

SEMINAR 1+2 - FUNDAMENTELE PROGRAMĂRII

Creați o aplicație care gestionează o listă de numere, care permite următoarele:

1. Citirea unei liste de numere.
 2. Adăugarea unui număr la listă.
 3. Căutarea și afișarea celui mai mic număr din listă și a poziției pe care se regăsește acesta.
 4. Eliminarea numerelor prime din listă.
 5. Verificarea listei pentru proprietatea de mulțime. În caz că nu este mulțime, afișarea frecvenței fiecărei valori.
- E. Ieșirea din aplicație.

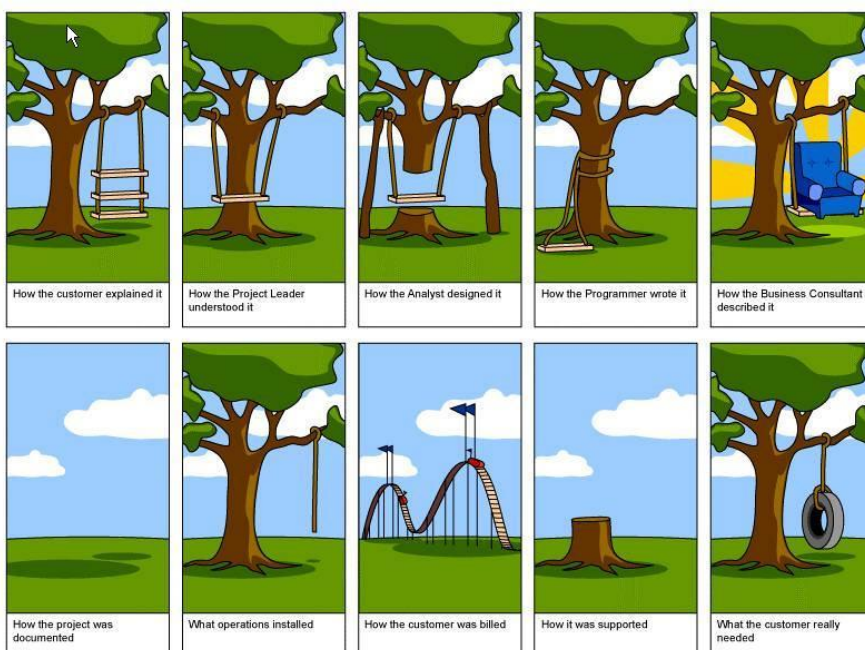
Folosiți un proces de dezvoltare incrementală bazată pe funcționalități (feature-driven development).

Ce ne propunem pe azi:

- ✧ Exersare scris codPython: lucru cu numere, string-uri, liste, dicționare, etc...
- ✧ Discutat concepte: **mutable** vs. **Immutable**



- ✧ Discutat aspecte dezvoltare aplicații software

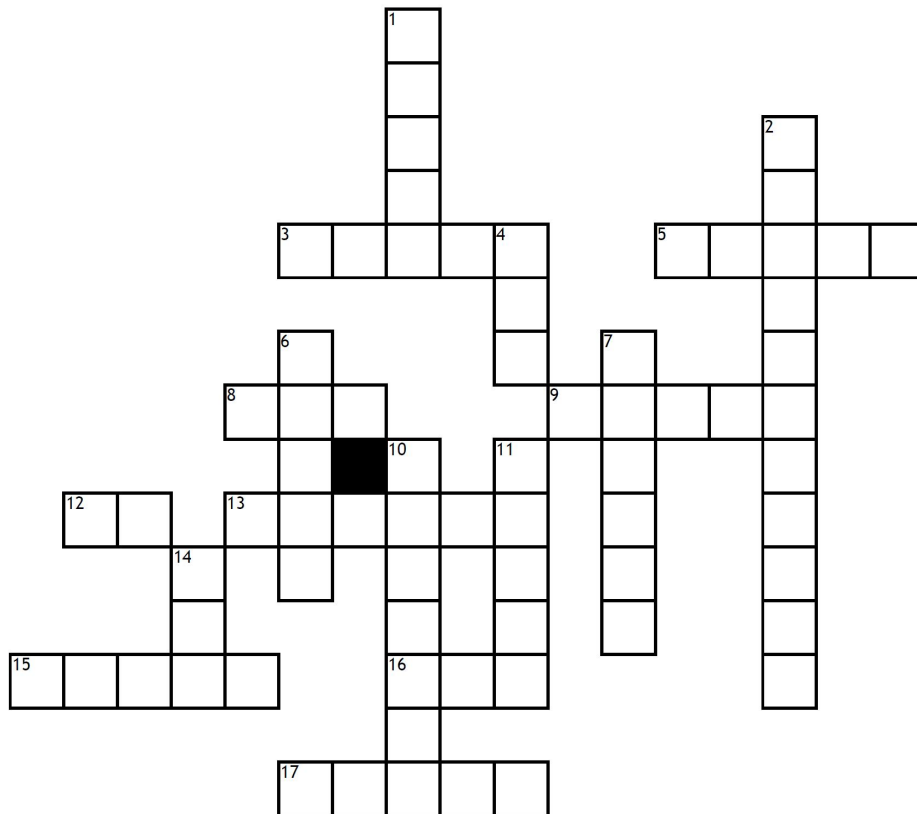


Remembering what FDD entails:

1. Enunț problemă ☒
2. Identificare funcționalități
3. Descriere scenarii de rulare
4. Identificare task-uri (activități) pe baza scenariului
5. Testare (tabel cu intrări - ieșirea așteptată)
6. Implementare

I. Asociați fiecare element din coloana din stânga cu item-ul corect din coloana din dreapta. Pot exista asocieri între un item din stânga și mai mulți itemi din dreapta. Pot rămâne itemi din dreapta fără asocieri.

- | | |
|---|------------------------------------|
| 1. Se poate schimba la obiectele <i>mutable</i> fără schimba identitatea obiectului | A. Tuplu (tuple) |
| 2. Accesarea elementului de pe poziția dată dintr-o listă | B. Listă (list) |
| 3. Tipul de element dintr-un dicționar | C. 4 |
| 4. <code>type({1, 2, 3}) == set</code> | D. Pereche cheie-valoare |
| 5. (1, 3, 5, 19) reprezintă un | E. Valoarea |
| 6. int, float, boolean, string, tuplu | F. Mulțime (set) |
| 7. Tip de dată <i>mutable</i> | G. str |
| 8. Dacă a = "1", tipul de dată pentru a este ... | H. 2 |
| 9. <code>isinstance(7.12, int)</code> | I. False |
| 10. <code>d = {1: "Mara", 2: "Adrian"}</code>
<code>print(len(d))</code> | J. Operatorul de indexare ([]) |
| | K. True |
| | L. Tipuri de date <i>immutable</i> |



Across

3. Elimină spațiile de la începutul și sfârșitul unui string.

5. `list(____(0, 4)) == [0, 1, 2, 3]` se evaluează ca fiind True.

8. `x = ____ (5,4)`, iar `x = 625`.

9. Cu ajutorul acestei metode putem împărți un string dat în mai multe substring-uri, delimitate de un separator dat.

12. Returnează „identitatea” unui obiect, un număr garantat a fi unic și constant pentru un obiect dat.

13. Pentru a obține lista `[1, 9, 10, 5]` din lista `[1, 10, 5]` avem nevoie de instrucțiunea _____

15. `fruits = ['cherry', 'banana', 'cherry', 'cherry', 'raspberry']`.
`print(fruits._____('cherry'))` afișează valoarea 3.

16. Returnează valoarea absolută a unui număr dat.

17. `d = {'red_apples': 20.1, 'green_apples': 8.25, 'grapes': 37.25}`. În dicționarul `d`, `'red_apples'` reprezintă o _____

Down

1. Transformă toate caracterele dintr-un string în uppercase.

2. Dacă `a = '1'`, `b = '2'`, prin operația `a + b` se realizează o _____.

4. Pentru a șterge un element de la o poziție dată dintr-o listă folosim metoda _____. Are un parametru opțional; dacă acesta nu este dat, metoda elimină elementul de pe ultima poziție din listă.

6. Returnează numărul rotunjit. Se poate specifica o anumită precizie (număr de cifre după virgulă).

7. Metodă prin care adăugăm un element la sfârșitul unei liste.

10. Înlocuiește un substring dat dintr-un string cu altul.

11. Metodă care returnează perechile (cheie, valoare) care se află într-un dicționar dat.

14. Metodă care returnează numărul de elemente ale unei liste.