МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ УПРАВЛІННЯ

**КУРСОВА РОБОТА**

з дисципліни «Програмування - 2. Структури даних та алгоритми»

на тему: «Факультет»

Варіант №13

|  |  |  |
| --- | --- | --- |
| **Керівник :**  доц. Крамар Ю. М. |  | **Виконав:**  студент гр. ІС-02, ФІОТ  Станков А. В.  залікова книжка № 0229 |
| Допущено до захисту |  |
| І\_\_\_І \_\_\_\_\_\_\_\_\_ 2021 \_\_\_\_\_\_\_\_\_  підпис |  |
| Захищено з оцінкою  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_  оцінка підпис  І\_\_\_І \_\_\_\_\_\_\_\_\_ 2021 |  |

Київ-2021

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматизованих систем обробки інформації і управління

Спеціальність 126 «Інформаційні системи та технології»

Курс 1 Група ІС-02 Семестр 2

**ЗАВДАННЯ**

**на курсовий проект (роботу) студента**

|  |
| --- |
| ***Станкова Артема Віталійовича*** |
| (прізвище, ім’я, по батькові) |

.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Тема роботи: | | ***Розробка програмного забезпечення з використанням*** | | | |
| ***об’єктно-орієнтованого підходу.*** | | | | | |
|  | | | | | |
|  | |  | | | |
| 2. Строк здачі студентом закінченого проекту (роботи) | | | | | ***18.05.2021*** |
|  | |  | | | |
| 3. Вихідні дані до проекту (роботи): | | | | ***Факультет. Облік студентів та планування занять*** | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
|  | |  | | | |
| 4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають | | | | | |
| розробці): | ***1. Визначення класів та застосування відношень класу*** | | | | |
| ***(асоціація, агрегація: група – потік – спеціальність – факультет – розклад; композиція - розклад)*** | | | | | |
| ***2. Створення об’єктів та використання контейнерів (типи контейнерів – ліст, словник)*** | | | | | |
| ***3. Можливість додавати та видаляти студентів, а також додавати спеціальності, потоки, групи, встановлювати предмети та розклад, отримувати загальну кількість студентів, що навчаються на факультеті*** | | | | | |
|  | | | | | |
|  | |  | | | |
| 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): | | | | | |
| ***діаграми класів в нотації UML*** | | | | | |
|  | | | | | |
|  | |  | | | |
| 6. Дата видачі завдання: | | | ***23.03.2021*** | | |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№п/п** | **Назва етапів курсового роботи** | **Термін виконання етапів роботи** | **Позначки керівника про виконання завдань** |
| 1. | Отримання завдання на курсову роботу | 23.03.2021 |  |
| 2. | Огляд ТЗ |  |  |
| 3. | Аналіз предметної області |  |  |
| 4. | Проектування архітектури програмної системи |  |  |
| 5. | Розробка сценарію роботи програми |  |  |
| 6. | Узгодження з керівником інтерфейсу користувача |  |  |
| 7. | Розробка програмного забезпечення |  |  |
| 8. | Узгодження з керівником плану тестування |  |  |
| 9. | Тестування програми |  |  |
| 10. | Підготовка пояснювальної записки |  |  |
| 11. | Здача курсової роботи на перевірку |  |  |
| 12. | Захист курсової роботи | 18.05.2021 |  |

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Станков Артем Віталійович\_

(підпис) (прізвище ім'я, по батькові)

**Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище ім'я, по батькові)

**«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 р.**

# **1.ЗМІСТ**

[**1.ЗМІСТ** 4](#_Toc72136057)

[**2.ВСТУП** 5](#_Toc72136058)

[**2.1. Постановка задачі** 5](#_Toc72136059)

[**3.1. Об’єкто-орієнтовний аналіз області** 6](#_Toc72136060)

[**3.2. Модель програми** 7](#_Toc72136061)

[**3.4. Опис програмного забезпечення** 8](#_Toc72136062)

[**3.5. Діаграма класів програмного забезпечення** 16](#_Toc72136063)

[**4. ІНСТРУКЦІЯ ПО ВИКОРИСТАННЮ** 17](#_Toc72136064)

[**5.ТЕСТУВАННЯ** 18](#_Toc72136065)

[**5.1.Ініціалізація уявної бази даних** 18](#_Toc72136066)

[**5.2.Сторона клієнта** 18](#_Toc72136067)

[**6.АНАЛІЗ І УЗАГАЛЬНЕННЯ РЕЗУЛЬТАТІВ** 18](#_Toc72136068)

[**7.ПЕРЕЛІК ПОСИЛАНЬ** 19](#_Toc72136069)

# **2.ВСТУП**

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 мовою С#, у якій реалізувати попередньо спроектовану об’єктно-орієнтовану модель.

4. Предметну галузь реалізувати окремим проектом – динамічною бібліотекою (С#).

5. Інтерфейс користувача, наприклад, введення\виведення з консолі, реалізовувати окремим проектом (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).

## **2.1. Постановка задачі**

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.**3.ОСНОВНА ЧАСТИНА**

## **3.1. Об’єкто-орієнтовний аналіз області**

Програма “Faculty” буде використовуватись як додаток, в якому користувачі можуть додавати спеціальності, потоки, групи, додавати та видаляти студентів, отримувати їх кількість на факультеті. Також реалізовані можливості додавати предмети, встановлювати розклад, переглядати деталі пари.

Головні завдання програми:

1. Додати спеціальність(спеціальності)

2. Додати потік(потоки)

3. Додати групу(групи)

4. Додати студента(студентів)

5. Додати предмет(предмети)

6. Встановити розклад для групи(груп)

## **3.2. Модель програми**

Користувачеві даної програми, що представляє собою модель факультету, надається можливість працювати з структурними елементами факультету, а саме з спеціальностями, потоками, групами, їх студентами. Окрім цього здійснюється встановлення предметів та розкладу.

Основна мета курсової роботи полягає в тому, щоб надати можливість створювати на факультеті спеціальності, що містять у собі потоки. Дані функції можна виконати, обравши відповідні пункти у головному меню програми, та ввівши назву спеціальності та потоку відповідно. Потоки, у свою чергу, містять певну кількість груп, які окрім свого номеру відрізняються також і номером курсу. Групи складаються з студентів, список яких вводиться при їх створенні. Окрім цього, передбачений окремий функціонал для подальшого додавання студента до групи, шляхом введення його персональних даних, а саме прізвища, ім’я та по батькові та даних групи, до якої його потрібно його долучити. Схожим чином, використовуючи відповідний пункт меню, можна також і видаляти студентів зі груп. У будь-який момент виконання програми можна переглянути загальну кількість студентів, що навчаються на факультеті, використавши реалізований функціонал програми. Класи програми логічно пов’язані між собою агрегацією, а тому для додавання чи видалення студентів з груп, ці групи повинні бути наявні в базі програми. Для цього потрібно щоб були попередньо створені потоки, на яких знаходяться дані групи. Це, у свою чергу, зобов’язує нас попередньо створити спеціальності, на яких будуть присутні ці потоки. Якщо якась із цих умов на певному кроці виконання програми не буде виконана, буде викликано виключення, обробивши яке, програма повідомить користувачу у чому його помилка. Якщо ж ми успішно створили усі необхідні елементи, то ми можемо встановити предмети, а також розклад.

Також, користувач може у будь-який момент завершити виконання програми, натиснувши відповідну клавішу.

## **3.4. Опис програмного забезпечення**

**Клас «Group»**

Це клас, який містить назву групи, потоку, на якому вона знаходиться та її номер, а також список її студентів та методи їх додавання і видалення.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **з/п** | **Сигнатура** | **Вхідні параметри** | **Повернене значення** | **Призначення** |
| 1. | Group(int course, string flowName, int groupNumber, string[] students) | Номер курсу групи (course), назва потоку (flowName), номер групи – (groupNumber), список студентів(students) | - | Конструктор з параметрами, який заповнює поля переданими значеннями |
| 2. | void RemoveStudent(string studentName) | Ім’я студента(studentName) | - | Видалення відповідного студента зі списку групи |
| 3. | void AddStudent(string name) | Ім’я студента(studentName) | - | Додавання відповідного студента до списку групи |

**Клас «Flow»**

Це клас, який містить назву потоку, його курс, список груп, а також словник, у якому містяться предмети та їх години, методи додавання групи та предмету.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **з/п** | **Сигнатура** | **Вхідні параметри** | **Повернене значення** | **Призначення** |
| 1. | Flow(string flowName, int course) | Назва потоку(flowName) курс потоку(course) | - | Конструктор з параметрами, який заповнює поля переданими значеннями |
| 2. | void AddGroup(int groupNumber, string[] students) | Номер групи(groupNumber), список студентів(students) | - | Додавання нової групи на потік |
| 3. | void AddSubject(string SubjectName, int LectureHours, int LabHours) | Назва предмету(SubjectName), лекційні години(LectureHours), лабораторні години(LabHours) | - | Додавання нового предмету на потік |

**Клас «Speciality»**

Це клас, який містить назву спеціальності, список її потоків та метод додавання потоку.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **з/п** | **Сигнатура** | **Вхідні параметри** | **Повернене значення** | **Призначення** |
| 1. | Speciality(string specialityName) | Ім’я спеціальності  (specialityName) | - | Конструктор з параметрами, який заповнює поля переданими значеннями |
| 2. | void AddFlow(string FlowName, int Course) | Назва потоку(FlowName), курс потоку(Course) | - | Додавання нового потоку на спеціальність |

**Клас «Timetable»**

Це клас, який містить дні тижня, розклад, список вчителів, методи для знаходження вільної пари та генерації розкладу а також структури, що відповідають за одну пару та за всі пари в один час.

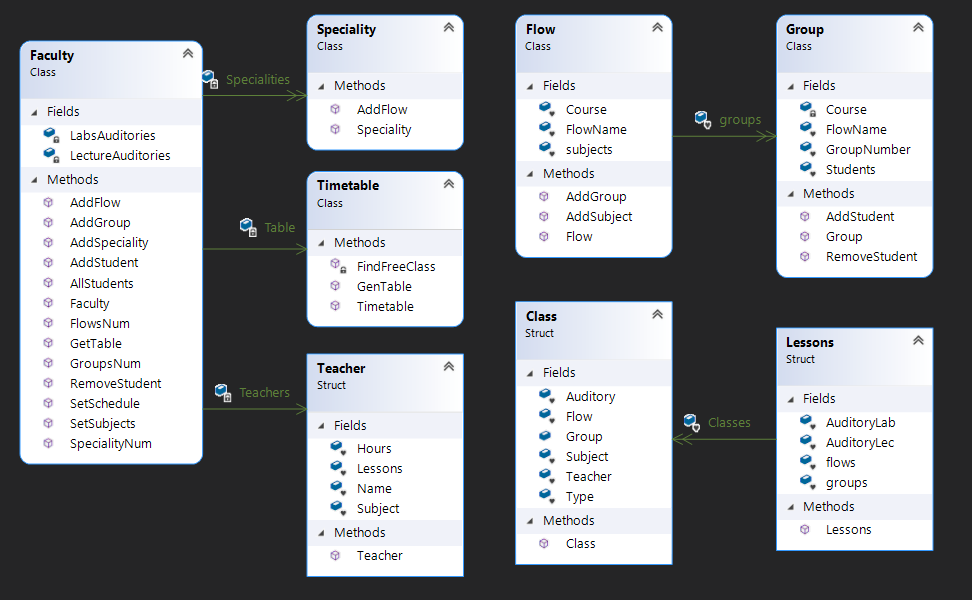
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **з/п** | **Сигнатура** | **Вхідні параметри** | **Повернене значення** | **Призначення** |
| 1. | Timetable(int[] audLec, int[] audLab, Teacher[] teachers) | Список лекційних аудиторій(audLec), список лабораторних аудиторій(audLab), список вчителів(teachers) | - | Конструктор з параметрами, який заповнює поля переданими значеннями |
| 2. | (string, int) FindFreeClass(Group curGroup, Flow curFlow, string subject, string type) | Поточна група(curGroup), потік(curFlow), предмет(subject), тип предмету(type) | None, 0 | Знаходження вільної пари |
| 3. | void GenTable(Dictionary<string, (int, int)> subj, Flow curFlow) | Список предметів та їх годин у вигляді словника(subj), обраний потік(curFlow) | - | Складання розкладу для потоку за його навчальним планом |
| 4. | Class(string subject, int auditory, string flow, string type, string teacher) | Назва предмету(subject), номер аудиторії(auditory), назва потоку(flow), типу предмету(type), ім’я вчителя(teacher) | - | Конструктор з параметрами, який заповнює поля переданими значеннями |

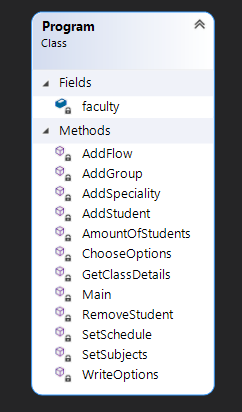
**Клас «Faculty»**

Це клас, який містить спеціальності, списки лекційних та лабораторних аудиторій, список вчителів, розклад. Також, у ньому реалізовані методи для: отримання кількості спеціальностей, потоків, груп, студентів факультету; додавання спеціальності, потоку, групи, додавання і видалення студента, встановлення предметів та розкладу, отримання інформації про пару, а також структуру, що відповідає за вчителя.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№**  **з/п** | **Сигнатура** | **Вхідні параметри** | **Повернене значення** | **Призначення** |
| 1. | Faculty(int[] aud\_lec, int[] aud\_lab, Teacher[] teachers) | Список лекційних аудиторій(audLec), список лабораторних аудиторій(audLab), список вчителів(teachers) | - | Конструктор з параметрами, який заповнює поля переданими значеннями |
| 2. | int SpecialityNum() | - | Кількість спеціальностей | Отримати кількість спеціальностей на факультеті |
| 3. | int FlowsNum() | - | Кількість потоків | Отримати кількість потоків на усіх спеціальностях |
| 4. | int GroupsNum() | - | Кількість груп | Отримати кількість груп на усіх потоках |
| 5. | void AddSpeciality(string specName) | Назва спеціальності(specName) | - | Додати спеціальність |
| 6. | void AddFlow(string specName, string flowName, int course) | Назва спеціальності(specName), назва потоку(flowName), номер курсу(course) | - | Додати потік |
| 7. | void AddGroup(string specName, string flowName, int groupName, int course, string[] students) | Назва спеціальності(specName) , назва потоку(flowName), номер курсу(course), назва групи(groupName), список студентів(students) | - | Додати групу |
| 8. | void AddStudent(string specName, string flowName, int groupName, int course, string student) | Назва спеціальності(specName) , назва потоку(flowName), номер курсу(course), назва групи(groupName), дані студента(student) | - | Додати студента до групи |
| 9. | void RemoveStudent(string specName, string flowName, int groupName, int course, string student) | Назва спеціальності(specName) , назва потоку(flowName), номер курсу(course), назва групи(groupName), дані студента(student) | - | Видалити студента з групи |
| 10. | void SetSubjects(string specName, string flowName, int course, string[] subjects, int[] lec, int[] lab) | Назва спеціальності(specName) , назва потоку(flowName), номер курсу(course), список предметів(subjects), номера лекційних аудиторій(lec), номера лабораторних аудиторій(lab) | - | Встановити предмети |
| 11. | int AllStudents() | - | Кількість студентів на факультеті | Отримати кількість студентів на факультеті |
| 12. | void SetSchedule(string specName, string flowName, int course) | Назва спеціальності(specName) , назва потоку(flowName), номер курсу(course) | - | Встановити розклад |
| 13. | string GetTable(string flow, int group, string day, int classNumber) | Назва потоку(flow), номер групи(group), день(day), номер пари(classNumber) |  | Отримати дані про пару |
| 14. | Teacher(string name, string subject, int hours) | Ім’я вчителя(name), предмет(subject), години(hours) | - | Конструктор з параметрами, який заповнює поля переданими значеннями |

## **3.5. Діаграма класів програмного забезпечення**





# **4. ІНСТРУКЦІЯ ПО ВИКОРИСТАННЮ**

**Для адміністратора**

По заданій формі додати у коді програми вчителів з різних предметів, сформувати з них масив, а також внести списки лекційних та лабораторних аудиторій.

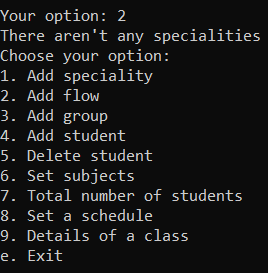
**Для користувача**

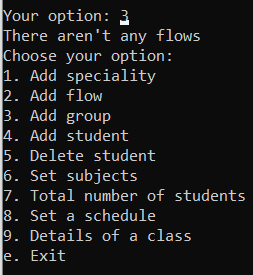
Слідувати рекомендаціям інтерфейсної частини програми.

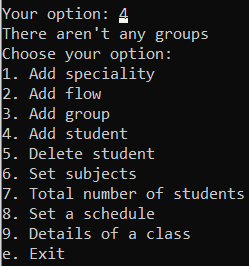
# **5.ТЕСТУВАННЯ**

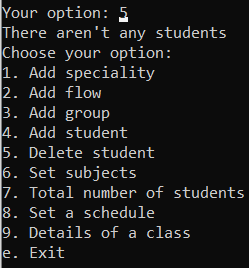
## **5.1.Приклади введення некоректних даних**

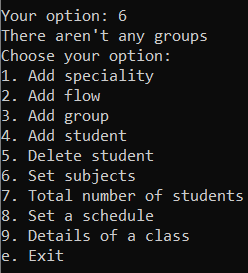
При запуску програми, у нас немає створених спеціальностей, потоків та груп, отже функції, які з ними взаємодіють, будуть повідомляти користувача про їх відсутність.

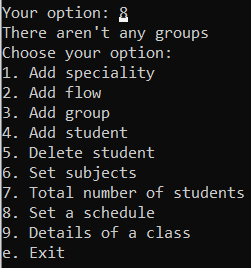


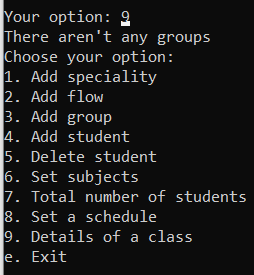




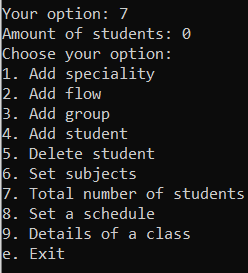








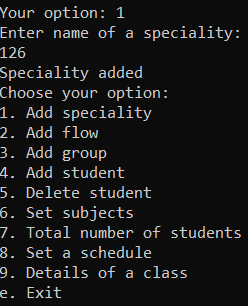
Так як немає груп, то немає і студентів, про що буде виведена інформація при виборі відповідної функції меню:

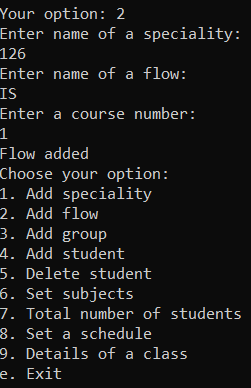


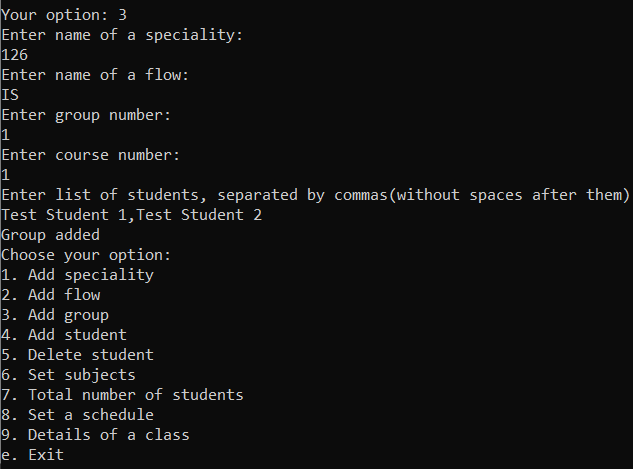
## 

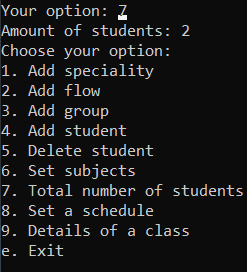
## **5.2. Введення коректних даних**

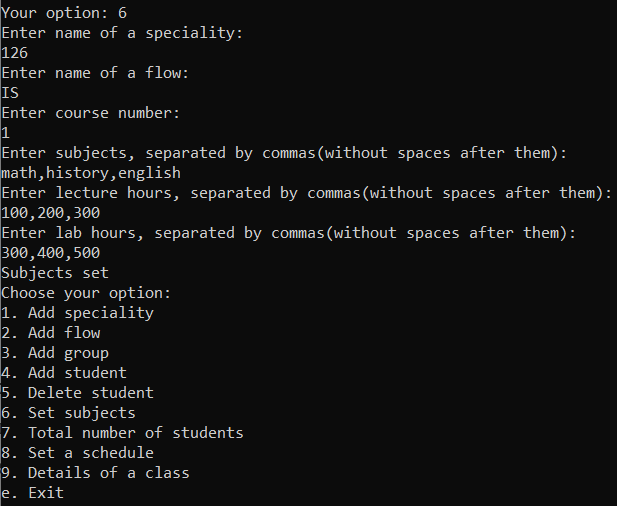
Для коректної роботи програми нам потрібно додати 1 або більше спеціальність, на ній зареєструвати 1 або більше потоків, на якому/яких створити 1 або більше груп. Після цього ми можемо встановлювати предмети та розклад, а також змінювати кількість студентів у групах, а саме додавати та видаляти їх.

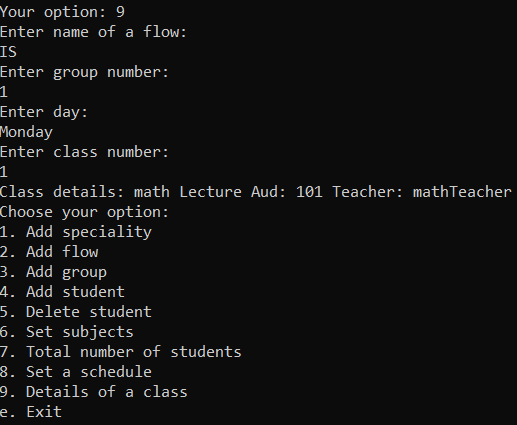


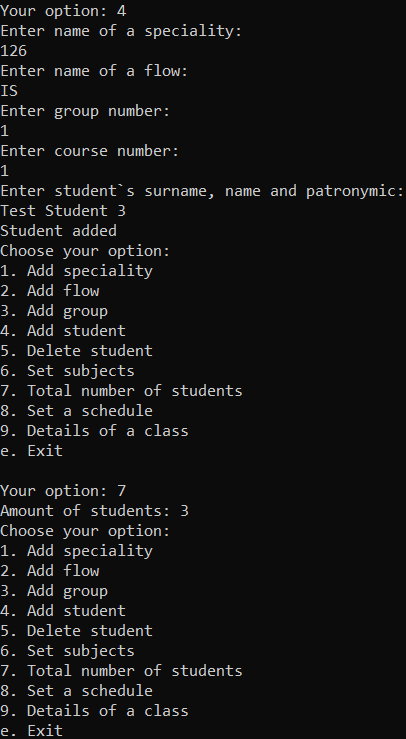


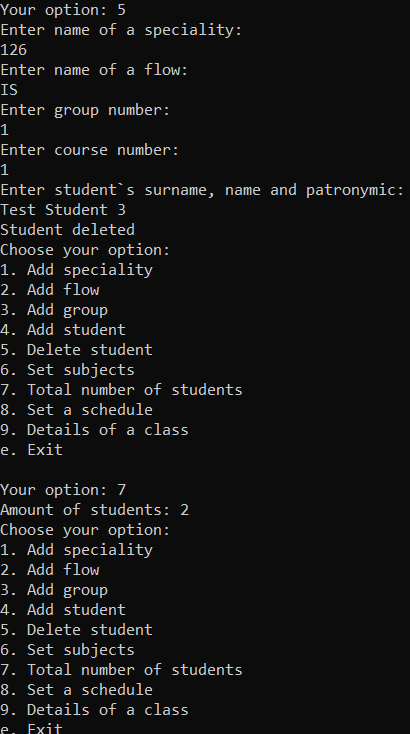












Висновок даної секції: при коректних даних програма працює як і повинна, при невиконанні необхідних умов, програма створює і оброблює відповідні виключення, повідомляючи користувача про його помилки.

# **6.АНАЛІЗ І УЗАГАЛЬНЕННЯ РЕЗУЛЬТАТІВ**

Основним завданням курсової роботи було реалізувати уявний факультет, в якому користувачі можуть створювати структурні елементи, такі як спеціальності, потоки, групи, предмети, розклад, та змінювати їх склад, наприклад, додаючи та видаляючи студентів з груп.

Критичні ситуації у роботі виявлені не були. Під час тестування було виявлено, що більшість помилок могли виникнути лише за умови неправильно введених даних на стороні користувача. Таким чином, уся отримана інформація від користувача перевіряється і подається на обробку після коректного введення.

# **7.ПЕРЕЛІК ПОСИЛАНЬ**

1. <https://metanit.com/sharp/tutorial/>
2. <https://docs.microsoft.com/en-us/dotnet/csharp/>