

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

**gr. IA-231, Chistol Maxim**

**Raport**

**pentru lucrarea de laborator Nr.3**

***la cursul de “Programarea Interactivă”***

Verificat:

**Alexei Victoria,** asistent.universitar.

Departamentul Informatică şi IS,

Facultatea FCIM, UTM

**Chișinău 2024**

**Tema:** Colecții de date.

**Obiective:**

• Studiul colecțiilor de date: liste, tupluri, seturi, dicționare.

• Metodele aplicabile tupluri, seturi, dicționare. Cerinţe:

• Revizuiți materialul didactic de la curs

• Executați sarcinile de la punctul I și II varianta conform registrului.

• La sarcina II salvați datele într-un fișier de tip json.

**I.**

Изучите предметную область варианта и продумайте, какую информацию следует хранить о ключевом объекте (например, для телефонного тарифа это наименование, стоимость, тип и т.д.).

Ориентируясь на Листинг 1, создайте небольшое приложение, которое хранит информацию об объекте выбранной предметной области и выводит ее на экран.

**Требования к данным:**

- кол-во объектов (элементов словаря) должно быть не менее 3;

- кол-во атрибутов объекта должно быть не менее 5;

- по крайней мере 3 атрибута объекта должны иметь различный тип: int, float, str и др.;

- по крайней мере 1 атрибут объекта должен иметь тип list.

|  |  |
| --- | --- |
| 6. | Мобильный оператор |
|  | Мобильный помощник оператора связи хранит информацию о тарифах: тип, стоимость звонка и др. |

**Вариант 6**

1. Описать структуру с именем WORKER, содержащую следующие поля: NАМЕ

- фамилия и инициалы работника; РОS

- название занимаемой должности; YЕАR - год поступления на работу.

2. Написать программу, выполняющую следующие действия:

ввод с клавиатуры данных в массив ТАВL, состоящий из десяти структур типа WORKER;

записи должны быть размещ. по алфавиту. вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;

если таких работников нет, вывести на дисплей соответ-щее сообщение.

|  |  |
| --- | --- |
| Codul: | Output: |
| tarife = [  {  "tip": "local",  "cost\_minut": 0.01,  "internet": 0,  },  {  "tip": "național",  "cost\_minut": 1.5,  "internet": 0.1,  },  {  "tip": "internațional",  "cost\_minut": 5.5,  "internet": 0.5,  },  {  "tip": "pro75",  "cost\_minut": 0.5,  "internet": 0,  }, ]  # Afișarea tarifelor def afiseaza\_tarife():  print("Tarife disponibile:")  for tarif in tarife:  tip = tarif.get("tip", "necunoscut")  cost\_minut = tarif.get("cost\_minut", )  internet = tarif.get("internet", )  print(f"Tip: {tip}, Cost/minut: {cost\_minut} lei, Internet: {internet} lei/MB\n\n")  # Căutare tarif după tip def cauta\_tarif(tip):  for tarif in tarife:  if tarif["tip"] == tip:  return tarif   def calculeaza\_cost(tip, durata):  tarif = cauta\_tarif(tip)  if tarif and "cost\_minut" in tarif:  cost\_total = tarif["cost\_minut"] \* durata  print(f"Costul unui apel de {durata} minute pentru tipul '{tip}' este: {cost\_total} lei")  else:  print("Tip de apel necunoscut sau tarif invalid pentru acest tip.")  def calculeaza\_cost\_internet(tip, durata):  tarif = cauta\_tarif(tip)  if tarif and "internet" in tarif:  cost\_total = tarif["internet"] \* durata  print(f"Costul {durata} MB pentru tipul '{tip}' este: {cost\_total} lei")  else:  print("Tip necunoscut sau tarif invalid.")  # Meniul principal def meniu():  while True:  print('1. Afișează tarife\n2. Calculează costul local\n3. Calculează costul național\n4. Calculează costul internațional\n5. Calculează costul pro75\n6. Calculează costul internetului\n0. Ieșire')  sel = input('Alege opțiunea: ')   if sel == '1':  afiseaza\_tarife()   elif sel == '2':  n = float(input('Introduce durata în minute: '))  calculeaza\_cost("local", n)  elif sel == '3':  n = float(input('Introduce durata în minute: '))  calculeaza\_cost("național", n)  elif sel == '4':  n = float(input('Introduce durata în minute: '))  calculeaza\_cost("internațional", n)  elif sel == '5':  n = float(input('Introduce durata în minute: '))  calculeaza\_cost('pro75',n)  elif sel == '6':  n = float(input('Introduce cate mb aveti nevoie?: '))  calculeaza\_cost\_internet('pro75', n)  calculeaza\_cost\_internet('internațional',n)  elif sel == '0':  print("Ieșire...")  break   else:  print("Opțiune invalidă. Alegeți o opțiune între 0 și 4.")   meniu() | 1. Afișează tarife  2. Calculează costul local  3. Calculează costul național  4. Calculează costul internațional  5. Calculează costul pro75  6. Calculează costul internetului  0. Ieșire  Alege opțiunea: 1  Tarife disponibile:  Tip: local, Cost/minut: 0.01 lei, Internet: 0 lei/MB  Tip: național, Cost/minut: 1.5 lei, Internet: 0.1 lei/MB  Tip: internațional, Cost/minut: 5.5 lei, Internet: 0.5 lei/MB  Tip: pro75, Cost/minut: 0.5 lei, Internet: 0 lei/MB  1. Afișează tarife  2. Calculează costul local  3. Calculează costul național  4. Calculează costul internațional  5. Calculează costul pro75  6. Calculează costul internetului  0. Ieșire  Alege opțiunea: 2  Introduce durata în minute: 4  Costul unui apel de 4.0 minute pentru tipul 'local' este: 0.04 lei  1. Afișează tarife  2. Calculează costul local  3. Calculează costul național  4. Calculează costul internațional  5. Calculează costul pro75  6. Calculează costul internetului  0. Ieșire  Alege opțiunea: 3  Introduce durata în minute: 6  Costul unui apel de 6.0 minute pentru tipul 'național' este: 9.0 lei  1. Afișează tarife  2. Calculează costul local  3. Calculează costul național  4. Calculează costul internațional  5. Calculează costul pro75  6. Calculează costul internetului  0. Ieșire  Alege opțiunea: 4  Introduce durata în minute: 3  Costul unui apel de 3.0 minute pentru tipul 'internațional' este: 16.5 lei  1. Afișează tarife  2. Calculează costul local  3. Calculează costul național  4. Calculează costul internațional  5. Calculează costul pro75  6. Calculează costul internetului  0. Ieșire  Alege opțiunea: 4  Introduce durata în minute: 7  Costul unui apel de 7.0 minute pentru tipul 'internațional' este: 38.5 lei  1. Afișează tarife  2. Calculează costul local  3. Calculează costul național  4. Calculează costul internațional  5. Calculează costul pro75  6. Calculează costul internetului  0. Ieșire  Alege opțiunea: 5  Introduce durata în minute: 34  Costul unui apel de 34.0 minute pentru tipul 'pro75' este: 17.0 lei  1. Afișează tarife  2. Calculează costul local  3. Calculează costul național  4. Calculează costul internațional  5. Calculează costul pro75  6. Calculează costul internetului  0. Ieșire  Alege opțiunea: 6  Introduce cate mb aveti nevoie?: 800  Costul 800.0 MB pentru tipul 'pro75' este: 0.0 lei  Costul 800.0 MB pentru tipul 'internațional' este: 400.0 lei |

|  |  |
| --- | --- |
| Codul: | Output: |
| import json from operator import itemgetter  def creeaza\_angajat(nume, functie, anul\_angajarii):  return {  "NAME": nume,  "POS": functie,  "YEAR": anul\_angajarii  }  def citeste\_angajati(n):  angajati = []  for i in range(n):  print(f"Angajatul {i + 1}:")  nume = input("Nume și Prenume: ")  functie = input("Funcția: ")  anul\_angajarii = int(input("Anul angajării: "))  angajati.append(creeaza\_angajat(nume, functie, anul\_angajarii))  return angajati  def afiseaza\_angajati\_vechime(angajati, stagiu, anul\_curent=2024):  angajati\_selectati = [ang for ang in angajati if (anul\_curent - ang["YEAR"]) > stagiu]   if angajati\_selectati:  print(f"\nAngajați cu stagiu mai mare de {stagiu} ani:")  for angajat in angajati\_selectati:  print(angajat["NAME"])  else:  print(f"Nu există angajați cu stagiu mai mare de {stagiu} ani.")  def salveaza\_json(angajati, nume\_fisier="angajati.json"):  with open(nume\_fisier, "w") as f:  json.dump(angajati, f, ) #indet = 4  print(f"\nDatele au fost salvate în fișierul '{nume\_fisier}'.")    def meniu():  angajati = []   while True:  print("\n--- Meniu ---")  print("1. Introduceți datele despre angajați")  print("2. Afișați angajații cu stagiu mai mare decât specificat")  print("3. Salvați datele în fișier JSON")  print("4. Ieșire")   optiune = input("Alegeți o opțiune: ")   if optiune == '1':  n=int(input('Introduceti NR de angajati'))  angajati = citeste\_angajati(n)  angajati.sort(key=itemgetter("NAME"))  print("Datele au fost introduse și sortate.")   elif optiune == '2':  if angajati:  stagiu = int(input("Introduceți stagiu în ani: "))  afiseaza\_angajati\_vechime(angajati, stagiu)  else:  print("Nu există date despre angajați. Introduceți mai întâi datele.")   elif optiune == '3':  if angajati:  salveaza\_json(angajati)  else:  print("Nu există date despre angajați de salvat.")   elif optiune == '4':  print("Ieșire din program.")  break   else:  print("Opțiune invalidă. Vă rugăm să alegeți o opțiune validă.")   meniu() | --- Meniu ---  1. Introduceți datele despre angajați  2. Afișați angajații cu stagiu mai mare decât specificat  3. Salvați datele în fișier JSON  4. Ieșire  Alegeți o opțiune: 1  Introduceti NR de angajati3  Angajatul 1:  Nume și Prenume: Chistol Maxim  Funcția: Director  Anul angajării: 1995  Angajatul 2:  Nume și Prenume: Vasile Popescu  Funcția: Programator  Anul angajării: 2000  Angajatul 3:  Nume și Prenume: Ciobanu Vasile  Funcția: Secretar  Anul angajării: 1995  Datele au fost introduse și sortate.  --- Meniu ---  1. Introduceți datele despre angajați  2. Afișați angajații cu stagiu mai mare decât specificat  3. Salvați datele în fișier JSON  4. Ieșire  Alegeți o opțiune: 2  Introduceți stagiu în ani: 5  Angajați cu stagiu mai mare de 5 ani:  Chistol Maxim  Ciobanu Vasile  Vasile Popescu  --- Meniu ---  1. Introduceți datele despre angajați  2. Afișați angajații cu stagiu mai mare decât specificat  3. Salvați datele în fișier JSON  4. Ieșire  Alegeți o opțiune: 3  Datele au fost salvate în fișierul 'angajati.json'. |

# Concluzii:

# Colecțiile de date precum listele, tuplurile, seturile și dicționarele sunt structuri fundamentale în gestionarea și organizarea datelor în Python. Acestea oferă flexibilitate și eficiență în stocarea și manipularea informațiilor diverse. Aplicarea metodelor specifice fiecărui tip de colecție facilitează accesul și modificarea datelor, în funcție de cerințele aplicației. Salvatul datelor în format JSON permite o păstrare structurată și portabilă a informațiilor, simplificând accesul și reutilizarea ulterioară. Utilizarea atributelor de diferite tipuri asigură o reprezentare complexă și realistă a obiectelor dintr-o anumită domeniu.