**1. Instalare Python**

Descărcați și instalați **Python** utilizând instrucțiunile de pe pagina oficială:  
<https://www.python.org/>

**2. Instalare Git**

Descărcați și instalați **Git** urmând pașii de pe pagina oficială:  
<https://git-scm.com/>

**3. Descărcarea codului sursă**

Codul sursă se află pe GitHub. Descărcați repository-ul complet de la adresa:  
<https://github.com/IordacheGeani/TemaPython>

Pentru clonare folosind Git, rulați comanda:

git clone https://github.com/IordacheGeani/TemaPython.git

## ****4.Instalarea bibliotecilor necesare****

Pentru a rula codul, este necesar să instalați câteva biblioteci folosind pip. Deschideți un terminal și rulați următoarele comenzi:

### ****Pentru Problema 1****:

pip install pytest

### ****Pentru Problema 2****:

pip install pandas matplotlib

## ****5. Rularea problemelor****

### ****Pentru Problema 1****

1. Accesați directorul Problema1: cd IordacheGeani/Problema1
2. **Rularea programului principal**:

python Problema1.py

1. **Rularea testelor**: Pentru a verifica testele unitare, rulați comanda:

python -m pytest test\_employee\_manager.py

### ****Pentru Problema 2****

1. Accesați directorul Problema2: cd IordacheGeani/Problema2
2. **Rularea programului principal**:  
   Programul va afișa o interfață grafică cu graficele corespunzătoare. Rulați comanda: python vizualizare\_tkinter.py

**Problema 1:**

* **Scop**:  
  Implementarea claselor Employee și Manager, cu testarea funcționalităților folosind pytest.
* **Fișiere implicate**:
  + **employee.py**: Conține clasa Employee cu atributele name, salary și variabila de clasă emp\_count.
  + **Problema1.py**:
    - Definirea clasei Manager care moștenește Employee.
    - Adaugă variabila mgr\_count și prefixul "F10-" pentru departamente.
    - Suprascrie metoda display\_employee pentru a afișa doar departamentul.
  + **test\_employee\_manager.py**: Conține teste pentru:
    - Crearea obiectelor Employee și Manager.
    - Incrementarea corectă a emp\_count și mgr\_count.

**Problema 2**

* **Scop**:  
  Vizualizarea datelor din fișierul data.csv sub formă de grafice, folosind o interfață grafică creată cu Tkinter.
* **Fișiere implicate**:
  + **data.csv**: Fișier CSV cu date pentru grafic (Durata, Puls etc.).
  + **vizualizare\_tkinter.py**:
    - Citește datele din fișierul CSV folosind pandas.
    - Generează grafice folosind matplotlib.
    - Creează o interfață Tkinter cu **3 butoane**:
      1. **Toate valorile**.
      2. **Primele 8 valori**.
      3. **Ultimele 12 valori pentru Durata și Puls**.