Звіт

З лабораторної роботи №5

Студента групи МІТ-31

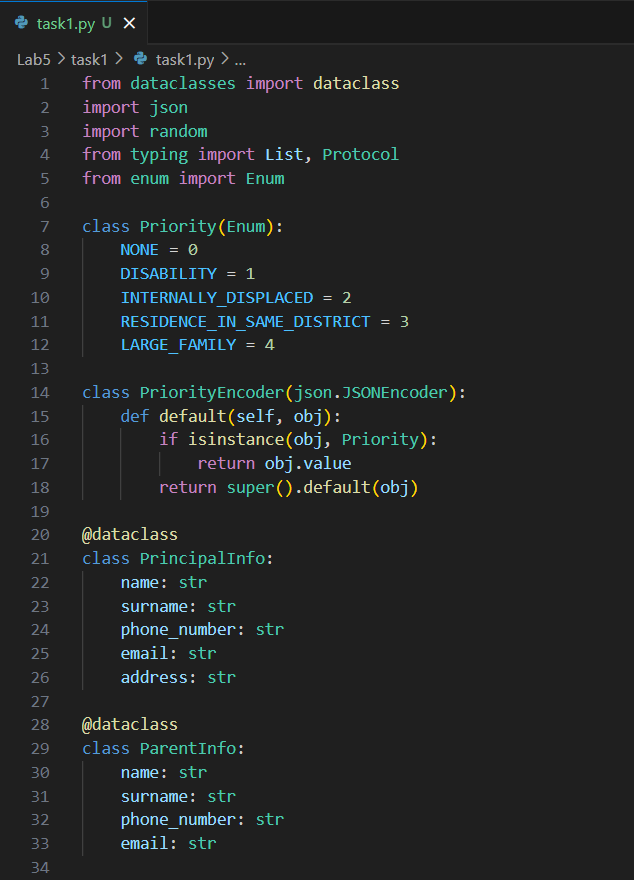
Добровольського Арсенія Михайловича

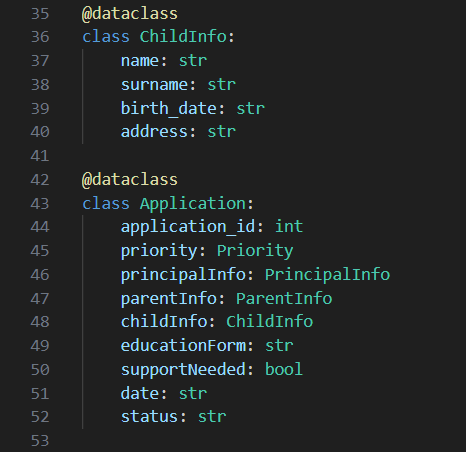
Варіант №9

**Тема роботи:** Інтерфейси та їх реалізація у Python

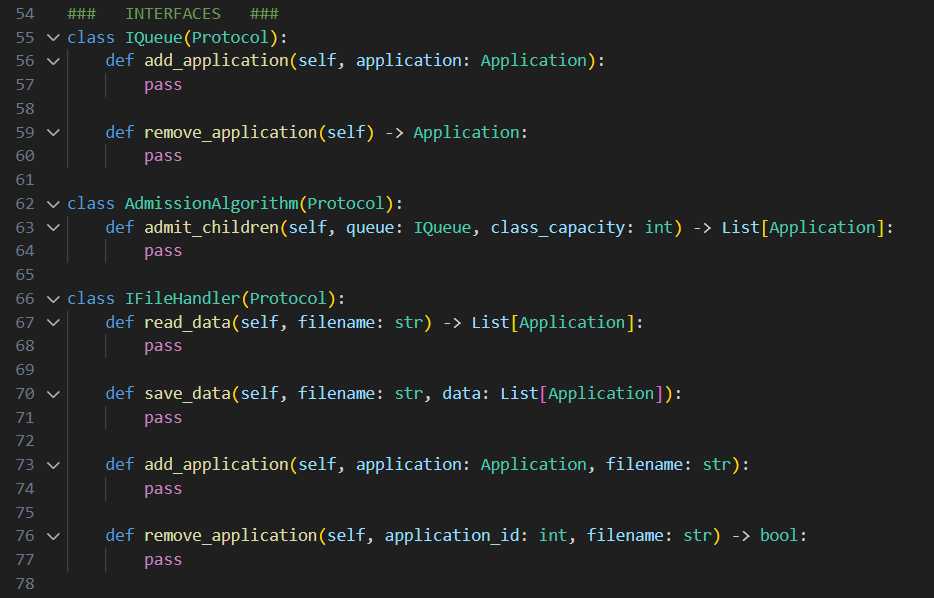
**Мета роботи:** розглянути поняття інтерфейсу та його призначення. Розглянути можливості Python зі створення інтерфейсів.

Спершу реалізовано клас перелічувального типу Enum, що містить пріоритети заявок, а також 4 датакласи, що представляють інформацію про директора, батьків, дитини та саму заявку відповідно:

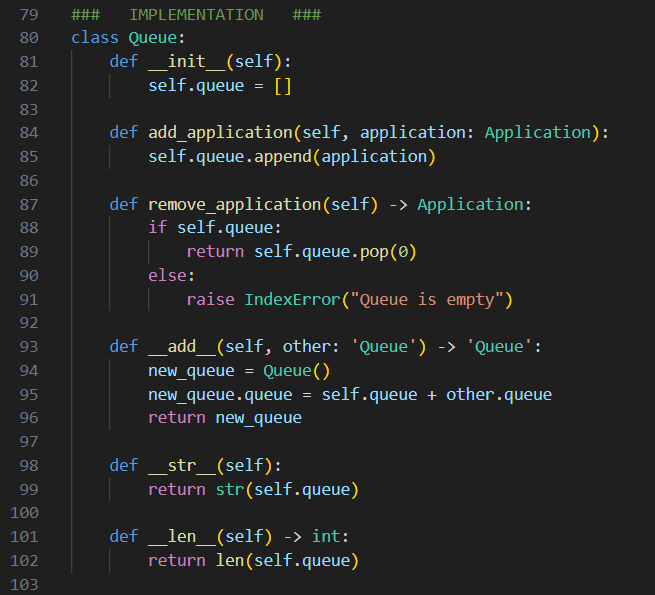


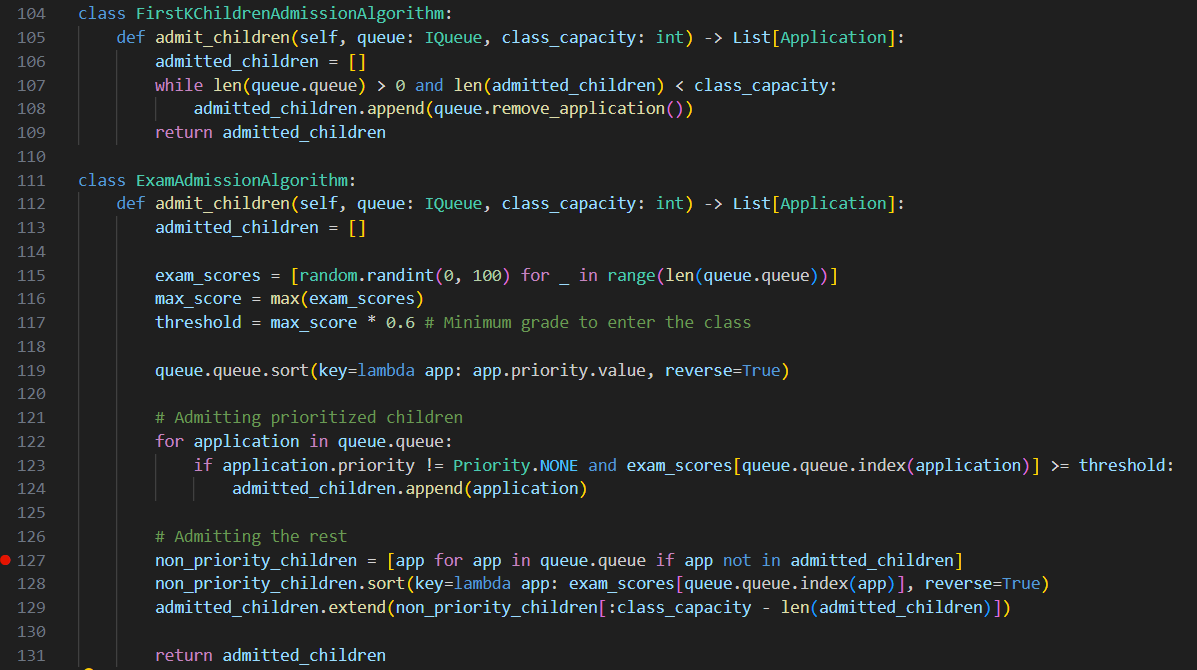


Інтерфейси IQueue, AdmissionAlgorithm та IFileHandler визначають методи, які мають бути реалізовані в класах черги, алгоритму прийому та обробника файлів відповідно:



Імплементація інтерфейсів:





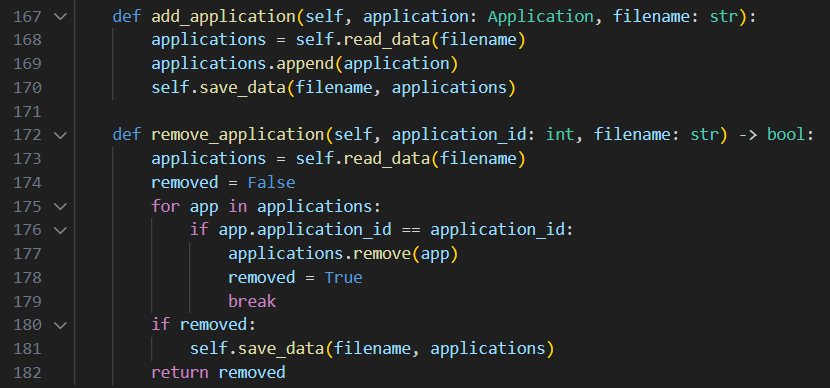
Алгоритм зарахування перших K дітей (FirstKChildrenAdmissionAlgorithm) призначає першим K дітям у черзі місця у класі. Допоки черга не порожня і кількість прийнятих дітей не досягла максимальної місткості класу, цей алгоритм продовжує зараховувати дітей до класу.

Водночас алгоритм прийому за результатами вступного іспиту (ExamAdmissionAlgorithm) передбачає присвоєння кожній дитині випадкового балу за іспит від 0 до 100. Далі обчислюється максимальний бал серед усіх «складених» іспитів для визначення порогового балу, який повинна досягти або перевищити дитина, щоб бути зарахованою до навчального закладу. Далі відбувається сортування черги за пріоритетом, щоб спочатку розглянути пріоритетні заявки. Після сортування алгоритм перевіряє кожну заявку в черзі. Якщо вона має пріоритет (тобто не дорівнює Priority.NONE) і має оцінку, яка перевищує пороговий бал, вона приймається і відповідно така дитина зараховується до школи. Після обробки пріоритетних заявок алгоритм приймає непріоритетних заявників, вибираючи їх за балом, починаючи з найвищого.

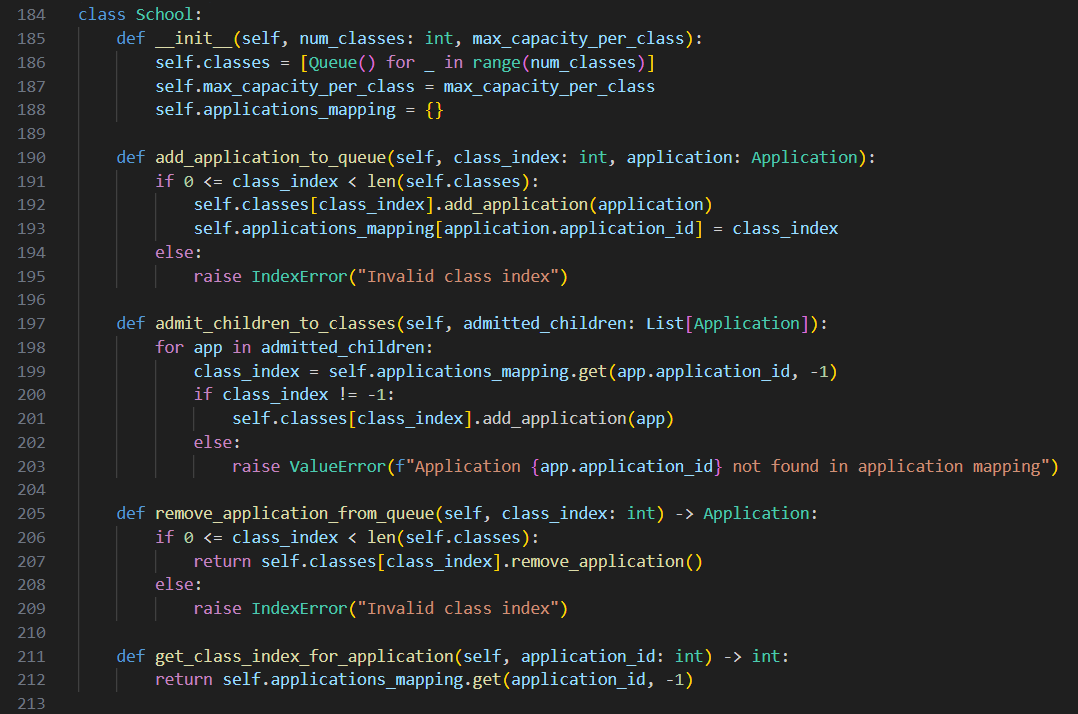
Внаслідок роботи обох алгоритмів повертається список зарахованих дітей.

Імплементація інтерфейсу IFileHandler, що відповідає за зчитування даних з файлу, зберігання даних у файл, додавання заявки в кінець файлу та видалення заявки за її ідентифікатором:

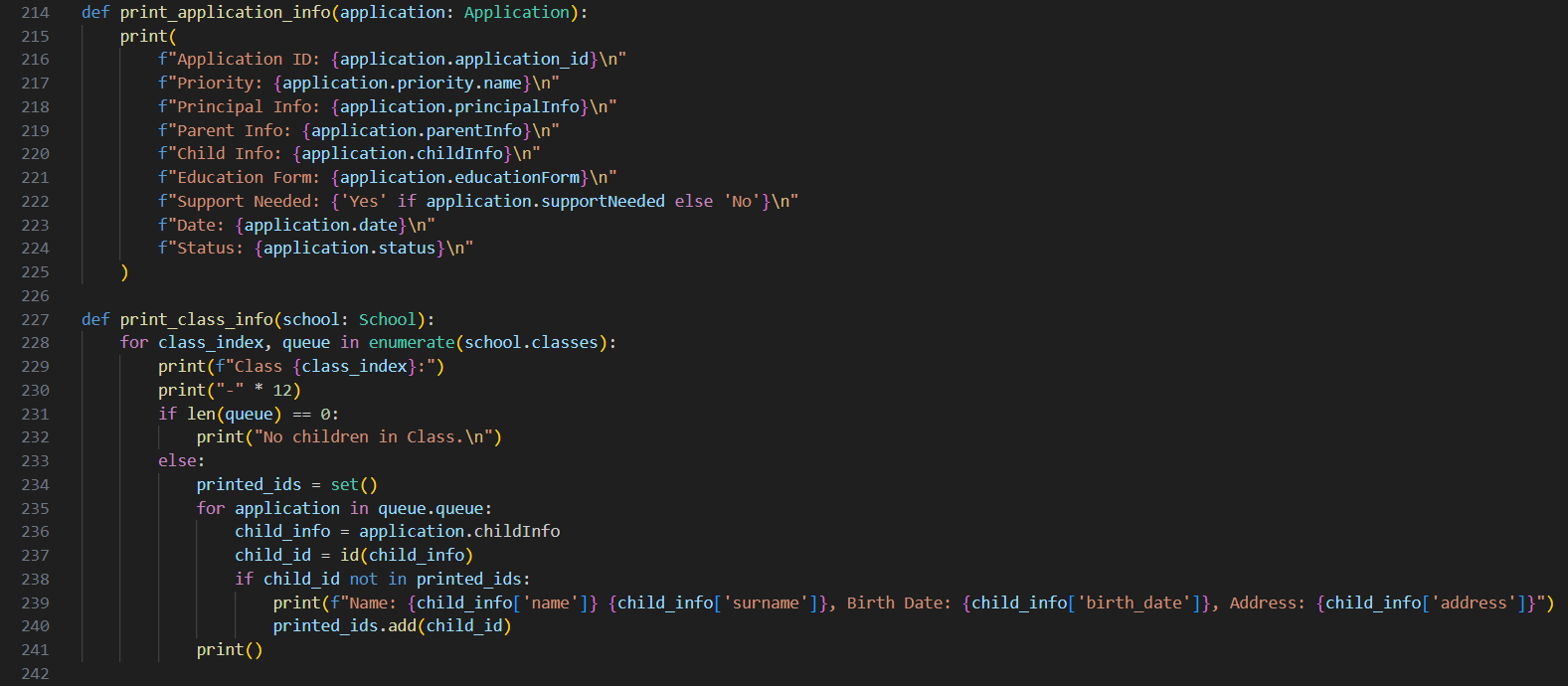




Також визначено клас Школа, який представляє навчальний заклад, куди зараховуються діти. Він надає методи додавання заявки до черги класу, прийому дітей до класів, видалення заявки з черги класу та отримання індексу класу для заявки, який приймає ідентифікатор заявки та повертає індекс класу, до якого вона відноситься, з використанням словника applications\_mapping:

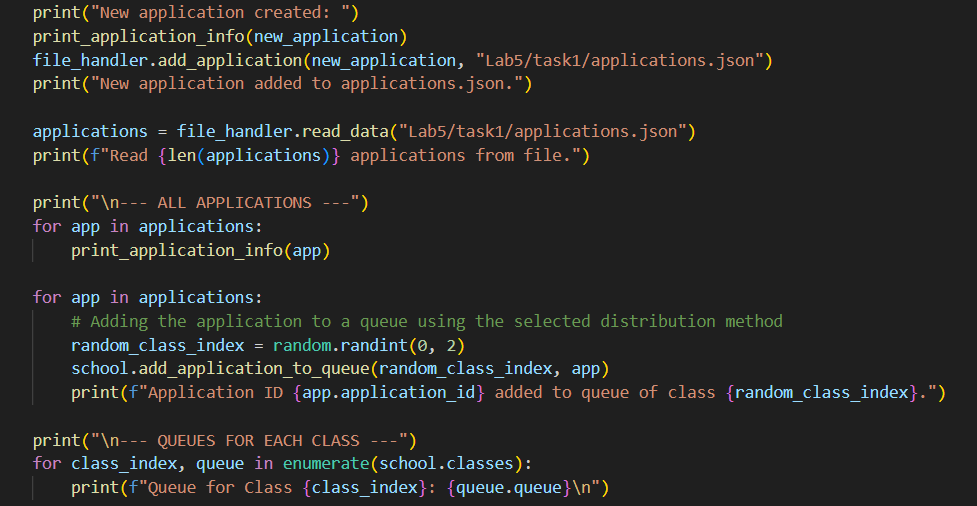


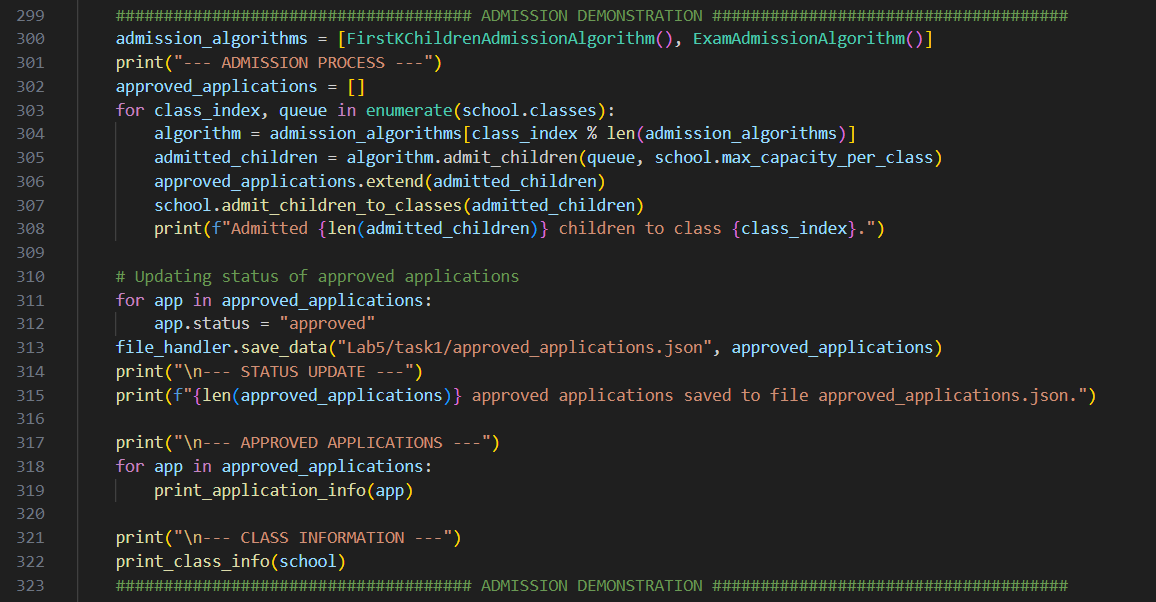
Метод print\_application\_info виводить інформацію про конкретну заявку, а метод print\_class\_info виводить інформацію про кожен клас та дітей, які в нього зараховані:

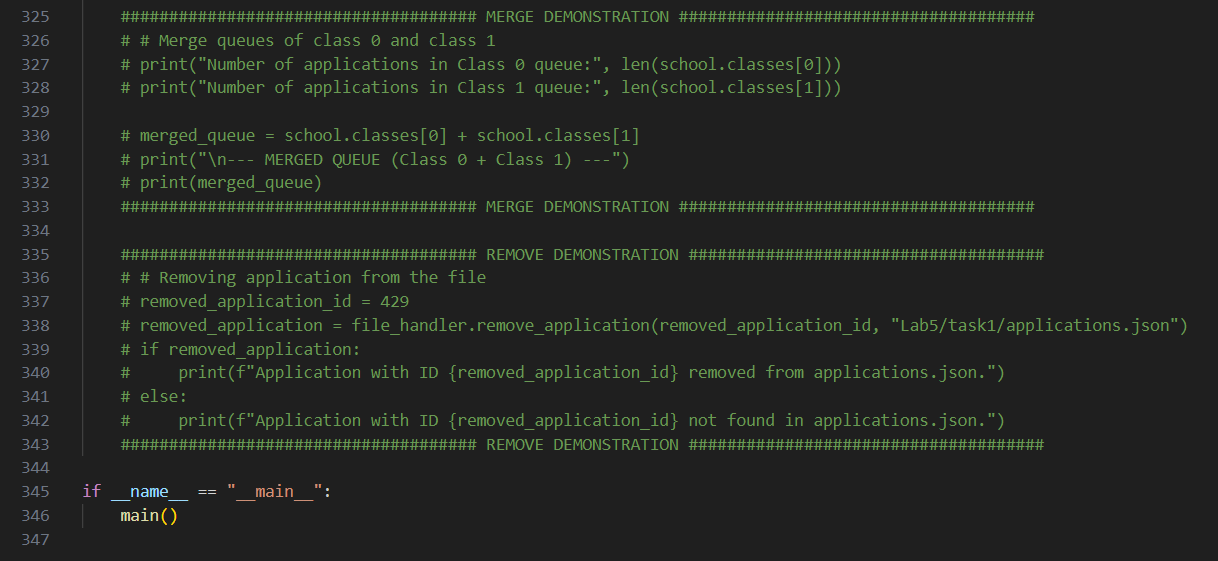


Код основної програми має вигляд:



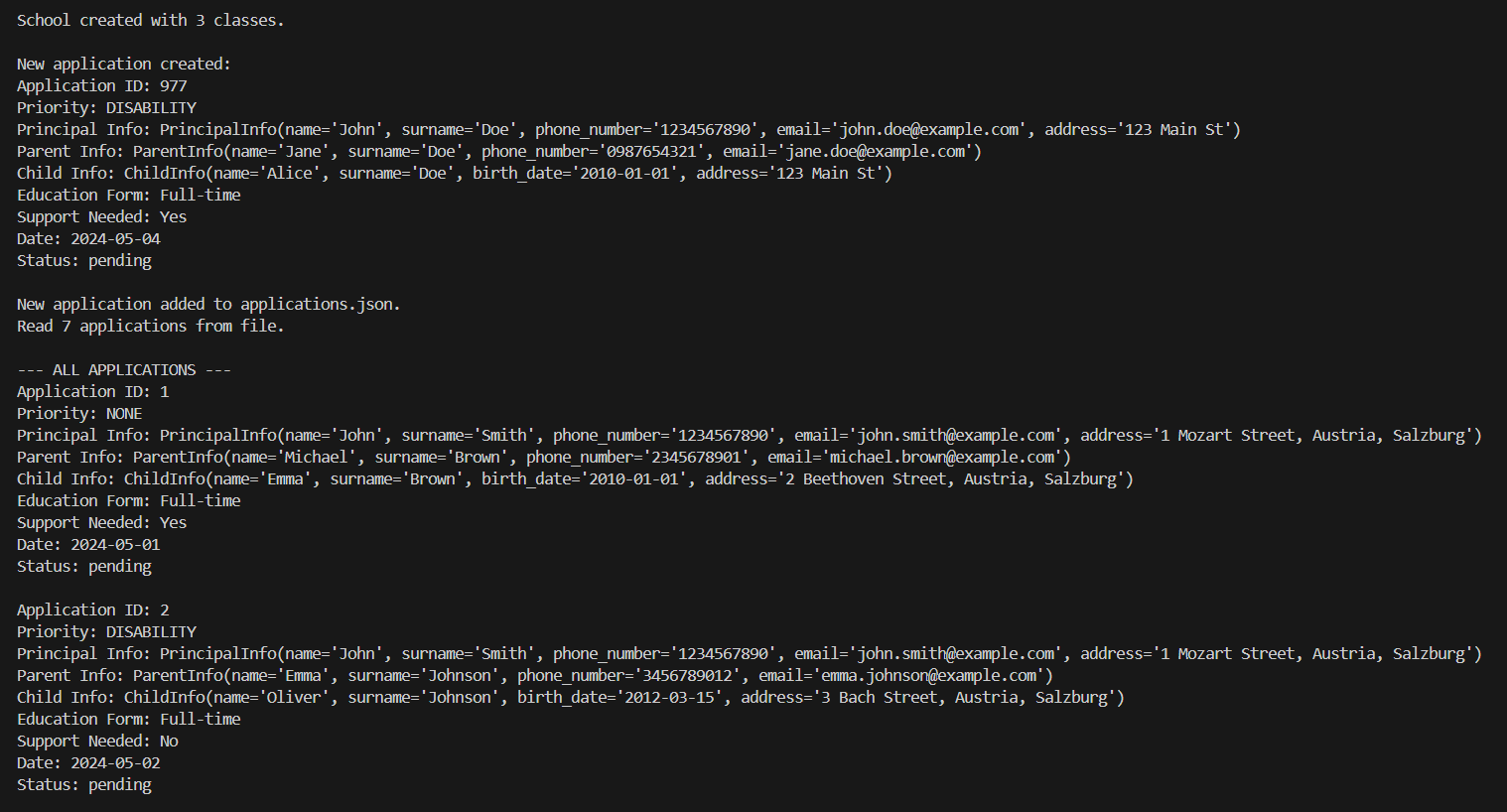


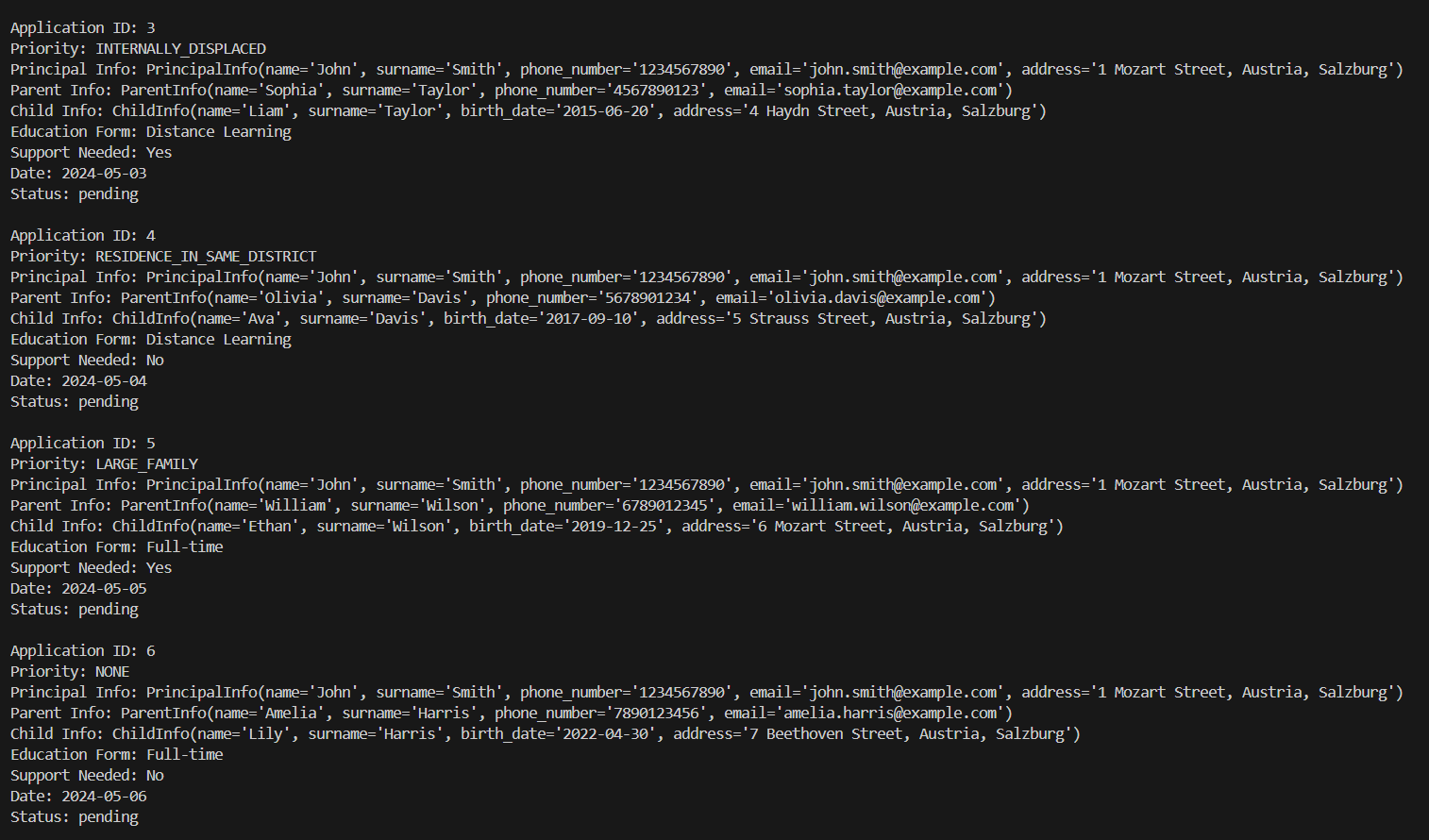


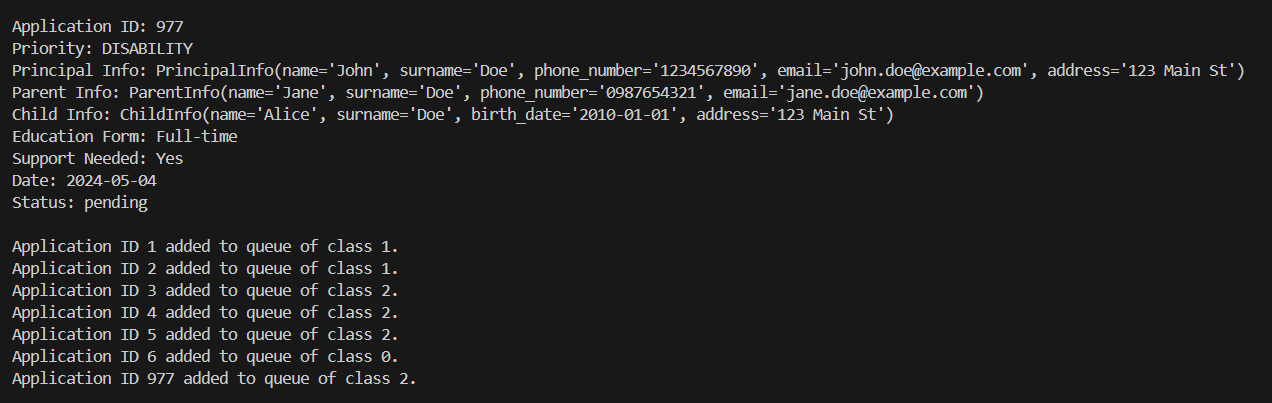


Демонстрація роботи програми:

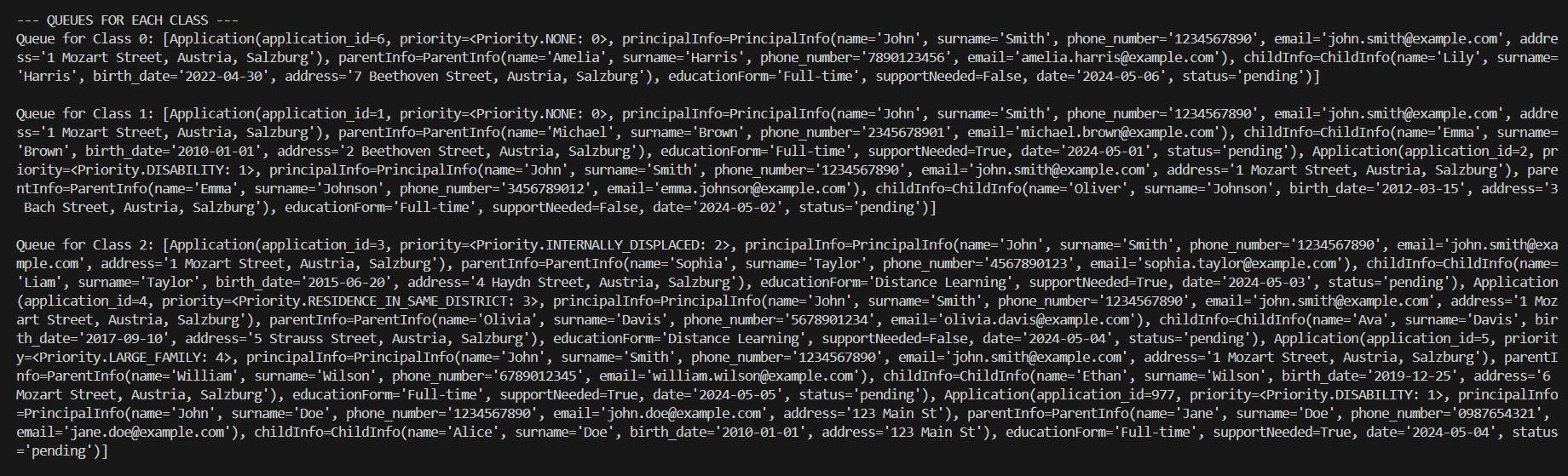
1. Створення нової заявки, додавання її у json файл і зчитування всіх заявок з файлу:



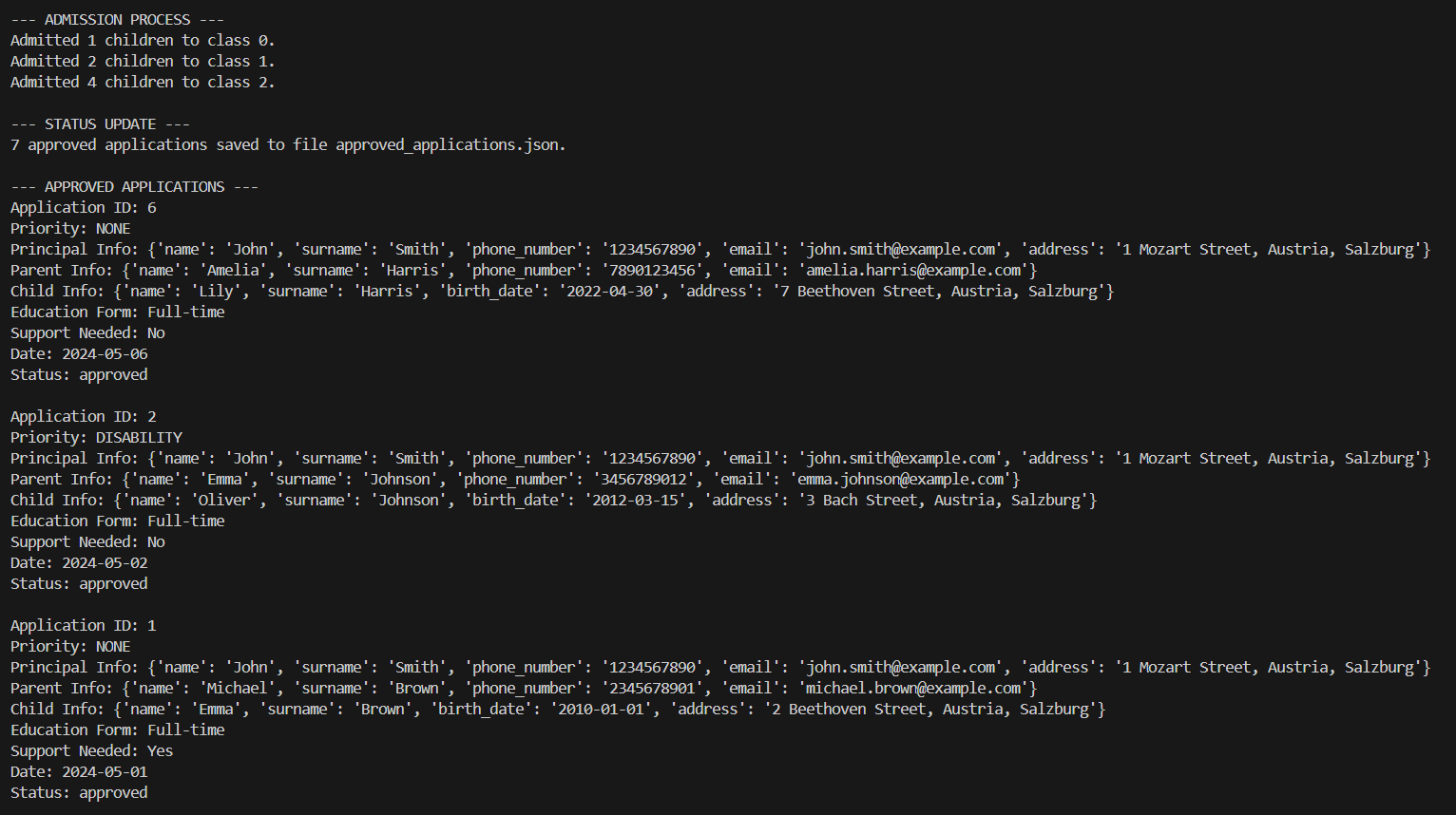


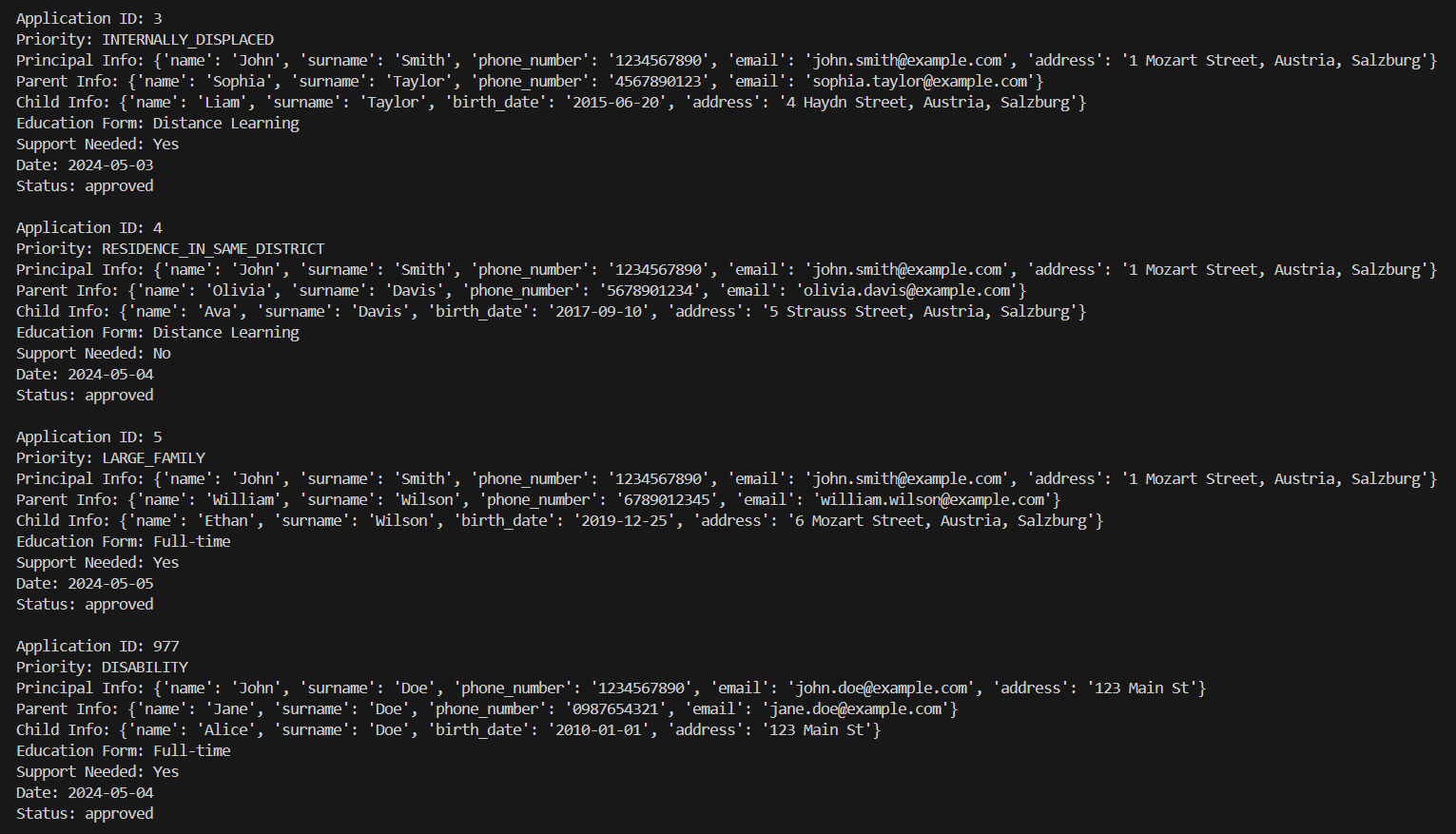


1. Виведення черг по кожному класу:

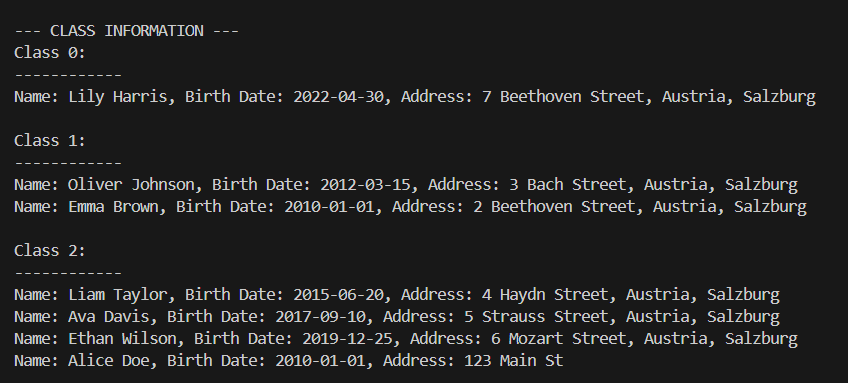


1. Процес зарахування дітей до класів, оновлення статусів заяв, збереження їх в інший json файл та виведення усіх схвалених заяв:

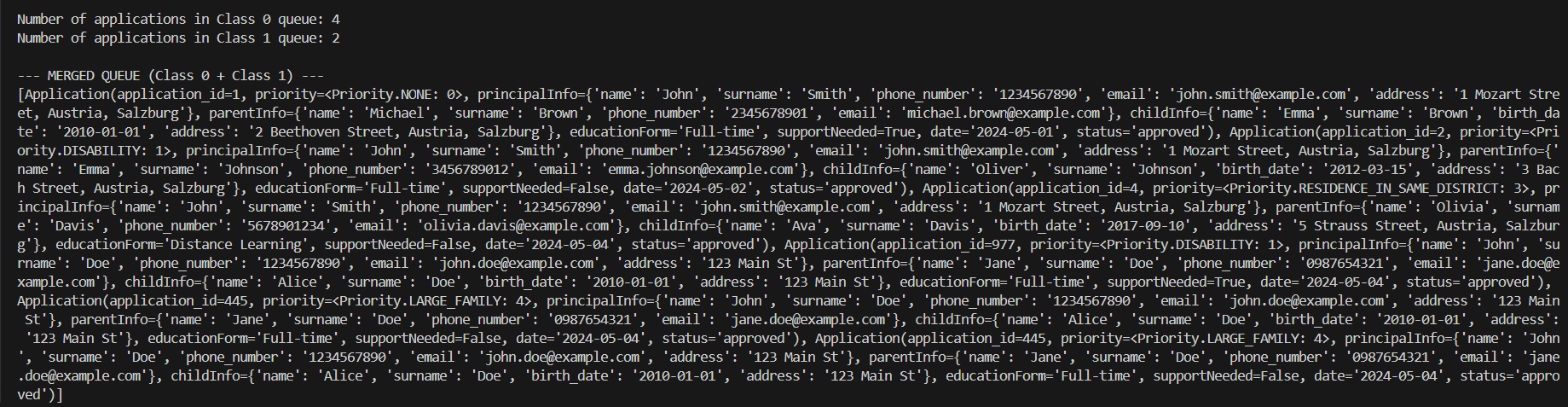




1. Виведення списку дітей по кожному класу:



1. Демонстрація злиття черг двох класів:



**Висновок:** в ході виконання лабораторної роботи я ознайомився з поняттям інтерфейсів в об’єктно-орієнтованому програмуванні, дізнався особливості їх реалізації у Python, а також написав програму, що дозволяє ефективно вести процес подачі заяв і зарахування дітей до перших класів школи.