

# Tema 5 - OOP \_ Clase & Obiecte

# Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează <u>întâlnirea 6</u> și ia notițe în caz că ți-a scăpat ceva.

Pentru toate clasele, creați cel puțin 2 obiecte cu valori diferite și apelați toate metodele clasei.

Te rog să scrii pe canalul de comunicare scrisă unde întâmpini dificultăți și te ajut.

Dacă stai blocat > 30 min, cere indiciu.

# Exerciții obligatorii - grad de dificultate: Usor spre Mediu

### 1.Clasa Cerc

Atribute: raza, culoare

Constructor pentru ambele atribute

## Metode:

- descrie\_cerc() va PRINTA culoarea și raza
- aria() va RETURNA aria
- diametru()
- circumferinta()

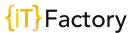
# 2. Clasa Dreptunghi

Atribute: lungime, latime, culoare

Constructor pentru toate atributele

#### Metode:

- descrie()
- aria()
- perimetrul()
- schimbă\_culoarea(noua\_culoare) această metodă nu returneaza nimic. Ea va lua ca și parametru o nouă culoare și va suprascrie atributul



self.culoare. Puteti verifica schimbarea culorii prin apelarea metodei descrie().

# 3. Clasa Angajat

Atribute: nume, prenume, salariu Constructor pt toate atributele

Metode:

- descrie()
- nume\_complet()
- salariu\_lunar()
- salariu\_anual()
- marire\_salariu(procent)

# 4.Clasa Cont

Atribute: iban, titular\_cont, sold

Constructor pentru toate atributele

Metode:

- afisare\_sold() Titularul x are în contul y suma de n lei
- debitare\_cont(suma)
- creditare\_cont(suma)

# Exerciții Opționale - grad de dificultate: Mediu spre greu: may need Google.

### 1. Clasa Factura

Atribute: seria (va fi constantă, nu trebuie constructor, toate facturile vor avea aceeași serie), număr, nume\_produs, cantitate, pret\_buc

Constructor: toate atributele, fara serie

### Metode:

- schimbă\_cantitatea(cantitate)
- schimbă\_preţul(pret)
- schimbă\_nume\_produs(nume)



• generează\_factura() - va printa tabelar dacă reușiți

Factura seria x numar y

Data: generați automat data de azi Produs | cantitate | preț bucată | Total Telefon | 7 | 700 | 49000

Indiciu pt data: <a href="https://www.geeksforgeeks.org/get-current-date-using-python/">https://www.geeksforgeeks.org/get-current-date-using-python/</a>

### 2.Clasa Masina

Atribute: marca, model, viteza maxima, viteza\_actuala, culoare, culori\_disponibile (set), inmatriculata (bool)

Culoare = gri - toate mașinile cand ies din fabrica sunt gri

Viteza\_actuală = 0 - toate mașinile stau pe loc când ies din fabrica

Culori disponibile = alegeți voi 5-7 culori

Marca = alegeți voi - fabrica produce o singură marca dar mai multe modele Inmatriculata = False

Constructor: model, viteza\_maxima

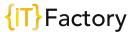
#### Metode:

- descrie() se vor printa toate atributele, în afară de culori\_disponibile
- înmatriculare() va schimba atributul înmatriculată în True
- vopsește(culoare) se va vopsi mașina în noua culoare DOAR dacă noua culoare e în opțiunea de culori disponibile, altfel afișați o eroare
- accelerează(viteza) se va accelera la o anumită viteza, dacă viteza e negativă-eroare, daca viteza e mai mare decat viteza\_max - masina va accelera până la viteza maximă
- franeaza() mașina se va opri și va avea viteza 0

### 3. Clasa TodoList

Atribute: todo (dict, cheia e numele taskului, valoarea e descrierea) La început nu avem taskuri, dict e gol și nu avem nevoie de constructor

Metode:



- adauga\_task(nume, descriere) se va adauga in dict
- finalizează\_task(nume) se va sterge din dict
- afișează\_todo\_list() doar cheile
- afișează\_detalii\_suplimentare(nume\_task) în funcție de numele taskului, printăm detalii suplimentare.
  - Dacă taskul nu e în todo list, întrebam utilizatorul dacă vrea să-l adauge.
  - o Dacă acesta răspunde nu la revedere
  - Dacă răspunde da îi cerem detalii task și salvăm nume+detalii în dict