****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**Calancea Catalin**

**MI-222**

**Raport**

**pentru lucrarea de laborator Nr.3**

**Tema: Colectii de date**

***la cursul de “Programare Interactivă”***

Verificat:

lector universitar

Strună Vadim

**Chișinău – 20****23**

**CUPRINS**

[1.INTRODUCERE 3](#_Toc150768669)

[2.SARCINA 4](#_Toc150768671)

[3.SCRIPT 6](#_Toc150768672)

[4.CONCLUZIA 9](#_Toc150768673)

[4.WEBOGRAFIE 10](#_Toc150768674)

# 1.INTRODUCERE

# Lucrarea de laborator propusă are ca obiectiv explorarea și înțelegerea detaliată a colecțiilor de date în limbajul de programare, cu accent pe liste, tupluri, seturi și dicționare. Aceste colecții reprezintă structuri de date esențiale în dezvoltarea programelor și oferă posibilități variate pentru gestionarea și organizarea informațiilor într-un mod eficient.

# Obiectivele principale ale acestei lucrări includ:

# 1. Studiul colecțiilor de date:

# - Liste: Structuri de date secvențiale și mutable, ce permit stocarea unui șir de elemente. Vom explora modurile de accesare, modificare și extindere a listelor.

# - Tupluri: Similar cu liste, dar imutabile, tuplurile sunt utile în situații în care se dorește o colecție de elemente care nu ar trebui să fie modificate. Vom examina caracteristicile și utilizările tuplurilor.

# - Seturi: Colecții neordonate și mutable de elemente unice. Vom investiga modurile de adăugare, ștergere și operații specifice seturilor.

# - Dicționare: Colecții de date cheie-valoare, care permit stocarea și accesarea eficientă a informațiilor. Vom analiza structura, manipularea și beneficiile dicționarelor în programare.

# 2. Metodele aplicabile tuplurilor, seturilor și dicționarelor:

# - Vom explora funcționalitățile specifice oferite de limbajul de programare pentru a lucra cu tupluri, seturi și dicționare.

# - Pentru tupluri, ne vom concentra pe utilizarea și manipularea acestora în contextul imutabilității lor.

# - Pentru seturi, vom analiza operațiile de bază, precum adăugarea, ștergerea și operațiile de set.

# - În ceea ce privește dicționarele, vom explora metodele de adăugare și accesare a elementelor, precum și gestionarea cheilor și valorilor.

# Prin această lucrare de laborator, ne propunem să dezvoltăm competențe solide în manipularea colecțiilor de date, înțelegând avantajele și limitările fiecăreia în contextul programării. Aceste cunoștințe vor constitui un fundament esențial pentru abordarea eficientă a problemelor practice și dezvoltarea de soluții software robuste.

# 2.SARCINA

**I. Studiați domeniul de subiect al opțiunii și gândiți-vă la ce informații ar trebui stocate despre obiectul cheie (de exemplu, pentru un tarif telefonic, acesta este numele, costul, tipul etc.).** **creați o aplicație mică care stochează informații despre un obiect din domeniul selectat și le afișează pe ecran.**

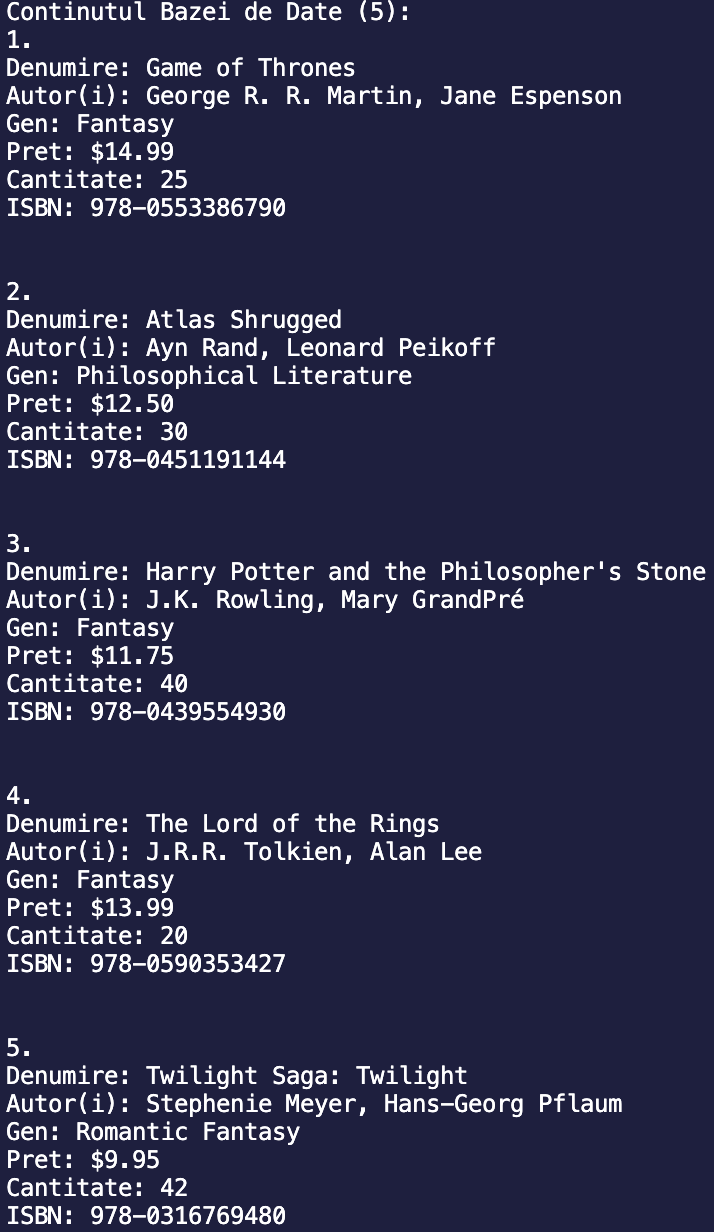
Cerințe de date:

- numărul de obiecte (elementele dicționarului) trebuie să fie de cel puțin 3;

- numărul atributelor obiectului trebuie să fie de cel puțin 5;

- cel putin 3 atribute obiect trebuie sa fie de diferite tipuri: int, float, str, etc.; - cel puțin 1 atribut al obiectului trebuie să fie de tip listă.

Varianta : Librărie O librărie folosește un sistem care stochează informații despre cărți: titlu, preț, cantitate, lista autorilor, ISBN etc.



**II.**

**1. Descrieți o structură denumită AEROFLOT, care conține următoarele câmpuri: DEST - denumirea destinației zborului;**

**NUMR - numărul zborului; SFAT - tipul aeronavei.**

**2. Scrieți un program care efectuează următoarele acțiuni:**

**introducerea datelor de la tastatură în matricea AIRPORT, constând din șapte elemente de tip AEROFLOT; intrările trebuie să fie ordonate în ordinea crescătoare a numărului zborului;**

**afișarea Niciun zbor și tipuri de aeronave care zboară la destinație, al căror nume a coincis cu numele introdus de la tastatură;**

**dacă nu există astfel de zboruri, afișați un mesaj corespunzător pe afișaj.**

# 3.1 SCRIPT CONDITIE I

book\_db = [

{

"title": "Game of Thrones",

"authors": ["George R. R. Martin", "Jane Espenson" ],

"genre": "Fantasy",

"price": 14.99,

"quantity": 25,

"isbn": "978-0553386790"

},

{

"title": "Atlas Shrugged",

"authors": ["Ayn Rand","Leonard Peikoff"],

"genre": "Philosophical Literature",

"price": 12.50,

"quantity": 30,

"isbn": "978-0451191144"

},

{

"title": "Harry Potter and the Philosopher's Stone",

"authors": ["J.K. Rowling" , "Mary GrandPré"],

"genre": "Fantasy",

"price": 11.75,

"quantity": 40,

"isbn": "978-0439554930"

},

{

"title": "The Lord of the Rings",

"authors": ["J.R.R. Tolkien" , "Alan Lee"],

"genre": "Fantasy",

"price": 13.99,

"quantity": 20,

"isbn": "978-0590353427"

},

{

"title": "Twilight Saga: Twilight",

"authors": ["Stephenie Meyer" , "Hans-Georg Pflaum"],

"genre": "Romantic Fantasy",

"price": 9.95,

"quantity": 42,

"isbn": "978-0316769480"

}

]

print("Continutul Bazei de Date ({}):".format(len(book\_db)))

for i, book in enumerate(book\_db, start=1):

print("{}.".format(i))

print("Denumire: {}".format(book["title"]))

print("Autor(i): {}".format(", ".join(book["authors"])))

print("Gen: {}".format(book["genre"]))

print("Pret: ${:.2f}".format(book["price"]))

print("Cantitate: {}".format(book["quantity"]))

print("ISBN: {}".format(book["isbn"]))

print("\n")

# 3.2 SCRIPT CONDITIE II

import json

class AEROFLOT:

def \_\_init\_\_(self, DEST, NUMR, TIP):

self.DEST = DEST

self.NUMR = NUMR

self.TIP = TIP

*# Introduceți datele în matricea AIRPORT și sortați după numărul de zbor crescător*

AIRPORT = []

for i in range(2):

DEST = input(f"Introduceți numele destinației zborului dvs {i + 1}: ")

numr = int(input(f"Introduceți numărul zborului {i + 1}: "))

tip = input(f"Introduceți tipul de aeronavă pentru zbor {i + 1}: ")

aeroflot\_instance = AEROFLOT(DEST, numr, tip)

AIRPORT.append(aeroflot\_instance)

*# Sortați după numărul de zbor crescător*

AIRPORT.sort(key=lambda x: x.NUMR)

*# Salvarea datelor într-un fișier JSON*

with open('airport\_data.json', 'w') as json\_file:

json.dump([vars(a) for a in AIRPORT], json\_file, ensure\_ascii=False, indent=4)

*# Introducerea numelui destinației de căutat*

search\_DEST = input("Introduceți numele destinației dvs. de căutare: ")

*# Căutați zboruri după numele destinației potrivite*

found\_flights = [(a.NUMR, a.TIP) for a in AIRPORT if a.DEST == search\_DEST]

*# Rezultate de ieșire*

if found\_flights:

print(f"Zboruri și tipuri de aeronave către destinația dvs '{search\_DEST}':")

for numr, tip in found\_flights:

print(f"Număr zbor: {numr}, tipul aeronavei: {tip}")

else:

print(f"Nu au fost găsite zboruri către destinația „{search\_DEST}”.")

# 4.CONCLUZIA

În cadrul acestei lucrări de laborator, am explorat și am înțeles în profunzime colecțiile de date fundamentale în limbajul de programare, cu un focus deosebit pe liste, tupluri, seturi și dicționare. Am descoperit caracteristicile distincte ale fiecărei colecții și am învățat cum să aplicăm metode specifice pentru a manipula și gestiona eficient datele în cadrul acestora.

Listele, ca structuri secvențiale și mutable, oferă flexibilitate în gestionarea unui șir de elemente, pe când tuplurile, prin imutabilitatea lor, aduc stabilitate și siguranță în anumite contexte de programare. Seturile, cu capacitatea lor de a stoca elemente unice într-o ordine nedefinită, se dovedesc utile în situații ce implică verificarea și gestionarea diversității de date. În același timp, dicționarele oferă o modalitate eficientă de a organiza informațiile prin perechi cheie-valoare, facilitând accesul rapid la datele relevante.

În cadrul studiului metodelor aplicabile acestor colecții, am acumulat cunoștințe practice esențiale. Am învățat cum să manipulăm și să gestionăm tuplele, seturile și dicționarele, utilizând operațiuni specifice precum adăugarea, ștergerea, accesarea și modificarea elementelor. Aceste abilități sunt cruciale în dezvoltarea programelor eficiente și scalabile.

În concluzie, această lucrare de laborator a furnizat un cadru solid pentru înțelegerea și aplicarea colecțiilor de date în programare. Competențele dobândite în manipularea listelor, tuplurilor, seturilor și dicționarelor vor constitui un fundament esențial în abordarea eficientă a problemelor practice și vor contribui la dezvoltarea de soluții software robuste și flexibile.

# 4.WEBOGRAFIE

1. Python Documentation. Disponibil online la: https://docs.python.org/

2. Stack Overflow. Comunitatea programatorilor. Disponibil online la: https://stackoverflow.com/

3. W3Schools. Tutoriale Python. Disponibil online la: https://www.w3schools.com/python/

4. GeeksforGeeks. Resurse Python. Disponibil online la: https://www.geeksforgeeks.org/python-programming-language/