МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра

автоматизованих систем обробки інформації та управління

**КУРСОВА РОБОТА**

з дисципліни Структури даних та алгоритми

на тему: інформаційна система «Факультет. Облік студентів та планування занять»

Студента 1 курсу, групи ІС-92

Тарасенка Романа Юрійовича

Спеціальності 126 «Інформаційні системи та технології»

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Члени комісії |  |  |  |
|  | (підпис) |  | (вчене звання, науковий ступінь, прізвище та ініціали) |
|  |  |  |  |
|  | (підпис) |  | (вчене звання, науковий ступінь, прізвище та ініціали) |

Київ - 2020 рік

ЗМІСТ

1. Постановка задачі.............................................................................................4
2. Основна частина...............................................................................................6
   1. Об’єктно-орієнтований аналіз області.................................................6
   2. Опис програмного забезпечення..........................................................8
      1. Зберігання інформації...........................................................................8
      2. Обробка інформації.............................................................................13
      3. Надання доступу до інформації.........................................................16
   3. Тестування............................................................................................18
3. Висновок.........................................................................................................19

**1. ПОСТАНОВКА ЗАДАЧІ**

1. Вивчити типи відношень між класами в ООП.

2. Спроектувати об’єктно-орієнтовану модель предметної галузі згідно з варіантом, визначивши необхідні для цього класи та їх структуру. При об’єктно орієнтованому проектуванні продемонструвати застосування всіх видів відношень.

3. Написати програми мовами С++ та С#, у яких реалізувати попередньо спроектовану об’єктно- орієнтовану модель.

4. Предметну галузь реалізувати окремим проектом – динамічною бібліотекою (С#) чи окремими класами з відсутнім в них введенням-виведенням (С++).

5. Інтерфейс користувача, наприклад, введення\виведення з консолі, реалізовувати окремим проектом (С#) чи класами з операціями введення\виведення (С++). Код інтерфейсу має бути простим (демонструється використання класів предметної галузі шляхом створення об’єктів та їх застосування, вимагається лише нескладна перевірка коректності вводу, введення з консолі мінімальне).

**Варіант 13. Факультет. Облік студентів та планування занять.**

На факультеті навчаються студенти певних груп та спеціальностей. Для студентів організовано робочий процес за їх навчальним планом. Факультет має фонд аудиторій. Викладачі ведуть дисципліни згідно їх навантаження. Заняття для студентів факультету плануються у загальний розклад, при чому лабораторні роботи проводяться для однієї групи в певній аудиторії, лекції – на цілий потік.

**2. ОСНОВНА ЧАСТИНА**

**2.1 Об’єктно-орієнтований аналіз області**

Моя програма може використовуватись в якості баз даних зі студентами та розкладом занять. За допомогою інтерфейсу можна легко виконати будь-яку дію. Користувач може керувати студентами, розкладом, отримувати потрібно інформацію про певного студента, отримувати розклад використовуючи фільтри.

**2.2 Головні завдання програми**

• Надати користувачу можливість створювати бази даних для обліку студентів та планування занять.

• Надати користувачу можливість редагування даних без втручання в код.

• Організувати процес отримання будь-якої інформації через інтерфейс.

**2.3 Модель програми**

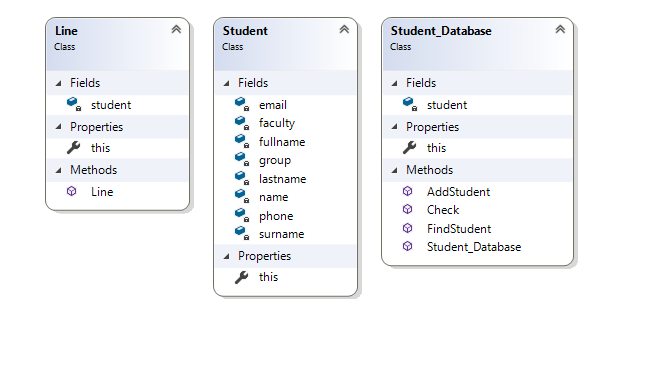
Була поставлена задача створити програму для обліку студентів та планування занять. Я створив програму, що відповідає запиту, а саме:

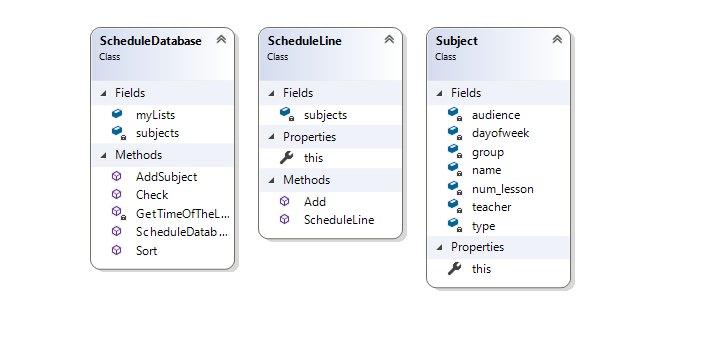
1. Створив дві таблиці (облік студентів/розклад занять);
2. Кожна таблиця має рядки, кожен рядок містить об’єкт (розклад або студента);
3. Об’єкт студент має наступні поля: ПІБ, факультет, група, електронна пошта та мобільний телефон;
4. Об’єкт предмет має наступні поля: тип(лекція/пратика), назва, ім’я викладача, день тижня, номер пари та аудиторія.
5. Рядок зі студентом створюється через базовий конструктор;
6. Рядок з розкладом створюється за допомогою методу Add;
7. База з розкладом має методи AddSubject – додати предмет в розклад, Check – отримати розклад, використовуючи фільтри. Також є локальні методи, що використовуються для сортування даних для виводу(Sort) та отримання часу початку та кінця лекції за її номером (GetTimeOfTheLesson);
8. База зі студентами має методи AddStudent – додати студента в перелік, FindStudent – отримати дані про студента з бази за прізвищем/ім’ям/по-батькові та Check – отримати розклад використовуючи фільтри;
9. Після чого було створене меню для зручного користування програмою, не втручаючись в код;
10. Інтерфейс складається з:
    1. Виведення розкладу (з фільтрами). Можна вивести тижневий розклад певного викладача/певної групи, а також перевірити яка зараз пара проходить у певного викладача/певної групи. Виводиться все в відсортованому порядку за парами;
    2. Додавання предмету до розкладу. Потрібно лише ввести дані, що просить програма і предмет буде успішно доданий. Якщо вибрана пара занята іншим предметом – дані цієї пари будуть оновлені.
    3. Виведення переліку студентів (з фільтрами). Можна вивести лише ПІБ студентів, або ж повну інформацію про кожного зі студентів. Виводиться все в відсортованому порядку за прізвищем студента;
    4. Пошук студента в базі та виведення інформації про нього. Потрібно лише ввести ім’я/прізвище/по-батькові і програма виведе всю інформацію про певного студента, якщо ж він існує в базі;
    5. Вихід з програми. Виконання програми завершується;

**2.4 Календарний план**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Назва етапів курсової роботи | Термін виконання етапів роботи (№тижня) | Підпис керівника, студента |
| 1 | Отримання теми курсової роботи | 6 |  |
| 2 | Підготовка ТЗ | 7 |  |
| 3 | Аналіз предметної області | 8 |  |
| 4 | Проектування архітектури програмної системи | 8 |  |
| 5 | Розробка сценарію роботи програми | 9 |  |
| 6 | Узгодження алгоритму з керівником | 9 |  |
| 7 | Узгодження з керівником плану тестування | 9 |  |
| 8 | Розробка програмного забезпечення | 10-13 |  |
| 9 | Тестування програми | 14 |  |
| 10 | Підготовка пояснювальної записки | 15 |  |
| 11 | Здача курсової роботи на перевірку | 16 |  |
| 12 | Захист курсової роботи | 18 |  |

**2.5 Діаграма класів програмного забезпечення**





2.6. Опис методів програми

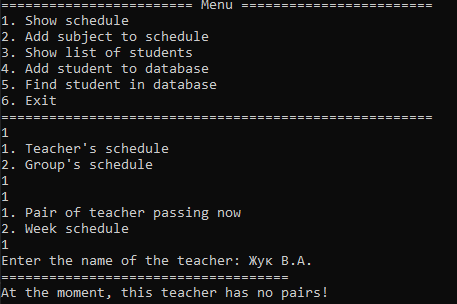
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Назва класу** | **Назва функції** | **Призначення функції** | **Опис вхідних параметрів** | **Опис вихідних параметрів** |
| Subject | this | Індексатор | - | - |
| ScheduleLine | ScheduleLine | Конструктор, що створює новий об’єкт Subject | - | - |
| ScheduleLine | Add | Створення нового рядка з об’єктом Subject | • Type – тип предмету (лекція/практика);  • name – назва предмету;  • teacher – ініціали викладача;  • dayofweek – день тижня, коли проходить пара;  • num\_lesson – номер пари;  • group – назва групи;  • audience – аудиторія; | - |
| ScheduleLine | this | Індексатор | - | - |
| ScheduleDatabase | ScheduleDatabase | Конструктор, що створює новий об’єкт SheduleLine | - | - |
| ScheduleDatabase | AddSubject | Створення нового рядка ScheduleLine з об’єктом Subject. Якщо предмет на вказаній парі існує – оновлює його. | • Type – тип предмету (лекція/практика);  • name – назва предмету;  • teacher – ініціали викладача;  • dayofweek – день тижня, коли проходить пара;  • num\_lesson – номер пари;  • group – назва групи;  • audience – аудиторія; | - |
| ScheduleDatabase | Check | Отримання тижневого розкладу по викладачу/групі, а також поточної пари | • propname – назва фільтру, за яким потрібно вивести розклад;  • value – назва групи/ім’я викладача | - |
| ScheduleDatabase | GetTimeIfTheLesson | За номером пари повертає час початку та закінчення пари | • num\_lesson -номер пари | • time – масив, де 0 елемент це дата початку пари, а 1 елемент дата закінчення. |
| ScheduleDatabase | Sort | Сортує дані для виводу | • arg – за якими даними сортування | - |
| Student | this | Індексатор | - | - |
| Line | Line | Конструктор, що створює новий об’єкт Student | • name – ім’я;  • lastname – прізвище;  • surname – по-батькові;  • faculty – факультет;  • group – група;  • email – ел. Пошта;  • phone – моб. Телефон; | - |
| Line | this | Індексатор | - | - |
| StudentDatabase | StudentDatabase | Конструктор, що створює рядок Line з об’єктом Student; | - | - |
| StudentDatabase | AddStudent | Додавання нового студента в базу; | • name – ім’я;  • lastname – прізвище;  • surname – по-батькові;  • faculty – факультет;  • group – група;  • email – ел. Пошта;  • phone – моб. Телефон; | - |
| StudentDatabase | FindStudent | Пошук студента в базі за прізвищем/ім’ям/по-батькові та вивід інформації про нього | • value – прізвище/ім’я/по-батькові | • I – номер рядка в таблиці зі студентом |
| StudentDatabase | Check | Отримання списку студентів з бази | • propname – фільтр виводу. Short – лише ПІБ, full – повна інформація по студенту; | - |

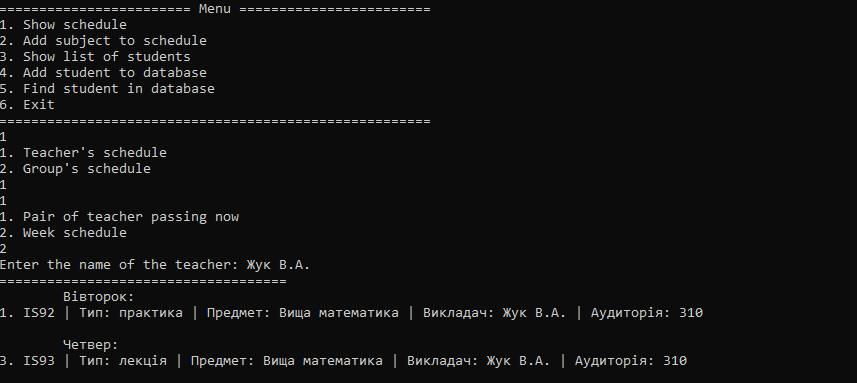
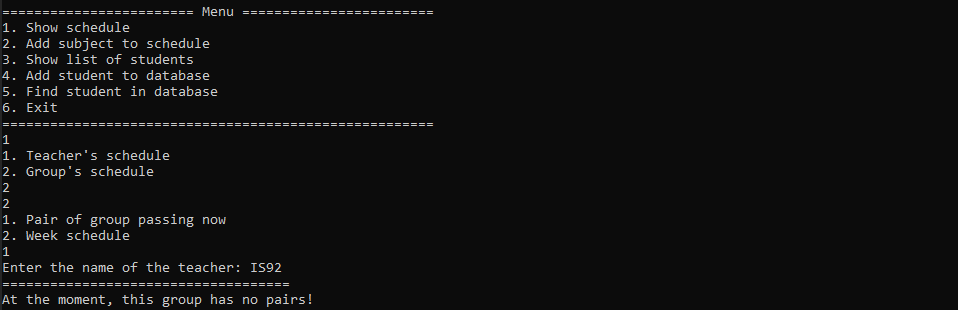
**3. КОДУВАННЯ**

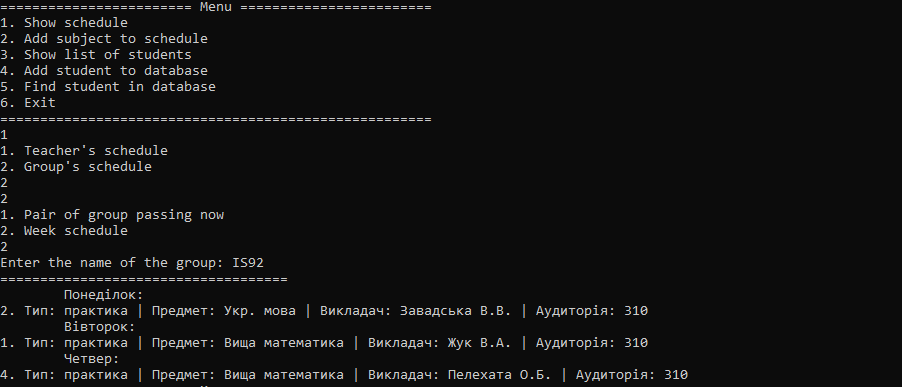
**3.1. Стратегія написання коду**

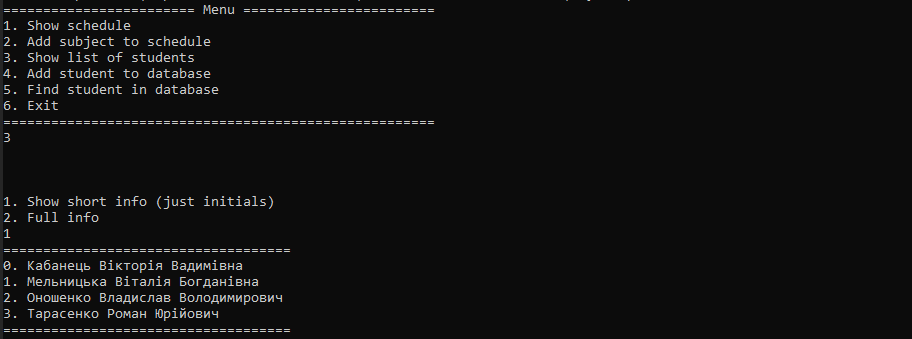
В даній роботі я створив два проекти – базу з розкладом та зі студентами. Інтерфейс написаний для використання через консоль, проте підключити інший вид інтерфейсу не буде проблемою. Спочатку був створений сам об’єкт (студент чи предмет), після чого він був поміщений в рядок таблиці, а кожен рядок є складовою таблиці.

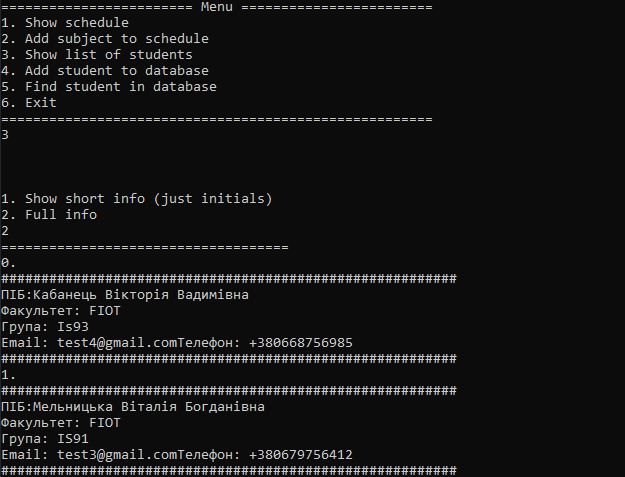
**4. ТЕСТУВАННЯ**









ВИСНОВОК

В ході виконання курсової роботи я на практиці застосував свої знання з об’єктно-орієнтованого програмування. Поглибив розуміння основних принципів об’єктно-орієнтованої парадигми програмування. При розробці програмного забезпечення відпрацював основні методи та засоби об’єктно-орієнтованого програмування. Навчився розробляти, будувати та використовувати ієрархію класів. Розроблена програма “Факультет” містить у собі застосування більшості основних понять об’єктно орієнтованої частини.