**Звіт про виконання лабораторної роботи №1 з курсу Технології програмування на мові Python**

Сиров Віталій КБ-241

Звіт до лабораторної роботи №1

Списки. Словники. Кортежі

**Мета роботи**: Використовуючи теоретичне підґрунтя про складні структури даних Списки, Словники Кортежі, та використовуючи існуючий код, доробити програму додавши функціонал, що буде вказано в завданні до лабораторної роботи.

Під час виконання лабораторної роботи №1 було виконано наступні завдання:

**Необхідно розширити відомості про студента до 4х полів. На даний час використовується лише два поля (name та phone).**

Хід виконання завдання:

Щоб розширити відомості про студента, треба додати інформації у список

Текст програми:

|  |
| --- |
| list = [      {"name":"Bob", "phone":"0631234567", "age":"17", "group":"CB-242"},      {"name":"Emma", "phone":"0631234567", "age":"19", "group":"CB-241"},      {"name":"Jon",  "phone":"0631234567", "age":"18", "group":"CB-241"},      {"name":"Zak",  "phone":"0631234567", "age":"21", "group":"CB-242"}  ] |

**Необхідно переробити існуючий функціонал враховуючи розширення відомості про студента до 4х полів.**

Хід виконання завдання:

Для переробки існуючого функціоналу треба додати 2 нові змінні про введення даних та їх додавання у список

Текст програми:

|  |
| --- |
| def addNewElement():      name = input("Pease enter student name: ")      phone = input("Please enter student phone: ")      age = input("Please enter age of student: ")      group = input("Please enter the group of student: ")      newItem = {"name": name, "phone": phone, "age":age, "group":group} |

**Необхідно реалізувати з нуля функціонал зміни інформації про студента враховуючи той факт, що вже існує реалізація додавання нового запису та видалення існуючого. При зміні інформації про студента список має залишатись відсортованим.**

Хід виконання завдання:

Для реалізації функціоналу зміни даних, створюємо цикл для пошуку відповідного значення у списку, видаляємо його після позначення для подальшої заміни за вибором користувача даних на нові. Після внесених змін вноситься новий запис у список із сортуванням

Текст програми:

|  |
| --- |
| def updateElement():      name = input("Please enter name to be updated: ")      found = False      for item in list:          if item["name"]==name:              upd = input(f"What do you wanna change? N - name, P - phone, A - age, G - group, Q - quit: ")              match upd:                  case "N" | "n":                      newname = input(f"Enter new name for student: ")                      item["name"] = newname                  case "P" | "p":                      newphone = input(f"Enter new phone for student: ")                      item["phone"] = newphone                  case "A" | "a":                      newage = input(f"Enter new age for student: ")                      item["age"] = newage                  case "G" | "g":                      newgroup = input(f"Enter new group for student: ")                      item["group"] = newgroup                  case "Q" | "q":                      print("Update canceled")                      return                  case \_:                      print("Wrong choice")              print(f"Student was update")              found = True              break      if not found:          print(f"Element wasn`t found")          return      for i in range(len(list)):          for j in range(i + 1, len(list)):              if list[i]["name"] > list[j]["name"]:                  list[i], list[j] = list[j], list[i] |

Повний код програми:

|  |
| --- |
| ## List [Item1, Item2, Item3]  ## Item {"name":"Jon", "phone":"0631234567"}  # already sorted list  list = [      {"name":"Bob", "phone":"0631234567", "age":"17", "group":"CB-242"},      {"name":"Emma", "phone":"0631234567", "age":"19", "group":"CB-241"},      {"name":"Jon",  "phone":"0631234567", "age":"18", "group":"CB-241"},      {"name":"Zak",  "phone":"0631234567", "age":"21", "group":"CB-242"}  ]  def printAllList():      for elem in list:          strForPrint = "Student name is " + elem["name"] + ",  Phone is " + elem["phone"] + ", Age is " +elem["age"] + ", Group is " + elem["group"]          print(strForPrint)      return  def addNewElement():      name = input("Pease enter student name: ")      phone = input("Please enter student phone: ")      age = input("Please enter age of student: ")      group = input("Please enter the group of student: ")      newItem = {"name": name, "phone": phone, "age":age, "group":group}      # find insert position      insertPosition = 0      for item in list:          if name > item["name"]:              insertPosition += 1          else:              break      list.insert(insertPosition, newItem)      print("New element has been added")      return  def deleteElement():      name = input("Please enter name to be delated: ")      deletePosition = -1      for item in list:          if name == item["name"]:              deletePosition = list.index(item)              break      if deletePosition == -1:          print("Element was not found")      else:          print("Dele position " + str(deletePosition))          # list.pop(deletePosition)          del list[deletePosition]      return  def updateElement():      name = input("Please enter name to be updated: ")      deletePosition = -1      for index, newitem in enumerate(list):          if newitem["name"] == name:              deletePosition = index              item = newitem              break      if deletePosition == -1:          print("Element was not found")          return      list.pop(deletePosition)      upd = input(f"What do you wanna change? N - name, P - phone, A - age, G - group, Q - quit: ")      match upd:          case "N" | "n":                      newname = input(f"Enter new name for student: ")                      item["name"] = newname          case "P" | "p":                      newphone = input(f"Enter new phone for student: ")                      item["phone"] = newphone          case "A" | "a":                      newage = input(f"Enter new age for student: ")                      item["age"] = newage          case "G" | "g":                      newgroup = input(f"Enter new group for student: ")                      item["group"] = newgroup          case "Q" | "q":                      print("Update canceled")                      return          case \_:                      print("Wrong choice")      insertPosition = 0      for s in list:          if item["name"] > s["name"]:              insertPosition += 1          else:              break        list.insert(insertPosition, item)      print("Student was update")      # implementation required  def main():      while True:          chouse = input("Please specify the action [ C create, U update, D delete, P print,  X exit ] ")          match chouse:              case "C" | "c":                  print("New element will be created:")                  addNewElement()                  printAllList()              case "U" | "u":                  print("Existing element will be updated")                  updateElement()              case "D" | "d":                  print("Element will be deleted")                  deleteElement()              case "P" | "p":                  print("List will be printed")                  printAllList()              case "X" | "x":                  print("Exit()")                  break              case \_:                  print("Wrong chouse")  main() |

Посилання на github:

<https://github.com/Renssell/TP-CB-241-VitaliySyrov>

Знімок екрану з посилання на github:

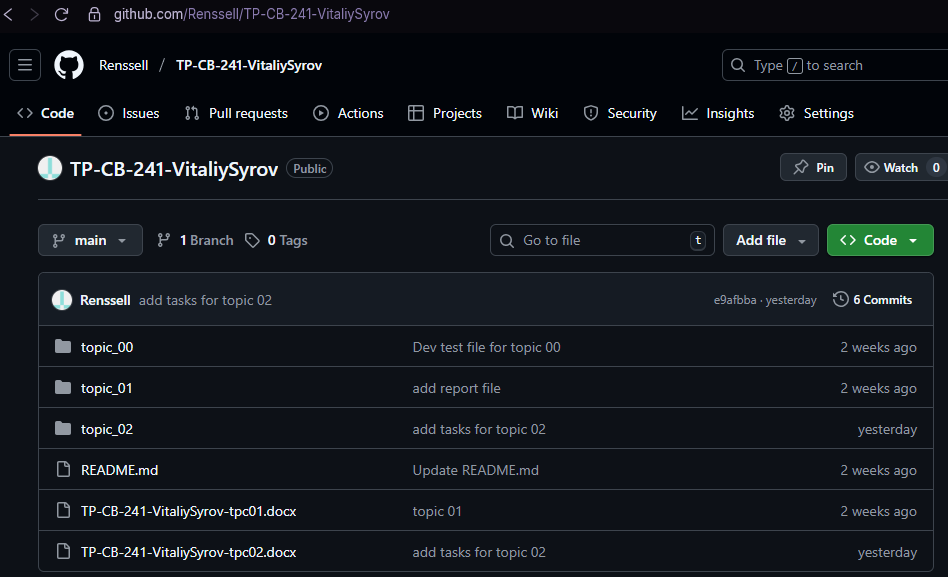


Рисунок 1 – GitHub репозиторій

Приклад роботи програми

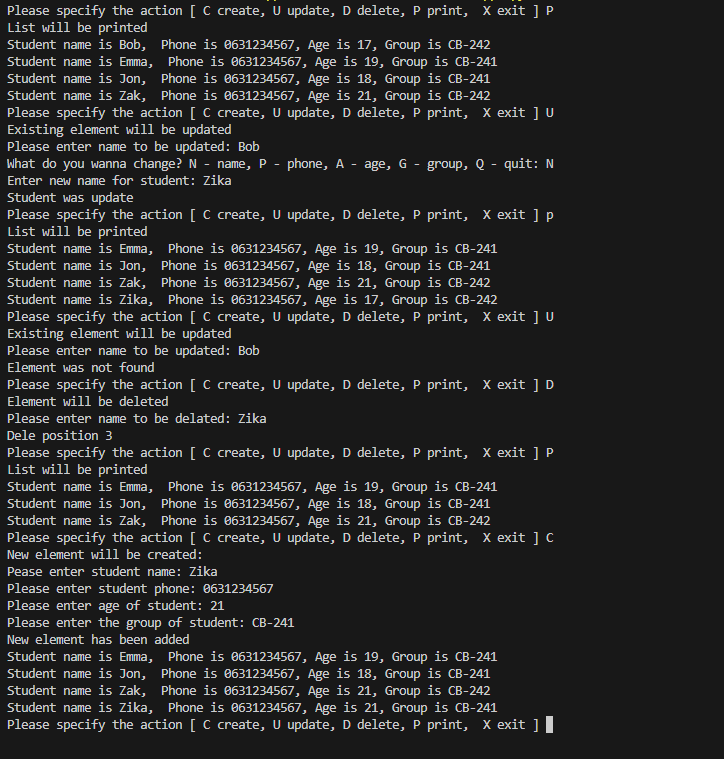


Рисунок 2 – виконання усіх функцій програми

Висновок: у ході виконання лабораторної роботи закріпив знання про складні структури даних у Python – списки, словники та кортежі. Було виконано модифікацію існуючого коду та реалізовано додатковий функціонал відповідно до завдання.