**Звіт**

**28 - Університет**

Стретович Тимур Артемович, студент 2 курсу, група КомпМат-1

# Анотація.

В даному проекті реалізовано систему обліку в університеті (це може бути будь-який навчальний заклад, в силу наслідувальної природи можна організувати роботу проекту і для інших закладів). Проект складається з одного заголовочного файлу func28.h, одного файлу з реалізацією методів func28.cpp та одного тестового файлу test1.cpp. В процесі роботи створюється також два файли: stud\_info\_28.txt та lect\_info\_28.txt.

# 1 Опис.

Проект містить вісім класів, шість з яких з’єднано наслідуванням. Базовий класс – Людина (Person). Він складається з поля name (ім’я, рядок) та методів введення-виведення, зокрема в файл у форматі JSON, для реалізації чого підключається додаткова бібліотека nlohmann/json.hpp. Створюється об’єкт класу nlohmann::json, якому передаються ключі і їх значення для запису в текстовий файл. За цими ключами і відбувається введення інформації в файл, що є досить практичним, надійним і однозначим способом, який до того ж є зручним.

Від класа Person йде клас Студент (Student), який його наслідує. Цей клас вже визначає людину як студента та має поля sredn (середній бал, дійсне число), course (курс, ціле число), marks (оцінки, вектор з цілих чисел), sch\_sh (стипендія, дійсне число). Аналогічно до Person, він містить методи введення/виведення, також містить методи обчислення середнього балу, обчислення стипендії (рис.1).

Обрахування стипендії виконується наступним чином:

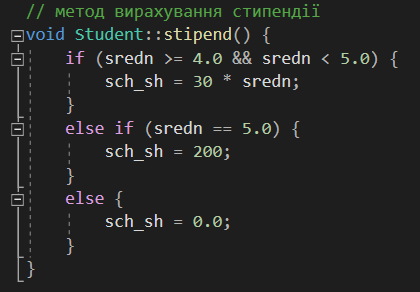


рис.1

Далі маємо клас Заліковка (Zachetka), батьківським для якого є Person. Цей клас також містить методи введення/виведення. Поля : lecturers (викладачі, вектор з рядків), disciple (дисципліни, вектор з рядків). Як і для marks, для них реалізовано методи введення.

Клас Співробітник (Employee) є наслідником класу Person. Містить методи введення/виведення. Поля: oklad (оклад, дійсний тип), stazh (стаж, цілий тип). Додатково крім методів введення-виведення є методи для встановлення\отримання значень цих полів.

Клас Викладач (Lecturer) є насілдником Employee та визначає співробітника як викладача. Містить поля: step (степінь, рядок), salary (зарплата, дійсний тип). Цей клас окрім методів введення/виведення та встановлення/отримання, містить методи обрахування заробітної плати викладача (рис.2), яка залежить від стажу та наукового степеня. Алгоритм обчислення наступний:

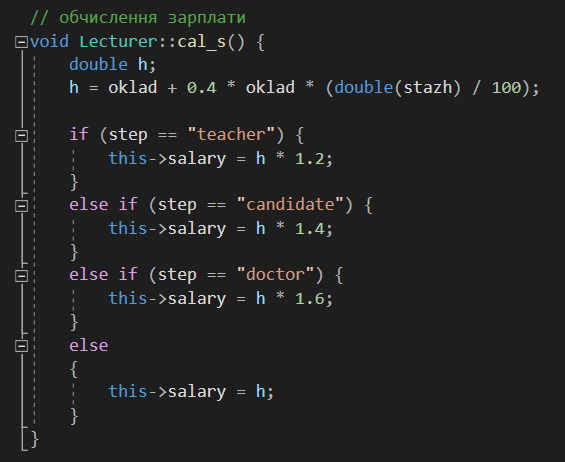


рис. 2

Клас Спеціаліст (Specialist) є нащадком класу Lecturer. Він визначає спеціальність викладача, або те, яку дисципліну він викладає. Містить поле spec (спеціальність, рядок). Має методи введення/виведення, встановлення/отримання.

Класи для роботи з файлами там массивами даних.

Клас File\_stud містить поле filename (назва файлу, рядок) та чотири методи: input\_stud\_information() – введення інформації про студентів (цей метод видаляє попередні записи в файлі), add\_stud\_information() – додавання інформації про студента в файл, де вже є якісь дані, del\_stud() –

видалення інформації про студента, get\_stud() – повертає дані про студентів з

файлу як вектор об’єктів типу Заліковка. Всі методи класу працюють з файлом, де інформація записана в форматі JSON.

Клас File\_lect є аналогічним до попереднього, протей ого методи працюють вже з викладачами. Він містить поле filename (назва файлу, рядок) та чотири методи: input\_lect() – введення інформації про викладачів (цей метод видаляє попередні записи в файлі), add\_lect() – додавання інформації про викладача в файл, де вже є якісь дані, del\_lect() – видалення інформації про викладача, get\_lect() – повертає дані про викладачів з файлу як вектор об’єктів типу Спеціаліст. Всі методи класу працюють з файлом, де інформація записана в форматі JSON, як і в попередньому випадку.

Для виконання завдань створено дві функції: poisk() - повертає відповідь чи є даний рядок в даному векторі рядкіа; інша – st() - виводить інформацію про тих студентів, які навчаються в даного викладача.

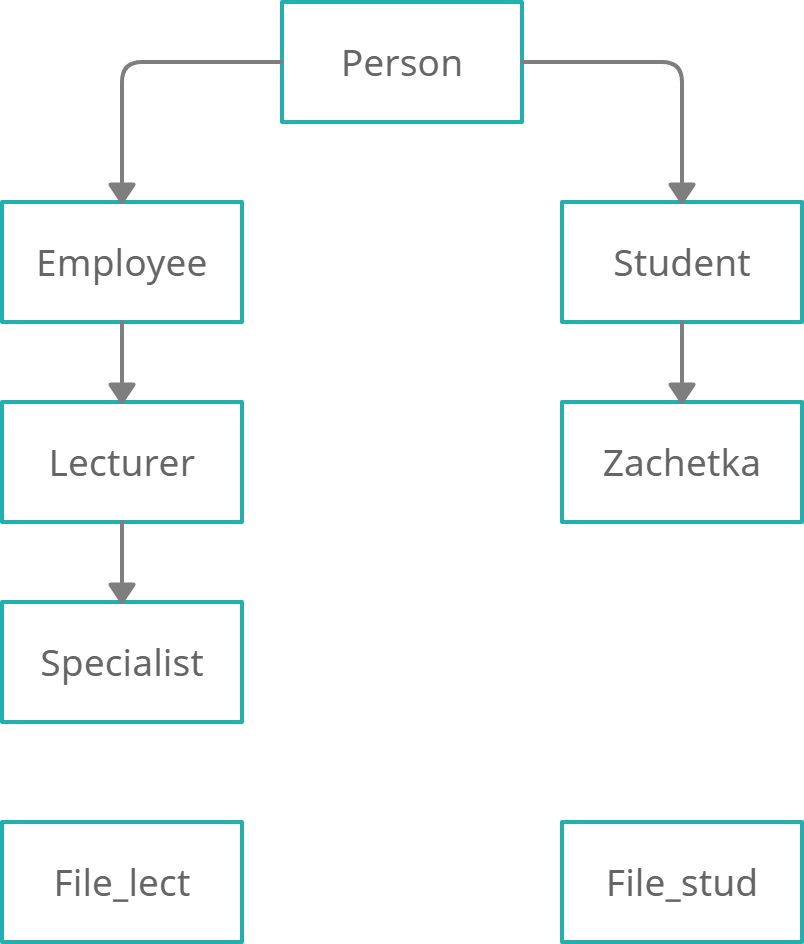


схема 1. UML-діаграма класів