.read\_csv('data.csv'): Încarcă conținutul fișierului CSV într-un DataFrame.
3. Grafic 1: Toate valorile
   * data.plot(): Plotează toate coloanele din data pe un singur grafic.
   * plt.show(): Afișează graficul.
4. Grafic 2: Primele X valori
   * data2 = data[:4]: Selectează primele 4 rânduri.
   * plt.plot(data2, marker='o'): Plotează valorile selectate.
5. Grafic 3: Ultimele Y valori pentru coloanele "Durata" și "Puls"
   * data3 = data[["Durata", "Puls"]][-10:]: Selectează ultimele 10 rânduri din coloanele "Durata" și "Puls".
   * plt.plot(data3, marker='\*'): Creează graficul.

**Bonus:**

1. **Funcții de test pentru exercițiul 1**
2. La începutul programului se importă modulele necesare:



* pytest: Este o librărie Python utilizată pentru testarea codului. Pytest trebuie instalat cu comanda: “pip install pytest”.
* Employee și Manager: Clasele care sunt testate. Acestea sunt definite în fișierul ex1.py.

1. Test pentru crearea unui obiect Employee: Verifică corectitudinea creării unui obiect Employee. Pașii sunt:
2. Crearea unui obiect Employee cu numele **"Ioana"** și salariul **5000**.
3. Verificarea atributelor individuale ale obiectului: emp.name == "Ioana";  
    emp.salary == 5000.
4. Verificarea atributului **clasei** Employee.empCount pentru a asigura că numărul total de angajați a fost incrementat corect.
5. Test pentru crearea unui obiect Manager: Verifică corectitudinea creării unui obiect Manager. Pașii sunt:
6. Crearea unui obiect Manager cu numele **"Marin"**,salariul **7000** șidepartamentul **“HR".**
7. Atributul department este verificat (se verifică dacă prefixul F33- este adăugat corect).
8. Se verifică că atributul clasei Manager.mgr\_count a fost incrementat corect.
9. Test pentru verificarea contorilor empCount și mgr\_count: Verifică incrementărea corectă a contorilor. Pașii sunt:
10. Crearea a două obiecte Employee: emp1 cu numele "Ana" și salariul 4000; emp2 cu numele "Maria" și salariul 4500.
11. Crearea unui obiect Manager: mgr1 cu numele "Andrei", salariul 8000 și departamentul "IT".
12. Se verifică contorii: Employee.empCount == 3 (în total, sunt 3 angajați creați).; Manager.mgr\_count == 1 (un singur manager a fost creat).
13. Executarea Testelor:pytest.main() -> Rulează toate funcțiile care încep cu prefixul “test\_”.
14. **Integrare grafice într-o interfață web – exercițiul 2**

* <!DOCTYPE html>: Specifică versiunea HTML utilizată (HTML5 în acest caz).
* <html>: Tag-ul rădăcină care conține toate elementele HTML din pagină.
* <head>: Secțiunea pentru configurări (de exemplu, titlul paginii).
* <title>: Definește titlul paginii, care apare în bara de titlu a browserului.
* <body>: Conține conținutul vizibil al paginii.
* <h1>: Este titlul principal care are cea mai mare dimensiune. Indică utilizatorului că pagina conține o serie de grafice generate.
* <h2>(Textul: "Grafic 1", "Grafic 2" și "Grafic 3"): Desemnează un titlu pentru fiecare grafic.
* Tag-ul Imagine (<img>):

1. src: Specifică sursa imaginii.
2. alt: Text alternativ afișat dacă imaginea nu poate fi încărcată.

**Referințe bibliografice:**

* [***https://www.w3schools.com/python/pandas/default.asp***](https://www.w3schools.com/python/pandas/default.asp)
* [***https://www.w3schools.com/python/matplotlib\_intro.asp***](https://www.w3schools.com/python/matplotlib_intro.asp)
* [**Python - Access List Items**](https://www.w3schools.com/python/python_lists_access.asp)
* [**Python Dictionaries**](https://www.w3schools.com/python/python_dictionaries.asp)
* [**Python Syntax**](https://www.w3schools.com/python/python_syntax.asp)
* [**Python Functions**](https://www.w3schools.com/python/python_functions.asp)
* [**Python Modules**](https://www.w3schools.com/python/python_modules.asp)
* [**Python Inheritance**](https://www.w3schools.com/python/python_inheritance.asp)
* [**Python Classes**](https://www.w3schools.com/python/python_classes.asp)
* [**Python Strings**](https://www.w3schools.com/python/python_strings.asp)
* [**https://www.w3schools.com/html/**](https://www.w3schools.com/html/)
* [**https://www.w3schools.com/html/html\_intro.asp**](https://www.w3schools.com/html/html_intro.asp)
* [**https://www.w3schools.com/html/html\_headings.asp**](https://www.w3schools.com/html/html_headings.asp)
* [**https://www.w3schools.com/html/html\_images.asp**](https://www.w3schools.com/html/html_images.asp)
* [**https://www.w3schools.com/html/html\_basic.asp**](https://www.w3schools.com/html/html_basic.asp)
* [***https://code.visualstudio.com/docs/python/testing***](https://code.visualstudio.com/docs/python/testing)
* [***https://www.tutorialspoint.com/pytest/index.htm***](https://www.tutorialspoint.com/pytest/index.htm)
* [**https://www.youtube.com/watch?v=yYALsys-P\_w&ab\_channel=b001**](https://www.youtube.com/watch?v=yYALsys-P_w&ab_channel=b001)
* [**https://www.youtube.com/watch?v=RSl87lqOXDE&ab\_channel=CoreySchafer**](https://www.youtube.com/watch?v=RSl87lqOXDE&ab_channel=CoreySchafer)
* [**https://matplotlib.org/cheatsheets/cheatsheets.pdf**](https://matplotlib.org/cheatsheets/cheatsheets.pdf)
* [**https://matplotlib.org/stable/api/index.html**](https://matplotlib.org/stable/api/index.html)
* [**https://pandas.pydata.org/docs/user\_guide/index.html#user-guide**](https://pandas.pydata.org/docs/user_guide/index.html#user-guide)