

# Exerciții Suplimentare OOP Python

Silviu Ojog

***LINKAcademy***

# Exercițiul cu Locuinta

- Se dorește implementarea unei aplicații care să permită gestionarea clienților și a proprietăților imobiliare în cadrul unei agenții imobiliare nou înființate. Agenția imobiliară închiriaza atât case, cât și apartamente. Pentru abstractizarea conceptelor din lumea reală se vor implementa clasele Locuinta, Apartament și Casa.

# Exercițiul cu Locuinta

- Definiti clasa Locuinta, continand urmatoarele date membre private: numeClient(sir de caractere de lungime variabila, None daca locuinta nu este inchiriata), suprafataUtila(intreg), discount(float).
- Sa se supraincarce operator+= si operator= pentru incrementarea si decrementarea discountului, cu asigurarea valorii in intervalul necesar.
- Se va supraincarca operatorul de cast la intreg pentru returnarea suprafetei utile.

# Exercițiul cu Locuinta

- Din clasa Locuinta se vor deriva clasele Apartament, avand in plus campul etaj(intreg) si Casa, avand in plus campurile suprafataCurte(intreg),nrEtaje(intreg) si suprafataEtaje(lista float).
- Testarea se va face folosind obiecte de tipul claselor Apartament si Casa

# Exercițiul cu Locuinta

- Sa se defineasca clasa **AgentieImobiliara** continand o lista obiecte de tip Locuinta si sa se supraincarce operator += si operator -= pentru inserarea, respectiv eliminarea unei locuinte din lista, indiferent de tipul acesteia.
- Se va supraincarca operator **\_\_str\_\_** pentru afisarea clientilor pentru locuintele inchiriate.

# Exercițiul cu Locuinta

- Sa se construiască clasa Gestiune care să stocheze numărul de apartamente sau case, fiecare cu chiria lunară aferentă. Să se construiască o logică de funcționare care să stocheze numărul de clienti, împreună cu numele acestora.

# Exercițiul cu **Votarea**

- Trebuie creată logica de abstractizare a unor alegeri democratice (ex: vot parlamentar). Trebuie luate în considerare următoarele entități: votant, opțiuni de vot, vot, scrutin (alegere) într-un tur sau două tururi de alegere, birou electoral central (BEC)
  - Votul poate fi exprimat o singură dată per alegere
  - Un votantat poate vota de mai multe ori
  - Birou electoral central numără voturile
  - Birou electoral central o listă cu votanți predefiniți

BEC

SCRUTIN

OPTIUNE

VOTANT

VOT



# Exercițiul cu **Factura la Energie**

- Trebuie creată clasa **Factura** care reprezintă factura la energie.
- Atributele necesare sunt: consumul (kWh), costul unui kWh și un cost fix de întreținere pe lună
- Creați un constructor pentru a inițializa atribute. Atenție la tipul lor. Creați setter și getter pentru atributul consum.
- Faceți verificările necesare atributelor Folosiți atributele pentru a calcula costul total al facturii în funcție de atribute. Folosiți `__str__` pentru a afișa acest cost

# Exercițiul cu **Închirierea Camerei**

- Trebuie creată clasa **Rent** care reprezintă închirierea unui camere în regim hotelier. Costul de închiriere este același pentru toate camerele, dar el poate varia în timp.
- Atributele necesare sunt: costul de închiriere, număr de nopți și o taxă de stațiune (de x lei/noapte)
- Creați un constructor pentru a inițializa atribute. Atenție la tipul lor.
- Creați setter și getter pentru numărul de nopți.
- Faceti verificările necesare atributelor
- Folosiți atributele pentru a calcula costul închirierii în funcție de atribute.

# Exercițiul cu Copil

- Pentru clasa Copil cu attributele: nume, alocatie, cnp, numar\_copii,
  - Inițializați constructul cu parametri
  - Operatorul cast catre float pentru a întoarce numele persoanei
  - Operatorul cast catre int pentru a întoarce cod-ul personal
- Testați soluția printr-un exemplu

# Exercițiul cu **Fotografia de Facebook**

- Trebuie creată clasa **FBPhotoView**
- Clasa **FBPhotoView** reprezintă o fotografie de Facebook vizualizată
  - Suprascrieți operatorul+ și operatorul- pentru a aprecia (like) pentru sau respectiv pentru a anula (unlike) aprecierea fotografiei.
    - Testați soluția printr-un exemplu
  - Suprascrieți operatorul de afișare (`__str__`) pentru a arata numărul de aprecieri (like-uri) și dacă a fost sau nu apreciată fotografia de obiectul curent

# Exercițiul cu Calendar

- Creați funcționalitatea unui calendar
  - Vor exista funcții pentru:
    - numărul de zile scurse între două date
    - care va fi data limită, având în vedere o dată de început și număr de zile până la data limită
    - care este data cu x zile în urmă

# Exercițiul cu CardulSTB

- Trebuie creată clasa **STBCard** care reprezintă cardul de la regia autonomă de transport București.
- Atributele necesare sunt: numele deținătorului, călătorii disponibile și creditul disponibil
  - În cazul în care nu primește numele deținătorului, setați-l automat pe "Nenominal"
- Setati valoarea unei călătorii validate cu credit la 3 lei.
- Creați funcții automate pentru validare cu credit/validare călătorie și reîncare credit/reîncarcare călătorii