# Tema de parcurs

1. *Se considera fisierul employee.py. Atasat.*  
*Creați o clasă care va* moșteni clasa Employee – clasa Manager.

* Numarul de obiecte create din clasa Manger va fi păstrat în variabila de clasă mgr\_count
* Constructorul clasei va avea 3 argumente: name, salary si department
  + La departament veti prefixa numele echipei voastre de la disciplina
* Dacă X%3==0 În clasa Manager, modificați metoda ‘display\_employee’ a.î. să afișeze doar numele angajatului.
* Dacă X%3==1 În clasa Manager, modificați metoda ‘display\_employee’ a.î. să afișeze doar salariul angajatului.
* Dacă X%3==2 În clasa Manager, modificați metoda ‘display\_employee’ a.î. să afișeze doar departamentul angajatului.
* Creați Y/3 obiecte ale clasei Manager. Apelați metoda ‘display\_employee’ pentru toate obiectele din clasa Manager și pentru obiectele din clasa Employee.
* Afișați valoarea atributului emp\_count pentru o instanță a clasei Employee și pentru una a clasei Manager.

2. Se considera fisierul data.csv atasat.

Sa se importe valorile din fisier si sa se ploteze (si afiseze grafic)

\* toate valorile

\* primele X valori

\* ultimele Y valori pentru coloanele Durata si Puls.

*Hint: folositi pandas pentru interactiunea cu fisierul csv si selectia de date pentru diagrame, apoi matplotlib.pyplot pentru afisare.*

*Referinte:*

* [*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)
* *calp.python.extras de la Files > Class.Materials*
* [*https://code.visualstudio.com/docs/python/tutorial-flask*](https://code.visualstudio.com/docs/python/tutorial-flask)
* [*https://github.com/microsoft/python-sample-vscode-flask-tutorial*](https://github.com/microsoft/python-sample-vscode-flask-tutorial)
* [*https://github.com/crchende/ppyan2*](https://github.com/crchende/ppyan2)
* [*https://code.visualstudio.com/docs/python/tutorial-django*](https://code.visualstudio.com/docs/python/tutorial-django)
* [*https://www.w3schools.com/django/index.php*](https://www.w3schools.com/django/index.php)
* [*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)

X = numarul de litere din numele studentului

Y = numarul de litere din (toate) prenumele studentului

Bonificatii posibile:

* Daca solutiile la problema 2 se integreaza pentru vizualizare pe o interfata nativa sau web
* Daca la problema 1 imaginati cateva functii de test si le intregrati in solutie (ex: cu pytest)
* Daca solutiile se incarca pe github, fiecare membru al echipei avand, intr-un branch separat, solutiile intr-un director cu numele sau, in subdirectoare cu numarul problemei,
* Daca gasiti referinte mai bune decat cele precizate si le precizati cadrului didactic.