МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №8**

з дисципліни «Програмування мовою Python»

на тему «Робота зі словниками»

Виконав студент групи КН-35

Авдасьов В.Д.

Перевірила Парфененко Ю.В.

Варіант 1

Суми 2024

**Завдання 1**

1. **Постановка задачі**

Перший студент має створити словник, в якому має зберігатися інформація про успішність студентів групи, а саме:

* номер групи;
* прізвище, ім'я та по батькові;
* курс;
* предмети та оцінки за семестр.

Окрім обов'язкових елементів словника можна додавати власні.

Також перший студент має написати програмний код, у якому реалізувати одну із функцій по роботі зі словником - наприклад, додавання даних, а також поставити задачу другому студенту, наприклад, розробити функцію сортування даних у словнику.

Усі студенти повинні працювати зі словником, який розмістив перший студент. Кожен наступний студент у відповіді на завдання повинен охарактеризувати словник, створений попереднім студентом, чи є структура оптимальною, за необхідності змінити словник та пояснити необхідність змін. Доопрацювати програмний код, написаний попереднім студентом, змінивши словник (за необхідності) та додавши ще одну функцію по роботі зі словником. У програмному коді коментарі обов'язкові.

1. **Текст програми**

# Глобальний словник для збереження даних про студентів за групами

students\_group = {}

# Функція для додавання студента з унікальним ID

def add\_student\_with\_id(group\_number, student\_id, full\_name, course, subjects\_grades):

# Якщо група ще не існує, створюємо її

if group\_number not in students\_group:

students\_group[group\_number] = {}

# Створення запису про студента з унікальним ID

student\_info = {

'ПІБ': full\_name,

'Курс': course,

'Предмети та оцінки': subjects\_grades

}

# Додаємо студента за унікальним ідентифікатором (ID)

students\_group[group\_number][student\_id] = student\_info

print(f"Студента {full\_name} додано до групи {group\_number} з ID {student\_id}")

# Функція для редагування оцінок студента

def edit\_grades(group\_number, student\_id, new\_grades):

if group\_number in students\_group and student\_id in students\_group[group\_number]:

students\_group[group\_number][student\_id]['Предмети та оцінки'] = new\_grades

print(f"Оцінки студента {students\_group[group\_number][student\_id]['ПІБ']} оновлено")

else:

print(f"Студента з ID {student\_id} у групі {group\_number} не знайдено")

# Функція для виведення інформації про студентів

def display\_students():

for group, students in students\_group.items():

print(f"Група: {group}")

for student\_id, student in students.items():

print(f" ID: {student\_id}, ПІБ: {student['ПІБ']}, Курс: {student['Курс']}")

for subject, grade in student['Предмети та оцінки'].items():

print(f" {subject}: {grade}")

# Функція для обчислення середнього бала студента

def calculate\_average\_grade(subjects\_grades):

total\_grades = sum(subjects\_grades.values())

number\_of\_subjects = len(subjects\_grades)

return total\_grades / number\_of\_subjects if number\_of\_subjects > 0 else 0

# Функція для сортування студентів за середнім балом

def sort\_students\_by\_average\_grade(group\_number):

if group\_number in students\_group:

# Отримуємо список студентів з групи

students\_list = list(students\_group[group\_number].items())

# Сортуємо за середнім балом (від вищого до нижчого)

sorted\_students = sorted(students\_list, key=lambda student: calculate\_average\_grade(student[1]['Предмети та оцінки']), reverse=True)

# Оновлюємо відсортовану інформацію в групі

students\_group[group\_number] = dict(sorted\_students)

print(f"Студентів у групі {group\_number} відсортовано за середнім балом.")

else:

print(f"Групу {group\_number} не знайдено.")

# Додавання студентів з використанням унікального ID

add\_student\_with\_id('401', 1, 'Іванов Юрій Сергійович', 2, {'Математика': 85, 'Фізика': 90, 'Програмування': 95})

add\_student\_with\_id('401', 2, 'Порощенко Петро Петрович', 2, {'Математика': 75, 'Фізика': 80, 'Програмування': 88})

add\_student\_with\_id('401', 3, 'Яценко Іван Олександрович', 2, {'Математика': 78, 'Фізика': 83, 'Програмування': 85})

# Оновлення оцінок для студента з ID 1

edit\_grades('401', 1, {'Математика': 90, 'Фізика': 92, 'Програмування': 97})

# Сортування студентів у групі 401 за середнім балом

sort\_students\_by\_average\_grade('401')

# Виведення інформації про студентів

display\_students()

#Я додав у код функцію сортування студентів за середнім балом (від більшого до меншого)

1. **Приклад роботи програми**

**Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис**

Github: <https://github.com/SolDima/python-lab5>

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис