

Tema 1 _ Setup, Variabile, Tipuri de date

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 1 și ia notițe în caz că ți-a scăpat ceva.
2. Vizualizează din videoul '*Primii pași în Programare*':
 - Variabile și Tipuri;
 - Operatori și Flow Control.

Astfel, la întâlnirea LIVE deja va fi a 2-a oară când vei auzi conceptele și sigur ți se vor întipări mai bine în minte.

Link: <https://www.itfactory.ro/8174437-intro-in-programare/>

Exerciții obligatorii - grad de dificultate: Ușor spre Mediu:

1. În cadrul unui comentariu, explică cu cuvintele tale ce este o variabilă.
2. Declară și initializează câte o variabilă din fiecare din următoarele tipuri de variabilă :
 - string
 - int
 - float
 - bool

Observație: Valorile vor fi alese de tine după preferințe.

3. Utilizează funcția type pentru a verifica dacă au tipul de date așteptat.
4. Rotunjește 'float'-ul folosind funcția round() și salvează această modificare în aceeași variabilă (suprascrisere):
 - Verifică tipul acesteia.
5. Folosește print() și printează în consola 4 propoziții folosind cele 4 variabile.

Rezolvă nepotrivirile de tip prin ce modalitate dorești.

6. Citește de la tastatură:

- numele;
- prenumele.

Afișează: *'Numele complet are x caractere'*.

7. Citește de la tastatură:

- lungimea;
- lățimea.

Afișează: *'Aria dreptunghiului este x'*.

8. Având stringul: *'Coral is either the stupidest animal or the smartest rock'*:

- afișează de câte ori apare cuvântul 'the';

9. Același string.

- Afișează de câte ori apare cuvântul 'the';
- Printează rezultatul.

10. Același string.

- Folosiți un assert ca să verificați dacă acest string conține doar numere.

Exerciții Opționale - grad de dificultate: Mediu spre greu (s-ar putea să ai nevoie de Google).

1. Exercițiu:

- citește de la tastatură un string de dimensiune impară;
- afișează caracterul din mijloc.

2. Folosind assert, verifică dacă un string este palindrom.

3. Folosind o singură linie de cod :

- citește un string de la tastatură (ex: *'alabala portocala'*);
- salvează fiecare cuvânt într-o variabilă;
- printează ambele variabile pentru verificare.

4. Exercițiu:

- citește un string de la tastatură (ex: *alabala portocala*);
- salvează primul caracter într-o variabilă - indiferent care este el, încearcă cu 2 stringuri diferite;
- capitalizează acest caracter peste tot, mai puțin pentru primul și ultimul caracter => *aAbAlA portocAla*.

5. Exercițiu:

- citește un user de la tastatură;
- citește o parolă;
- afișează: *'Parola pt user x este ***** și are x caractere'*;
- ***** se va calcula dinamic, indiferent de dimensiunea parolei, trebuie să afișeze corect.

eg: parola abc => ***
parola abcd => ****

Tema 2 - Operatori, condiționale

Exerciții Recomandate - grad de dificultate: Ușor .

1. Revizualizează întâlnirea 2 și ia notițe în caz că ți-a scăpat ceva.
2. Vizualizează din '*Primii pași în Programare*' video.
 - Operatori și Flow Control

Astfel, la întâlnirea LIVE deja va fi a 2-a oară când vei auzi conceptele și sigur ți se vor întipări în minte mai bine.

Link: <https://www.itfactory.ro/8174437-intro-in-programare/>

Exerciții obligatorii - grad de dificultate: Ușor spre Mediu .

Pentru toate exercițiile se va folosi noțiunea de if în rezolvare.

Indirect vei exersa și operatorii în cadrul condițiilor ramurilor din if.

X poate fi inițializat direct în cod sau citit de la tastatură, după preferințe.

X este un int.

1. Explică cu cuvintele tale în cadrul unui comentariu cum funcționează un if else.
2. Verifică și afișează dacă x este număr natural sau nu.
3. Verifică și afișează dacă x este număr pozitiv, negativ sau neutru.
4. Verifică și afișează dacă x este între -2 și 13.
5. Verifică și afișează dacă diferența dintre x și y este mai mică de 5.
6. Verifică dacă x NU este între 5 și 27 - a se folosi 'not'.

7.

x și y (int)

Verifică și afișează dacă sunt egale, dacă nu afișează care din ele este mai mare.

8.

X, y, z - laturile unui triunghi.

Afișează dacă triunghiul este isoscel, echilateral sau oarecare.

9. Citește o literă de la tastatură.

Verifică și afișează dacă este vocală sau nu

10. Transformă și printează notele din sistem românesc în >

Peste 9 => A

Peste 8 => B

Peste 7 => C

Peste 6 => D

Peste 4 => E

4 sau sub => F

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

1. Verifică dacă x are minim 4 cifre (x e int).

(ex: 7895 are 4 cifre, 10 nu are minim 4 cifre)

2. Verifică dacă x are exact 6 cifre.

3. Verifică dacă x este număr par sau impar (x e int).

4. x, y, z (int)

Afișează care este cel mai mare dintre ele?

5.

X, y, z - reprezintă unghiurile unui triunghi

Verifică și afișează dacă triunghiul este valid sau nu.

6. Având stringul: 'Coral is either the stupidest animal or the smartest rock'

- Citiți de la tastatură un int x
- Afișează stringul fără ultimele x caractere

Exemplu: dacă ai ales 7 => 'Coral is either the stupidest animal or the smarte'

7. Același string. Declară un string nou care să fie format din primele 5 caractere + ultimele 5

8. Același string.

- Salvează într-o variabilă și afișează indexul de start al cuvântului rock (hint: este o funcție care te ajută să faci asta)
- Folosind această variabilă + slicing, afișează tot stringul până la acest cuvânt
- output: 'Coral is either the stupidest animal or the smartest '

9. Citește de la tastatură un string

Verifică dacă primul și ultimul caracter sunt la fel. Se va folosi un assert

Atenție: se dorește ca programul să fie case insensitive - 'apA' e acceptat

10. Având stringul '0123456789'

- Afișați doar numerele pare
- Afișați doar numerele impare

(hint: folosiți slicing, controlați din pas)

Exerciții Bonus (may need google)

11. Joc ghicit zarul

Caută pe net și vezi cum se generează un număr random

Ne imaginăm că dai cu zarul și salvăm acest număr în dice_roll

Luați un număr ghicit de la utilizator

Verificați și afișați dacă utilizatorul a ghicit

Veți avea 3 opțiuni

- Ai pierdut. Ai ales un număr mai mic. Ai ales x dar a fost y
- Ai pierdut. Ai ales un număr mai mare. Ai ales x dar a fost y
- Ai ghicit. Felicitari! Ai ales x și zarul a fost y

Tema 3 - Structuri de date

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 3 și ia notițe în caz că ți-a scăpat ceva.
2. Vizualizează din '*Primii pași în Programare*' video
 - Structuri de date
 - Flow Control

Astfel, la întâlnirea LIVE deja va fi a 2-a oară când vei auzi conceptele și sigur ți se vor întipări în minte mai bine.

Link: <https://www.itfactory.ro/8174437-intro-in-programare/>

Exerciții obligatorii - grad de dificultate: Usor spre Mediu

Pentru toate exercițiile se va folosi noțiunea de if în rezolvare.

Indirect vei exersa și operatorii în cadrul condițiilor ramurilor din if.

X poate fi inițializat direct în cod sau citit de la tastatură, după preferințe.

X este un int.

1. Declară o listă note_muzicale în care să pui do re mi etc până la do
 - Afișează-o
 - Inversează ordinea folosind slicing și suprascrie această listă.
 - Printează varianta actuală (inversată).
 - Pe această listă aplică o metodă care bănuiești că face același lucru, adică să îi inverseze ordinea. Nu trebuie să o suprascrii în acest caz, deoarece metoda face asta automat.
 - Printează varianta actuală a listei. Practic ai ajuns înapoi la varianta inițială.

Concluzii: slicing e temporar, dacă vrei să păstrezi noua variantă va trebui să suprascrii lista sau să o salvezi într-o listă nouă. Metoda găsită de tine face

suprascrierea automat și permanentizează aceste modificări. Ambele variante își găsesc utilitatea în funcție de ce ne dorim în acel moment.

2. De câte ori apare 'do'?

3. Având 2 liste, [3, 1, 0, 2] și [6, 5, 4]

Găsește 2 variante să le unești într-o singură listă.

4.

- Sortează și afișază lista generată la exercițiul anterior.
- Sortează numărul 0 folosind o funcție.
- Afișaza iar lista.

5. Folosind un if verifică lista generată la exercițiul 3 și afișază:

- Lista este goală.
- Lista nu este goală.

6. Folosește o funcție care să șteargă lista de la exercițiul 3.

7. Copy paste la exercițiul 5. Verifică încă o dată.

Acum ar trebui să se afișeze că lista e goală.

8. Având dicționarul dict1 = {'Ana' : 8, 'Gigel' : 10, 'Dorel' : 5}

Folosește o funcție că să afișezi Elevii (cheile)

9. Printează cei 3 elevi și notele lor

Ex: 'Ana a luat nota {x}'

Doar nota o vei lua folosindu-te de cheie

10. Dorel a făcut contestație și a primit 6

- Modifică nota lui Dorel în 6
- Printează nota după modificare

11. Gigel se transferă din clasă

- Căuta o funcție care să îl șteargă
- Vine un coleg nou. Adaugă Ionică, cu nota 9
- Printează noii elevi

12.

Set

```
zile_sapt = {'luni', 'marți', 'miercuri', 'joi', 'vineri', 'sâmbăta', 'duminică'}
```

```
weekend = {'sâmbăta', 'duminică'}
```

- Adaugă în zilele_sapt 'luni'
- Afișează zile_sapt

13. Folosește un if și verifică dacă:

- Weekend este un subset al zilelor din săptămânii.
- Weekend nu este un subset al zilelor din săptămânii.

14. Afișează diferențele dintre aceste 2 seturi.

15. Afișază intersecția elementelor din aceste 2 seturi.

Exerciții Opționale - grad de dificultate: Mediu spre greu(may need google) .

1. Ne imaginăm o echipă de fotbal pt teren sintetic.

3 Schimbări maxime admise:

- Declară o Listă cu 5 jucători
- Schimbări_efectuate = te joci tu cu valori diferite
- Schimbări_max = 3

Dacă Jucătorul x e în teren și mai avem schimbări la dispoziție

- Efectuează schimbarea

- Șterge jucătorul scos din listă
- Adaugă jucătorul intrat
- Afișaza a intra x, a ieșit y, mai ai z schimbări

Dacă jucătorul nu e în teren:

- Afișază ' nu se poate efectua schimbarea deoarece jucătorul x nu e în teren'
- Afișază 'mai ai z schimbări'

Testează codul cu diferite valori

Google search hint

"how to check if item is in list python"

Tema 4 - Funcții

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 4 și ia notițe în caz că ți-a scăpat ceva.
2. Vizualizează din '*Primii pași în Programare*' video.
 - Flow Control;
 - Funcții.

Astfel, la întâlnirea LIVE deja va fi a 2-a oară când vei auzi conceptele și sigur ți se vor întipări în minte mai bine.

Link: <https://www.itfactory.ro/8174437-intro-in-programare/>

Iterațiile sunt mai dificile deoarece necesită puțină gândire algoritmică. 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. Având lista:

```
mașini = ['Audi', 'Volvo', 'BMW', 'Mercedes', 'Aston Martin', 'Lăstun', 'Fiat', 'Trabant', 'Opel']
```

Folosește un for că să iterezi prin toată lista și să afișezi;

- 'Mașina mea preferată este x'.
- Fă același lucru cu un for each.
- Fă același lucru cu un while.

2. Aceeași listă:

Folosește un for else

În for

- Modifică elementele din listă astfel încât să fie scrise cu majuscule, exceptând primul și ultimul.

În else:

- Printează lista.

3. Aceeași listă:

Vine un cumpărător care dorește să cumpere un Mercedes.

Iterează prin ea prin modalitatea aleasă de tine.

Dacă mașina e mercedes:

Printează *'am găsit mașina dorită de dvs'*

ieși din ciclu folosind un cuvânt cheie care face acest lucru

Altfel:

Printează *'Am găsit mașina X. Mai căutam'*

4. Aceași listă;

Vine un cumpărător bogat dar indecis. Îi vom prezenta toate mașinile cu excepția Trabant și Lăstun.

- Dacă mașina e Trabant sau Lăstun:

Folosește un cuvânt cheie care să dea skip la ce urmează (nu trebuie else).

- Printează S-ar putea să vă placă mașina x.

5. Modernizează parcul de mașini:

- Crează o listă goală, masini_vechi.
- Iterează prin mașini.
- Când găsești Lăstun sau Trabant:
 - Salvează aceste mașini în masini_vechi.
 - Suprascrie-le cu 'Tesla' (în lista inițială de mașini).

- Printează Modele vechi: x.
- Modele noi: x.

6. Având dict:

```
pret_masini = {  
    'Dacia': 6800,  
    'Lăstun': 500,  
    'Opel': 1100,  
    'Audi': 19000,  
    'BMW': 23000  
}
```

Vine un client cu un buget de 15000 euro.

- Prezintă doar mașinile care se încadrează în acest buget.
- Itereaza prin dict.items() și accesează mașina și prețul.
- Printează Pentru un buget de sub 15000 euro puteți alege mașină xLastun
- Iterează prin listă.

7. Având lista:

```
numere = numere = [5, 7, 3, 9, 3, 3, 1, 0, -4, 3]
```

- Iterează prin ea.
- Afișează de câte ori apare 3 (nu ai voie să folosești count).

8. Aceeași listă:

- Iterează prin ea
- Calculează și afișează suma numerelor (nu ai voie să folosești sum).

9. Aceeași listă:

- Iterează prin ea.
- Afișază cel mai mare număr (nu ai voie să folosești max).

10. Aceeași listă:

- Iterează prin ea.
- Dacă numărul e pozitiv, înlocuiește-l cu valoarea lui negativă.

Ex: dacă e 3, să devină -3

- Afișază noua listă.

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

1.

```
alte_numere = [-5, 7, 2, 9, 12, 3, 1, -6, -4, 3]
```

```
numere_pare = []
```

```
numere_impere = []
```

```
numere_pozitive = []
```

```
numere_negative = []
```

Iterează prin listă alte_numere

Populează corect celelalte liste

Afișează cele 4 liste la final

2. Aceeași listă:

Ordonează crescător lista fără să folosești sort.

Te poți inspira vizual de aici.

<https://www.youtube.com/watch?v=lyZQPjUT5B4>

3. Ghicitoare de număr:

numar_secret = Generați un număr random între 1 și 30

Numar_ghicit = None

Folosind un while

 User alege un număr

Programul îi spune:

- Nr secret e mai mare
- Nr secret e mai mic
- Felicitări! Ai ghicit!

4. Alege un număr de la tastatură

Ex: 7

Scrie un program care să genereze în consolă următoarea piramidă

```
1
22
333
4444
55555
666666
7777777
```

Ex:3

```
1
22
333
```

5.

```
tastatura_telefon = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],
    [0]
]
```

Iterează prin listă 2d

Printează 'Cifra curentă este x'

(hint: nested for - adică for în for)

Tema 5 - Funcții

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 5 și ia notițe în caz că ți-a scăpat ceva.
2. Vizualizează din '*Primii pași în Programare*' video.
 - OOP;

Astfel, la întâlnirea LIVE deja va fi a 2-a oară când vei auzi conceptele și sigur ți se vor întipări în minte mai bine.

Link: <https://www.itfactory.ro/8174437-intro-in-programare/>

Pentru toate exercițiile. Apelați funcția de cel puțin 2 ori cu valori diferite pentru a testa. Dacă funcția are return, printați răspunsul

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. Funcție care să calculeze și să returneze suma a două numere
2. Funcție care să returneze TRUE dacă un număr este par, FALSE pt impar
3. Funcție care returnează numărul total de caractere din numele tău complet. (nume, prenume, nume_mijlociu)
4. Funcție care returnează aria dreptunghiului
5. Funcție care returnează aria cercului

6. Funcție care returnează True dacă un caracter x se găsește într-un string dat și False dacă nu găsește.
7. Funcție fără return, primește un string și printează pe ecran:
 - Nr de caractere lower case este x
 - Nr de caractere upper case este y
8. Funcție care primește o LISTA de numere și returnează o LISTA doar cu numerele pozitive
9. Funcție care nu returnează nimic. Primește două numere și PRINTEAZA
 - Primul număr x este mai mare decât al doilea nr y
 - Al doilea nr y este mai mare decât primul nr x
 - Numerele sunt egale.
10. Funcție care primește un număr și un set de numere.
 - Printează 'am adăugat numărul nou în set' + returnează True
 - Printează 'nu am adăugat numărul în set. Acesta există deja' + returnează False

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

1. Funcție care primește o lună din an și returnează câte zile are acea lună
2. Funcție calculator care să returneze 4 valori. Suma, diferența, înmulțirea, împărțirea a două numere.

În final vei putea face:

a, b, c, d = calculator(10, 2)

- print("Suma: ", a)

- `print("Diferenta: ", b)`
- `print("Inmultirea: ", c)`
- `print("Impartirea: ", d)`

3. Funcție care primește o lista de cifre (adică doar 0-9)

Exemplu: [1, 3, 1, 5, 9, 7, 7, 5, 5]

Returnează un DICT care ne spune de câte ori apare fiecare cifră

=> dict {

0: 0

1 :2

2: 0

3: 1

4: 0

5: 3

6: 0

7: 2

8: 0

9: 1

}

4. Funcție care primește 3 numere. Returnează valoarea maximă dintre ele

5. Funcție care să primească un număr și să returneze suma tuturor numerelor de la 0 la acel număr

Exemplu: daca dam nr 3. Suma va fi 6 (0+1+2+3)

Exerciții Opționale - Bonus

1.Funcție care primește 2 liste de numere (numerele pot fi dublate). Returnați numerele comune

Exemplu:

`list1 = [1, 1, 2, 3]`

```
list2 = [2, 2, 3, 4]
```

Răspuns: {2, 3}

2.. Funcție care să aplice o reducere de preț

Dacă produsul costa 100 lei și aplicăm reducere de 10%. Pretul va fi 90

Tratați cazurile în care reducerea e invalidă. De exemplu o reducere de 110% e invalidă.

3. Funcție care să afișeze data și ora curentă din ro

(bonus: afișați și data și ora curentă din China)

4. Funcție care să afișeze câte zile mai sunt până la ziua ta / sau până la

Crăciun dacă nu vrei să ne zici cand e ziua ta :)

Tema 5 - OOP _ Clase & Obiecte

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 6 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 attributele

Metode:

- descrie()
- nume_complet()
- salariu_lunar()
- salariu_anual()
- marire_salariu(procent)

4. Clasa Cont

Atribute: iban, titular_cont, sold

Constructor pentru toate attributele

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 attributele, fara serie

Metode:

- schimbă_cantitatea(cantitate)
- schimbă_prețul(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: <https://www.geeksforgeeks.org/get-current-date-using-python/>

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_actuala = 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, dacă viteza e mai mare decât 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
- `finalizeaza_task(nume)` - se va sterge din dict
- `afiseaza_todo_list()` - doar cheile
- `afiseaza_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.
 - Dacă acesta răspunde nu - la revedere
 - Dacă răspunde da - îi cerem detalii task și salvăm nume+detalii în dict

Tema 7 - OOP _ Cei 4 Piloni

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 7 și ia notițe în caz că ți-a scăpat ceva.

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

Dacă stai blocat > 30 min, cere indiciu.

1. Git setup

Exerciții obligatorii - grad de dificultate: Usor spre Mediu

1. Creează-ți cont de github și încarcă într-un repo nou tot ce am facut până acum la ore.

Curs git/github <https://bit.ly/3yfFvqL>

VIDEOS 1, 2 si 3

2. Rezolvă exercițiul după ce ai urcat proiectul (tot ce am facut până acum împreună).

ABSTRACTION

Clasa abstractă FormaGeometrica

Conține un field PI=3.14

Conține o metodă abstractă aria (opțional)

Conține o metodă a clasei descrie() - aceasta printează pe ecran 'Cel mai probabil am colturi'

INHERITANCE

Clasa Pătrat - moștenește FormaGeometrica

constructor pentru latură

ENCAPSULATION

latura este proprietate privată

Implementează getter, setter, deleter pentru latură

Implementează metoda cerută de interfață (opțional, doar dacă ai ales să implementezi metoda abstractă aria)

Clasa Cerc - moștenește FormaGeometrica

constructor pentru rază

raza este proprietate privată

Implementează getter, setter, deleter pentru rază

Implementează metoda cerută de interfață - în calcul folosește field PI

mostenit din clasa părinte (opțional, doar dacă ai ales să implementezi metoda abstractă aria)

POLYMORPHISM

Definește o nouă metodă descrie - printează 'Eu nu am colturi'

Creează un obiect de tip Patrat și joacă-te cu metodele lui

Creează un obiect de tip Cerc și joacă-te cu metodele lui

3. Actualizează proiectul pe github facand un push la noul cod

În Foldereul de teme ajunge să pui doar linkul cu proiectul tău public

Curs git/github

<https://bit.ly/3yfFvqL> - VIDEO 4

Tema 8 - SELECTORS

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 7 și ia notițe în caz că ți-a scăpat ceva.

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

Alege-ți unul sau mai multe din sugestiile de site-uri de mai jos

- <https://www.phptravels.net/>
- <http://automationpractice.com/index.php>
- <https://formy-project.herokuapp.com/>
- <https://the-internet.herokuapp.com/>
- <https://www.techlistic.com/p/selenium-practice-form.html>
- [jules.app](https://julesapp.com/)

Alege câte 3 elemente din fiecare tip de selector din următoarele categorii:

- Id
- Link text
- Parțial link text
- Name
- Tag*
- Class name*
- Css (1 după id, 1 după clasă, 1 după atribut=valoare_partiala)

observație:

- Probabil nu vei găsi un singur website care să cuprindă toate variantele de mai sus, astfel că îți recomandăm să folosești mai multe site-uri

- Opțional: La tag și class name vei folosi find elementS! - salvează în listă. Interacționează cu un element la alegere din listă.

Pentru xpath identifică elemente după criteriile de mai jos:

- 3 după atribut valoare
- 3 după textul de pe element
- 1 după parțial text
- 1 cu SAU, folosind pipe |
- 1 cu *
- 1 în care le iei ca pe o listă de xpath și în python ajunge 1 element, deci cu (xpath)[1]
- 1 în care să folosești parent::
- 1 în care să folosești fratele anterior sau de după (la alegere)
- 1 funcție ca și cea de la clasă prin care să pot alege eu prin parametru cu ce element vreau să interacționez.

Exerciții extra - Opțional

<https://www.automatetheplanet.com/most-exhaustive-xpath-locators-cheat-sheet/>

Tema 9 - VERIFICATORI

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 8 și ia notițe în caz că ți-a scăpat ceva.

Exerciții obligatorii - grad de dificultate: Usor spre Mediu

Implementează o clasă Login care să moștenească unittest.TestCase

Gasește elementele în partea de sus folosind ce selectors dorești:

- setUp()
- Driver
<https://the-internet.herokuapp.com/>

Click pe Form Authentication

tearDown()

Quit browser

- Test 1
 - Verifică dacă noul url e corect
- Test 2
 - Verifică dacă page title e corect
- Test 3
 - Verifică textul de pe elementul xpath=//h2 e corect
- Test 4
 - Verifică dacă butonul de login este displayed

- Test 5
 - Verifică dacă atributul href al linkului 'Elemental Selenium' e corect
- Test 6
 - Lasă goale user și pass
 - Click login
 - Verifică dacă eroarea e displayed
- Test 7
 - Completează cu user și pass invalide
 - Click login
 - Verifică dacă mesajul de pe eroare e corect
 - Este și un x pus acolo extra așa că vom folosi soluția de mai jos

```
expected = 'Your username is invalid!'
self.assertTrue(expected in actual, 'Error message text is incorrect')
```
- Test 8
 - Lasă goale user și pass
 - Click login
 - Apasă x la eroare
 - Verifică dacă eroarea a dispărut
- Test 9
 - Ia ca o listă toate //label
 - Verifică textul ca textul de pe ele să fie cel așteptat (Username și Password)
 - Aici e ok să avem 2 asserturi
- Test 10
 - Completează cu user și pass valide
 - Click login
 - Verifică ca noul url CONTINE /secure
 - Folosește un explicit wait pentru elementul cu clasa 'flash succes'
 - Verifică dacă elementul cu clasa='flash succes' este displayed

- Verifică dacă mesajul de pe acest element CONȚINE textul 'secure area!'
- Test 11
 - Completează cu user și pass valide
 - Click login
 - Click logout
 - Verifică dacă ai ajuns pe <https://the-internet.herokuapp.com/login>

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

- Test 12 - brute force password hacking
 - Completează user tomsmith
 - Găsește elementul //h4
 - la textul de pe el și fă split după spațiu. Consideră fiecare cuvânt ca o potențială parolă.
 - Folosește o structură iterativă prin care să introduci rând pe rând parolele și să apeși pe login.
 - La final testul trebuie să îmi printeze fie
'Nu am reușit să găsesc parola'
'Parola secretă este [parola]'

Tema 10 - VERIFICARI EXTRA

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 9 și ia notițe în caz că ți-a scăpat ceva.

Exerciții obligatorii - grad de dificultate: Usor spre Mediu .

1. Faceti o suita care sa contina testele voastre de la tema 9 + testele de la intalnirea 10 (care au clasa). Generati raportul
2. Ganditi voi o clasa de test din paginile sugerate in tema 8
 - Scrieti cel putin 3 functii de test intr-o clasa (cum am facut la clasa)
Folositi firefox in loc de chrome (ce doriti voi, cate functii de test doriti, freestyle ca sa incepeti sa ganditi si singuri niste scenarii de test).

Bine de stiut: o functie de test este echivalentul unui test case

Mai multe aici:

https://www.tutorialspoint.com/software_testing_dictionary/test_case.htm

<https://www.phptravels.net/>

<http://automationpractice.com/index.php>

<https://formy-project.herokuapp.com/>

<https://the-internet.herokuapp.com/>

<https://www.techlistic.com/p/selenium-practice-form.html>

Sau puteti alege voi ce pagina doriti

Tema 11 - BDD, POM

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 10 și ia notițe în caz că ți-a scăpat ceva.

Exerciții obligatorii - grad de dificultate: Usor spre Mediu .

Avand ca exemplu pagina <https://the-internet.herokuapp.com/login>

Impementati 3 pagini intr-un nou proiect BDD cu POM

- a. Home page <https://the-internet.herokuapp.com/>
 - Sa aiba cel putin 3 elemente inclusiv Form Authentication
 - Sa contina metode de click pe ele
- b. Login page <https://the-internet.herokuapp.com/login>
 - Sa contina user, pass, login_btn si metode pt interactiune cu ele
- c. Secure page <https://the-internet.herokuapp.com/secure>
 - Sa contina mesajul de succes si verificare ca e displayed
 - Sa contina logout_btn si click pe el

Tema 12 - Final Project

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 11 și ia notițe în caz că ți-a scăpat ceva.
2. Recap:
 - Pages (elem + metoda de interactivitate)
 - Environment: punem în context un obiect de tipul paginii
 - Steps: Legăm metodele din pages de o propoziție gherkin
 - Features: Aducem cu copy paste din steps pașii necesari. Putem aduce din fișiere multiple, după nevoie

Exerciții obligatorii - grad de dificultate: Usor spre Mediu

În continuarea proiectului de testare a paginii Jules, implementați 3 teste:

1. Test 1
 - Open jules.app
 - Input correct email
 - Leave pass empty
 - Verify error 'Please enter your password!' is displayed
 - Verify Log in btn is disabled
2. Test 2
 - Base page: implementează metoda de verificare url care primește ca si param url-ul paginii
 - Open jules.app
 - Click sign up
 - Verify url folosind metoda din base page: <https://jules.app/sign-up>
 - Click login
 - Verify url folosind metoda din base page: <https://jules.app/sign-in>

- Hint: in steps vei avea obiectul signUpPage() care mosteneste din BasePage(), astfel vei avea acces la metoda definita in BasePage(), numita de exemplu verifyUrl(url).

Avantajul este ca definim aceasta functie o singura data, in BasePage si ptem sa o apelam din orice alta pagina cand avem nevoie (fara sa duplicam de n ori aceasta logica)

3. Test 3

- Open jules.app
- Click sign up
- Click Persona
- Click Continue
- Input correct first name
- Click Continue
- Input correct last name
- Click Continue
- Input wrong email
- Verify msg is displayed "Please enter a valid email address."
- Clear email input
- Input correct email
- Verify that error is not displayed anymore ("Please enter a valid email address.")

Tema 13 - Final Project 2

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 12 și ia notițe în caz că ți-a scăpat ceva.

Exerciții obligatorii - grad de dificultate: Usor spre Mediu

1. Finalizati tema 12 (daca nu ati apucat)
2. Pentru proiectul de la intalnirea 13, creati un repo in github si urcati sus codul
3. Extindeti proiectul facut la clasa cu cel putin inca un test

*Instalati Postman pe data viitoare: <https://www.postman.com/downloads/>
Open and create account. (Ne trebuie pt a intelege ce e un API)*

Tema 14 - Unit Testing

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 13 și ia notițe în caz că ți-a scăpat ceva.

Exerciții obligatorii - grad de dificultate: Usor spre Mediu .

La tema 6 ati avut de facut clase.

1. **Obligatoriu** UNIT TESTS pt ex2 - clasa Dreptunghi

- Clasa Dreptunghi
- Atribute: lungime, latime, culoare
- Constructor pt toate attributele
- Metode:
- descrie()
- aria()
- perimetrul()
- schimba_culoarea(noua_culoare) - aceasta metoda nu returneaza nimic. Ea va lua ca si param o noua culoare si va suprascrie atributul self.culoare. Puteti verifica schimbarea culorii prin apelarea metodei descrie().

1. Unit tests pt toate metodele

- Cum poti testa totusi schimba culoarea daca nu returneaza nimic?
- Fie verificati direct valoarea atributului, fie verific ce returneaza metoda descrie()

OPTIONAL (sau dupa terminarea cursului pentru a extinde framework de unit tests)

- Unit tests pt clasele Cerc, Cont si Angajat

2. OBLIGATORIU:

- a. Creati o noua colectie in Postman, folosind acest api: <https://reqres.in/>
- b. Params ca url params - deci ii veti pune in tabule de params la rubricile Key si Value, astfel se va forma un url de genul
<https://reqres.in/api/users?name=Andy&job=trainer>
- c. Salvati, trimiteti urmatoarele requestur si studiat raspunsul (http code si jsoni):
 - Get all users (list users) -> 200
 - Get user (single user) -> 200
 - Get user inexistent (single user not found) -> 404
 - Create User (post) -> 201
 - Update User (patch) -> 200
 - Delete user -> 204
 - Successful login -> 200
 - Unsuccessful login (fara pass param) -> 404
 - Pentru verificare din optiunile colectiei (...) dati export si la teme incarcati fisierul exportat

OPTIONAL:

Puteti sa adaugati si sa va jucati si cu restul metodelor

Tema 15 - API Testing

Exerciții Recomandate - grad de dificultate: Ușor

1. Revizualizează întâlnirea 14 și ia notițe în caz că ți-a scăpat ceva.

Exerciții obligatorii - grad de dificultate: Usor spre Mediu .

Continuati/finalizati framework de testare automata API - Simple Books API
Dupa ce terminati urcati proiectul pe github intr-un repo nou 'API Testing Framework'

<https://github.com/vdespa/introduction-to-postman-course/blob/main/simple-books-api.md>