

Lab Python 1 - Lab 1

Module 8 - Miscellaneous

Note:

Acest laborator cuprinde laboratorul capitolului 8: Miscellaneous.

Obiective:

Exersarea cunoștințelor de iteratori, generatori, functii lambda, map().

Cerinte:

1. Creați un iterator care sa parcurgă un cuvânt si sa genereze următorul output:

```
0: a
1: l
2: f
3: a
4: b
5: e
6: t
```

Functia next() va trebui să returneze o **listă valoare** la fiecare iterație: primul element din listă va fi indexul (i, începând cu 0), iar al doilea element fiind litera curentă din șir.

```
class EnumeratorCuvant():
    def __init__(self, cuvant: str):
        self.cuvant = cuvant
        self.i = 0

    def __iter__(self):
        return self

    def __next__(self):
        # todo
        return valoare

for i, litera in EnumeratorCuvant("alfabet"):
    print(f"{i}: {litera}")
```

2. Se da urmatorul cod:

```
class Masina:
    def __init__(self, marca: str, usi: int, culoare: str, an:
int, pret: float):
        self.marca = marca
        self.usi = usi
        self.culoare = culoare
        self.an = an
        self.__pret = pret

    def getPret(self):
        return self.__pret

masina1 = Masina("Audi", 4, "gri", 2006, 3400)
masina2 = Masina("BMW", 2, "maro", 2007, 4788.60)
masina3 = Masina("Volvo", 4, "gri", 2017, 27000)
masina4 = Masina("Audi", 4, "negru", 2013, 10200)
masina5 = Masina("Audi", 2, "gri", 2005, 3400)
masina6 = Masina("BMW", 4, "negru", 2017, 22000)
masina7 = Masina("Volvo", 4, "gri", 2017, 27000)

masini =
[masina7,masina6,masina5,masina4,masina3,masina1,masina2]
```

Cerinte:

- Calculați și afișați prețul mediu al mașinilor din lista *masini*.
- Afișați numărul de mașini marca BMW din lista cu ajutorul funcției filter
- Creați un generator care sa extraga masinile marca Audi din lista de masini și calculati anul mediu de fabricație a acestora.
- Cereți input de la utilizator cu o marca de masina și afisati cate masini dorite de acesta avem in stoc.

*Observatie: Cat mai putine linii de cod!!

3. Se cere un script care sa calculeze mărimi de salar. Salariile vor fi introduse de către user cu virgula, după care se va cere input cu procentul de majorare și se vor afișa salariile mărite.

Sugestii de implementare:

- Folosiți comprehensiune de liste pentru transformarea valorilor în int
- Folositi metoda map() si dati ca parametru o functie anonima lambda și lista cu salariile inițiale pentru a calcula mărirea în funcție de procentul dat de către user

Output:

Introduceti salariile separate cu virgula: 2300,2577,3200,5400

Introduceti procentul de marire: 8

Salariile marite sunt:

2484.0

2783.16

3456.0

5832.0

4. Creați o clasa Angajat care sa contina:
- O lista pentru toate obiectele de tip Angajat
 - O metoda de clasa care sa adauge toți angajații în lista aceasta, indiferent de numărul de obiecte
 - O metoda de afișare a detaliilor angajaților
 - Un generator care sa extraga angajații unui departament introdus de către utilizator și afișarea acestora
 - Pe baza rezultatului de la punctul trecut, se va cere input de la utilizator cu un procent de mărirea a salariilor angajatilor care fac parte din acel departament
 - Calcularea și afișarea salariilor mărite