

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра інженерії програмного забезпечення в енергетиці

КУРСОВА РОБОТА

з дисципліни «Основи програмування. Курсова робота»

на тему: «Автоматизація робочого місця»

Студент групи _____

(прізвище, ініціали)

Керівник доц., к.т.н. Шпурик В.В.
(науковий ступінь, посада, прізвище, ініціали)

Національна оцінка _____
Кількість балів: _____ Оцінка ECTS _____

Члени комісії:

_____	<u>доц., к.т.н. Шпурик В.В.</u>
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)
_____	<u>асистент Оленєва К.М.</u>
(підпис)	(вчене звання, науковий ступінь, прізвище та ініціали)

Київ- 2023 рік

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Інститут Навчально-науковий інститут атомної та теплової енергетики
(повна назва)

Кафедра Інженерії програмного забезпечення в енергетиці
(повна назва)

Спеціальність 121 Інженерія програмного забезпечення

**З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ СТУДЕНТУ**

1. Тема роботи Автоматизація робочого місця
керівник курсової роботи - доц., к.т.н. Шпурик В.В.
2. Строк подання студентом роботи 29 травня 2023 року
3. Вихідні дані до роботи: середовище розробки – Clion 2022.3.2
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) – проектування системи управління студентами, написання коду

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсової роботи	Строк виконання етапів роботи	Примітки
1	Затвердження обраної теми роботи	1 тиждень	
2	Вивчення та аналіз задачі	2 – 3 тижні	
3	Розробка алгоритмів реалізації	4 – 5 тижні	
4	Програмна реалізація роботи	6 – 10 тижні	
5	Тестування програми	10 – 11 тижні	
6	Вдосконалення коду програми	12 – 14 тижні	
7	Оформлення пояснювальної записки	15 – 16 тижні	
8	Захист	17 – 18 тижні	

Студент _____

(підпис)

(ім'я, прізвище)

Керівник курсової роботи _____

(підпис)

Вадим ШПУРИК

(ім'я, прізвище)

АНОТАЦІЯ

У цій курсовій, створено програмний продукт який автоматизує роботу працівника деканату. У даній програмі реалізовано створення груп студентів, їх подальшого редагування, або видалення. Формування загальної відомості з усіх груп, пошук студента у всій системі. Виконано на мові C++ і застосовано як об'єктно-орієнтоване програмування так і процедурне.

Обсяг пояснювальної записки 24 аркушів, кількість ілюстрацій – 16, 3 додатки

ANNOTATION

In this course, a software product has been created that automates the work of a dean's office employee. This program implements the creation of groups of students, their subsequent editing or deletion. Formation of general information from all groups, search for a student in the entire system. The program is written in C++ and uses both object-oriented and procedural programming.

The volume of the explanatory note is 24 pages, the number of illustrations is 16, 3 appendices

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ЗАВДАННЯ.....	6
1.1 Що таке C++	6
1.2 Теоретичні відомості	7
1.3 Середовище розробки Clion.....	8
1.4 Система керування студентами	8
2 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	10
2.1 Бібліотеки.....	10
2.2 Класи.....	10
3 КЕРІВНИЦТВО КОРИСТУВАЧА	15
ВИСНОВКИ.....	22
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	24

ВСТУП

На сьогоднішній день інформаційні технології дуже спрощують діяльність людини. Зазвичай автоматизація сприймається нарівні з ефективністю, оскільки спрямована поліпшити якість процесів, зробити їх простішими. Наприклад, спрощення регулярних завдань, отримання максимальних результатів та поліпшення роботи бізнесу в цілому[1]. ІТ дозволяє здійснювати більш швидку та ефективну обробку і аналіз великих обсягів даних, виконувати рутинні завдання швидше і автоматично, зменшуючи витрати часу та зусиль. Автоматизувавши роботу, з'являється можливість уникнути людських помилок та підвищити точність обробки інформації та виконання завдань, зменшити витрати на розміщення, зберігання та обробку фізичних документів, зберігати дані в електронному вигляді та забезпечувати ефективний доступ до них.

Окрім цього всього потрібно брати до уваги те що ІТ постійно розвивається і пропонує нові можливості для автоматизації роботи, впровадження нових технологій та покращення процесів.

1 АНАЛІЗ ЗАВДАННЯ

Перед початком роботи потрібно дізнатись про умови завдання, особливості мови написання, а саме C++, середовищем Clion в якому буде розроблятися програмний продукт.

1.1 Що таке C++

Найбільш поширеною мовою програмування упродовж кількох останніх десятиріч, поза жодним сумнівом, є мова C++, на підставі якої “виросло” багато сучасних мов програмування і програмних середовищ[2, с. 8].

C++ - універсальна мова загального призначення, область застосування якої – програмування систем (більших програм). C++, разом із C, не лише найпоширеніша мова програмування, але й загальноприйнята мова спілкування програмістів[3].

Розроблена Б'ярном Страуструпом в підрозділі Bell Laboratories компанії AT&T в 1979 році. Метою створення C++ було доповнення мови програмування C можливостями, зручними для більш глобальної розробки ПО, зі збереженням гнучкості, швидкості і портативності мови C. Разом з тим творці C++ прагнули зберегти сумісність цієї мови програмування з мовою C[4].

C – це фундамент на якому було побудовано C++. Мова C++ наслідує усі засоби й атрибути C і має всі його переваги. Для неї також залишається чинним принцип C, згідно з яким програміст, а не мова, несе відповідальність за результати роботи своєї програми. Саме цей момент дає змогу зрозуміти, що винахід C++ не був спробою створити нову мову програмування. Це було скоріше вдосконалення вже існуючої(і при цьому досить успішної) мови. Більшість нововведень, якими Страуструп збагатив мову C, було призначено для підтримки об'єктно-орієнтованого програмування. По суті, C++ став об'єктно-орієнтованою версією C.

Зараз мова програмування C++ , будучи однією з найпопулярніших мов програмування в світі , широко застосовується в розробці програмного забезпечення. На цій мові створюють операційні системи, різноманітні прикладні програми, драйвери пристроїв, ігри тощо. Найбільш популярними ПО, розробленими на мові C++(або з її використанням), є популярна СУБД MySQL, інтернет-браузер Mozilla Firefox, системи сімейства Windows, IDE Visual Studio, Internet Explorer, Microsoft Office і, звичайно ж, Adobe Photoshop, Adobe Illustrator і Adobe Premiere Pro повністю написані на C++. Також C++ лежить в основі ігрового движка Unity[4].

1.2 Теоретичні відомості

C++ є об'єктно-орієнтованою мовою програмування. ООП має 4 основні принципи, які розглянемо діла.

Наслідування - це річ, яка дозволяє створювати класи, похідні від інших, тим самим наслідуючи методи та поля батьківського(базового, того від якого наслідують) класу. Класи які наслідують називаються похідними, або дочірніми.

Інкапсуляція – цей механізм забороняє напряду взаємодіяти з полями об'єктами, замість цього робота з полями проводиться через методи, гетери та сетери. У мові C++ інкапсуляція реалізована через модифікатори доступу:

- private – елемент недоступний будь-яким методам інших класів, що не є «дружніми», успадкованим класам, а доступний лише «дружнім» функціям та функціям, які реалізовані безпосередньо у цьому класі;

- protected – має всі ті самі властивості попереднього модифікатора, але ще може бути доступним в успадкованих класах;

- public - члени класу оголошенні з таким модифікатором доступу, доступні в усіх функціях.

Поліморфізм – це властивість програмної системи використовувати об'єкти з однаковим інтерфейсом без інформації про їх тип та внутрішню структуру.

Інакше, поліморфізм - це використання під одним іменем різних функцій, призначених для опрацювання даних різних типів.

У C++ є такі види поліморфізму:

- статичний - перевантаження і перевизначення функцій та операцій; шаблони функцій та класів;
- динамічний – перевизначення віртуальних функцій.

1.3 Середовище розробки CLion

CLion - це інтегроване середовище розробки (IDE) для мови програмування C++ від компанії JetBrains. Воно спеціалізується на наданні потужних інструментів для розробки C++-проектів і надає зручний інтерфейс користувача, дружній до розробників.

JetBrains CLion підтримує параметри завершення коду, налаштовані стилі програмування, використання карт і різних кутів тощо. JetBrains CLion орієнтований на крос-платформну розробку з використанням GCC або Clang, з CMake як моделлю дизайну та GDB для налагодження. Завдяки аналізу коду на льоту виділяються не тільки потенційні проблеми, але і відразу пропонуються швидкі виправлення. CLion надає понад 60 перевірок коду C і C++, включаючи пошук невикористаного коду; пропущені випадки в операторі switch-case; Пошук нескінченної рекурсії та нескінченних циклів (тут CLion використовує повноцінний аналіз потоків даних); невідповідності типів і багато іншого[5].

1.4 Система керування студентами

Система керування студентами є програмним забезпеченням, спеціально розробленим для збору, зберігання, обробки та аналізу інформації про студентів. Вона надає централізований доступ до даних студентів для всіх відділів та співробітників деканату.

Основні функції системи керування студентами розглянемо далі.

Особисті дані студента. Система дозволяє зберігати та оновлювати особисті дані студентів, такі як ім'я, прізвище, адреса, контактна інформація, дата народження, стать тощо. Ця інформація може бути легко оновлювана та використовуватися для різних адміністративних процесів.

Академічні досягнення. Система дозволяє вести облік академічних досягнень студентів, включаючи дані про курси, предмети, оцінки, успішність та академічний прогрес. Це допомагає відстежувати успішність студентів, здійснювати аналіз результатів і забезпечувати потрібну підтримку для студентів, що потребують додаткової уваги.

Розклад занять. Система може містити розклад занять для студентів, викладачів та груп. Це дозволяє студентам швидко отримувати інформацію про свій розклад, включаючи аудиторії, час та дату занять. Викладачі також можуть отримувати доступ до свого розкладу та необхідної інформації про студентів.

Результати іспитів. Система забезпечує можливість вводу та збереження результатів іспитів та оцінок студентів. Це дозволяє ефективно ведення обліку академічних досягнень, а також студенти можуть отримувати свої оцінки та аналізувати свої академічні досягнення.

Звіти та аналітика. Система може генерувати різноманітні звіти та аналітичні дані, такі як звіти про успішність студентів, прогнозування закінчення програм, аналіз академічного прогресу тощо. Це допомагає деканату проводити аналіз даних та приймати обґрунтовані рішення.

Загалом, система керування студентами спрощує та автоматизує багато рутинних адміністративних завдань, забезпечує централізований доступ до інформації та поліпшує продуктивність та ефективність роботи працівників деканату.

Наш програмний продукт є спрощеною версією системи керування студентами

2 ПРОГРАМНА РЕАЛІЗАЦІЯ

У коді було застосовано об'єктно-орієнтоване програмування при створенні класів для опису предмету, студента, групи, та самого університету та методів для них. Було проведено декілька тестувань, виправлено баги, виявленні під час процесу тестування.

2.1 Бібліотеки

Розпочнемо з переліку підключених бібліотек. Для початку бібліотека – це файл з кодом програми, призначений зберігання часто використовуваних функцій, класів, сталих. При програмуванні мовою C++ бібліотекою зазвичай називають вже скомпільований код, який можна підключати до своєї програми

Бібліотеки:

- `<iostream>` - основи введення та виведення
- `<string>` - стандартні класи і шаблони для роботи з рядками
- `<vector>` - робота з векторами
- `<fstream>` - робота з файлами
- `<iomanip>` - форматований вивід
- `<exception>` - опрацювання винятків.

2.2 Класи

Всього для програми було створено 4 класи, кожен з яких мітиться в окремих файлах(.cpp і .h), далі ми розглянемо ці класи.

Subject – описує предмет, який містить 2 поля, а саме float mark - оцінка за цей предмет(число з плаваючою крапкою) та string name – назва предмету(текстова змінна).

Створено 2 конструктори, з яких 1 один відразу задає параметри, а інший пустий. Наявні гетери, сетери, деструктор, та метод toString() який повертає текстову змінну, в якій міститься інформація про предмет.

Student – описує студента, який містить 5 полів, а саме string surname, string name, string lastname – прізвище, ім'я, по-батькові студента(відповідно), vector<Subject> subjects – вектор, який містить в собі об'єкти попереднього класу(Subject), тобто предмети, int index – номер студента в групі.

Наявні 2 конструктори, один з яких пустий, а інший встановлює всі поля окрім номера студента та деструктор. Окрім конструкторів з деструктором створено в цьому класі гетери, сетери та метод toString() який повертає тестову змінну в якій міститься інформація про всі поля та в якому використовується метод toString() для виводу інформації про предмети.

Group – описує групу студентів, містить 2 поля: vector<Student> list – безпосередньо список студентів в цій групі, string name – назва групи. Створено 2 конструктори(пустий і той який задає 2 поля), деструктор, гетери та наступні методи, як далі розглянемо.

void setIndexes() – створюється цикл, який перебирає всіх студентів по порядку та встановлює кожному номер i+1 через сетер.

string toString() – створюється текстова змінна res, створюється цикл який перебирає всіх студентів і через метод toString() з класу Student додає у res.

void toFile() – створюється текстова змінна nameOfFile яка дорівнює назві групи до якої додано “.txt”, створюється файл з назвою цієї змінної та записується інформація через метод toString()

void removeFile() – створюється текстова змінна яка дорівнює назві групи з доданим “.txt”. Видаляється файл з заданою назвою.

void addStudent(const Student &a) – приймається як аргумент об'єкт типу Student. До списку всіх студентів добавляється цей об'єкт, визиваються методи setIndexes() та toFile().(встановлюються номери та перезаписуються у файл)

`void removeStudent(int num)` - отримується на вхід ціле число, яке виступає в ролі номера студента, за яким видаляється зі списку студент, визиваються методи `setIndexes()` та `toFile()`. (встановлюються номери та перезаписуються у файл)

`void refactorStudent()` – редагування даних студента. Створюються 2 змінні `string response` і `int numberOfStudent`. Запитується номер студента, якщо студента зі заданим номером не існує то, метод завершується. Якщо студент під заданим номером є, запитується що конкретно користувач хоче змінити у студента та змінює в процесі запитуючи нові дані з відповідними перевірками.

`void toInfo()` – вивід даних групи студентів через цикл у вигляді таблиці.

`void toInfoF(ofstream& file)` - запис даних групи у вигляді таблиці у файл.

`Unik` – описує університет, який має лише 1 поле `vector<Group> groups` – список груп які містяться у в університеті. Створено конструктор який приймає 1 аргумент та деструктор. Далі ми переглянемо методи даного класу.

`void printNames()` – цей метод виводить назви груп які містяться у цьому об'єкті.

`void printall()` – створюється цикл, який перебирає всі групи у списку та виводиться їх використовуючи метод `toString()` з класу `Group`.

`void printGroup()` – запитується у користувача назву групи та передається у змінну `string nameOfGroup` і за допомогою циклу проходиться пошук групи з заданою назвою і якщо його знайдено виводиться інформація про групу.

`void removeFile()` – створюється текстова змінна яка дорівнює назві групі з доданим “.txt”. Видаляється файл з заданою назвою.

`void addGroup(Group a)` – на вхід методу дається група , яка відразу використовує методи `setIndexes()` і `toFile()` і добавляється в список груп(`groups`) .

`void createGroup()` – створюється група яка відразу добавляється в університет. Створюється відразу `Group tempGroup` та `string nameOfGroup`(назва групи), `vector<Student> students`(список студентів), `int numberOfStudents`(кількість студентів у групі). Метод просить ввести кількість студентів у групі, потім на основі цього створює цикл який при кожній ітерації створює нового студента через метод

createStudent() та додає його до students, після закінчення циклу через конструктор створюється на основі отриманих даних група та додає її в список груп через метод addGroup().

Student createStudent() – створює Student temp, string res, int numberOfSub(кількість предметів), float mark(оцінка за предмет), vector<Subject> subjects(список предметів) Текстова змінна буде застосовуватись для вводу тестових даних для сетерів. Вводимо кількість предметів для студента, вводимо назви та оцінки. Повертаємо Student temp.

void removeStudent() – створюються змінні: string res, int num(номер групи в списку груп університету), bool isHere(чи наявна група), int index(номер студента). Спочатку запитується група з якої хоче користувач видалити студента , перевіряється, якщо групу знайдено, то булевій змінній надається значення true і у змінну num передається значення ітератора на якому назви збіглись і далі запитується номер в групі студента якого хоче користувач видалити, коли студента під заданим номером буде знайдено, програма перепитає чи дійсно користувач хоче видалити студента. Якщо так, то викликається метод removeStudent() з класу Group, якщо ні, то метод завершується нічого не видаливши.

void addStudent() - створюються змінні: string res, int num(номер групи в списку груп університету), bool isHere(чи наявна група), якщо група наявна, то створюється Student temp який дорівнює результату методу createStudent() викликається метод addStudent() з класу Group, якщо ні, то метод завершується.

void refactorGroup() - створюються змінні: string res, int num(номер групи в списку груп університету), bool isHere(чи наявна група), якщо група наявна, то викликається метод refactorStudent() з класу Group, якщо ні, то метод завершується.

void removeGroup() - створюються змінні: string res, int num(номер групи в списку груп університету), bool isHere(чи наявна група), якщо група наявна, то програма перепитає чи дійсно користувач хоче видалити вказану групу, після того як користувач підтвердить свій вибір, то буде викликаний метод removeFile() з класу Group та видалиться група з groups.

`void refactor()` – створюється `string res`, виводиться інформація про те, що можна вибрати три дій(додати, видалити, редагувати студента в групі). Якщо користувач ввів “add”, то буде викликано метод `addStudent()`. Якщо буде введено “remove”, то буде викликано `removeStudent()`. Якщо “refactor” то викликається метод `refactorGroup()`.

`void getInformation()` – виводиться початок таблиці, починається цикл який викликає метод `toInfo()`. Потім створюється змінна `string res`, і програма питає чи записати цю таблицю у файл. Запис проводиться у файл під назвою “Відомість.txt”. Створюється цикл, який проходить всі групи та викликає метод `toInfoF()` для запису у файл .

`void find()` – створюються змінні `string surname`, `string name`, `string lastname`(прізвище, ім'я, по-батькові), проситься ввести відомі дані, якщо невідомо то потрібно ввести “0”. Потім на основі того, які дані відомі створюються цикли з відповідними умовами, якщо були знайдені збіги то студент виводиться.

2.3 Методи поза класами

`void commands()` – вивід списку команд

`void menu()` - створено змінну для відповіді `string response`, в залежності від того яку ввів команду, виконується певна дія.

`void clear()` – викликається `system(“clear”)` та очищується консоль.

Окрім цього створено вручну 3 групи з готовими даними для роботи.

3 КЕРІВНИЦТВО КОРИСТУВАЧА

Після старту програми на з'являється список команд, які можна ввести та просить ввести одну з них, вигляд стартового меню, яке містить список команд зображено на рисунку 3.1.

```
=====Команди=====
--> Створити групу - createGroup
--> Видалити групу - removeGroup
--> Переглянути всі групи - printAll
--> Переглянути певну групу - print
--> Вивести назви груп - groups
--> Сформувати відомість - info
--> Редагувати групу - refactor
--> Пошук студента по всіх групах - find
--> Вивести список команд - commands
--> Вийти - exit <--
=====
--->:
```

Рисунок 3.1 – Головне меню

Ввівши неправильну команду, то виведеться повідомлення, про те, що користувач не правильно ввів команду та просить ще раз ввести команду, так як зображено на рисунку 3.2.

```
=====
--->:
В
=====
Невідома команда!
=====
--->: _
```

Рисунок 3.2 – Помилка при введенні команди

Ввівши команду createGroup програма просить ввести ім'я, кількість студентів у групі та їх дані, приклад введення даних для новоствореної групи зображено на рисунку 3.3.

```

--->:createGroup
=====
Введіть назву групи:ТВ-24
Скільки студентів у групі буде?(1-10):3
-----Введіть дані-----
Ім'я:Василь
Прізвище:Пупкін
По-батькові:Бітькович
Скільки у нього буде предметів?(максимум 5):1
Назва предмету:АСД
Оцінка за цей предмет(1 - 10):4.56
-----Введіть дані-----
Ім'я:Олександр
Прізвище:Михайлович
По-батькові:Анатолійович
Скільки у нього буде предметів?(максимум 5):3
Назва предмету:ОП
Оцінка за цей предмет(1 - 10):11
Неправильно введено!Оцінка за цей предмет(1 - 10):10
Назва предмету:МАТАН
Оцінка за цей предмет(1 - 10):2
Назва предмету:АНГЛ
Оцінка за цей предмет(1 - 10):6.3456
-----Введіть дані-----
Ім'я:Руслан
Прізвище:Макаринський
По-батькові:Батькович
Скільки у нього буде предметів?(максимум 5):0
=====
--->:

```

Рисунок 3.3 – Створення нової групи

Ввівши команду `removeGroup` програма запитує назву групи яку користувач хоче видалити, якщо користувач правильно ввів назву, то користувач повинен підтвердити свій вибір, приклад видалення є на рисунку 3.4.

```

--->:removeGroup
=====
Введіть назву групи:d
Групу не знайдено!
=====
--->:removeGroup
=====
Введіть назву групи:ТВ-21
Ви впевнені що хочете видалити групу?:так
Видалено.

```

Рисунок 3.4 – Видалення групи

Для виведення списків усіх груп потрібно ввести команду `printAll`, ввівши цю команду користувачу буде виведено подібним чином як на рисунку 3.5.


```

--->:printAll
=====
-----ТВ-22-----
1. Карпенко Микита Віталійович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8;);
2. Кошинський Віталік Іванович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8;);
3. Оніщук Максим Іванович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8;);
4. Кривой Володимир Ігорович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5;);
5. Лапа Константин Володимирович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5;);
6. Макаринський Руслан Константинович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5;);
7. Михайлович Олександр Анатолійович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5; ІНТ: 10.0;);
8. Омельченко Вероніка Олександрівна(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5; ІНТ: 10.0;);
9. Переход Богдан Русланович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5; ІНТ: 10.0;);

-----ТВ-23-----
1. Стариченко Дмитро Анатолійович(Предмети: КПІ: 4.4;);
2. Полупан Михайло Дмитрович(Предмети: КПІ: 4.4;);
3. Семенчук Олександр Ігорович(Предмети: КПІ: 4.4;);
4. Сівоздрав Артем Володимирович(Предмети: КПІ: 4.4; АНГЛ: 5.5;);
5. Сущенко Дмитро Михайлович(Предмети: КПІ: 4.4; АНГЛ: 5.5;);
6. Турніцький Владислав Анатолійович(Предмети: КПІ: 4.4; АНГЛ: 5.5; ЗОЖ: 7.3;);
7. Федорчук Ярослав Максимович(Предмети: КПІ: 4.4; АНГЛ: 5.5; ЗОЖ: 7.3;);
8. Чеповський Дмитро Олександрович(Предмети: КПІ: 4.4; АНГЛ: 5.5; ЗОЖ: 7.3;);

-----ТВ-24-----
1. Пупкін Василь Бітькович(Предмети: АСД: 4.5;);
2. Михайлович Олександр Анатолійович(Предмети: ОП: 10.0; МАТАН: 2.0; АНГЛ: 6.3;);
3. Макаринський Руслан Бітькович   тькович(Предмети:);

=====
--->:

```

Рисунок 3.5 – Виведення списків груп

Наступна команда, яку можна ввести це print. Вона просить ввести назву групи яку користувач хоче вивести, якщо група з заданою назвою знайдена, то виведеться список, на рисунку 3.6 показано вивід групи «ТВ-22»:

```

--->:print
=====
Яку групу вивести?:ТВ-22
-----ТВ-22-----
1. Карпенко Микита Віталійович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8;);
2. Кошинський Віталік Іванович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8;);
3. Оніщук Максим Іванович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8;);
4. Кривой Володимир Ігорович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5;);
5. Лапа Константин Володимирович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5;);
6. Макаринський Руслан Константинович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5;);
7. Михайлович Олександр Анатолійович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5; ІНТ: 10.0;);
8. Омельченко Вероніка Олександрівна(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5; ІНТ: 10.0;);
9. Переход Богдан Русланович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5; ІНТ: 10.0;);

=====
--->:

```

Рисунок 3.6 – Вивід заданої групи

Можна ввести ще команду groups, яка виводить назви всіх груп, так як зображено на рисунку 3.7.

```

--->:groups
=====
Групи:ТВ-22 ТВ-23 ТВ-24 ;
=====

```

Рисунок 3.7 – Вивід назв груп

Наступна команда – це info. Дана команда формує відомість у вигляді таблиці та запитує чи записати відомість у файл, відомість зображено на рисунку 3.8.

№	ПІБ	Предмети	Оцінки(відповідно)
1	Карпенко Микита Віталійович	ОКСМ; ЛАТАГ;	6.7; 3.8;
2	Кошинський Віталік Іванович	ОКСМ; ЛАТАГ;	6.7; 3.8;
3	Юнішук Максим Іванович	ОКСМ; ЛАТАГ;	6.7; 3.8;
4	Кривой Володимир Ігорович	ОКСМ; ЛАТАГ; АНГЛ;	6.7; 3.8; 5.5;
5	Лапа Константин Володимирович	ОКСМ; ЛАТАГ; АНГЛ;	6.7; 3.8; 5.5;
6	Макаринський Руслан Константинович	ОКСМ; ЛАТАГ; АНГЛ;	6.7; 3.8; 5.5;
7	Михайлович Олександр Анатолійович	ОКСМ; ЛАТАГ; АНГЛ; ІНТ;	6.7; 3.8; 5.5; 10.0;
8	Омельченко Вероніка Олександрівна	ОКСМ; ЛАТАГ; АНГЛ; ІНТ;	6.7; 3.8; 5.5; 10.0;
9	Переход Богдан Русланович	ОКСМ; ЛАТАГ; АНГЛ; ІНТ;	6.7; 3.8; 5.5; 10.0;
1	Стариченко Дмитро Анатолійович	КПІ;	4.4;
2	Полупан Михайло Дмитрович	КПІ;	4.4;
3	Семенчук Олександр Ігорович	КПІ;	4.4;
4	Сівоздрав Артем Володимирович	КПІ; АНГЛ;	4.4; 5.5;
5	Сущенко Дмитро Михайлович	КПІ; АНГЛ;	4.4; 5.5;
6	Турніцький Владислав Анатолійович	КПІ; АНГЛ; ЗОЖ;	4.4; 5.5; 7.3;
7	Федорчук Ярослав Максимович	КПІ; АНГЛ; ЗОЖ;	4.4; 5.5; 7.3;
8	Чеповський Дмитро Олександрович	КПІ; АНГЛ; ЗОЖ;	4.4; 5.5; 7.3;
1	Пупкін Василь Бітькович	АСД;	4.5;
2	Михайлович Олександр Анатолійович	ОП; МАТАН; АНГЛ;	10.0; 2.0; 6.3;
3	Макаринський Руслан Батькович		

Записати у файл відомість?:

Рисунок 3.8 – Виведення відомості

На рисунку 3.9 показано приклад записаної у файл відомості про всіх студентів.

Ввівши команду find програма запитує які дані відомі про студента, якщо щось невідомо, то користувач повинен ввести у позицію «0», на рисунку 3.10 зображено, яким чином проходить пошук студентів.

Відомість.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

№	ПІБ	Предмети	Оцінки(відповідно)
1	Карпенко Микита Віталійович	ОКСМ; ЛАТАГ;	6.7; 3.8;
2	Кошинський Віталік Іванович	ОКСМ; ЛАТАГ;	6.7; 3.8;
3	Оніщук Максим Іванович	ОКСМ; ЛАТАГ;	6.7; 3.8;
4	Кривой Володимир Ігорович	ОКСМ; ЛАТАГ; АНГЛ;	6.7; 3.8; 5.5;
5	Лапа Константин Володимирович	ОКСМ; ЛАТАГ; АНГЛ;	6.7; 3.8; 5.5;
6	Макаринський Руслан Константинович	ОКСМ; ЛАТАГ; АНГЛ;	6.7; 3.8; 5.5;
7	Михайлович Олександр Анатолійович	ОКСМ; ЛАТАГ; АНГЛ; ІНТ;	6.7; 3.8; 5.5; 10.0;
8	Омельченко Вероніка Олександрівна	ОКСМ; ЛАТАГ; АНГЛ; ІНТ;	6.7; 3.8; 5.5; 10.0;
9	Переход Богдан Русланович	ОКСМ; ЛАТАГ; АНГЛ; ІНТ;	6.7; 3.8; 5.5; 10.0;
1	Стариченко Дмитро Анатолійович	КПІ;	4.4;
2	Полупан Михайло Дмитрович	КПІ;	4.4;
3	Семенчук Олександр Ігорович	КПІ;	4.4;
4	Сівоздрав Артем Володимирович	КПІ; АНГЛ;	4.4; 5.5;
5	Сущенко Дмитро Михайлович	КПІ; АНГЛ;	4.4; 5.5;
6	Турніцький Владислав Анатолійович	КПІ; АНГЛ; ЗОЖ;	4.4; 5.5; 7.3;
7	Федорчук Ярослав Максимович	КПІ; АНГЛ; ЗОЖ;	4.4; 5.5; 7.3;
8	Чеповський Дмитро Олександрович	КПІ; АНГЛ; ЗОЖ;	4.4; 5.5; 7.3;
1	Пупкін Василь Бітькович	АСД;	4.5;
2	Михайлович Олександр Анатолійович	ОП; МАТАН; АНГЛ;	10.0; 2.0; 6.3;
3	Макаринський Руслан Батькович		

Рисунок 3.9 – Записана відомість у файл

```

--->:find
=====
Введіть відомі дані(0 - якщо невідомо)
Ім'я:0
Прізвище:0
По-батькові:Ігорович
ТВ-21: 5. Грицюк Максим Ігорович(Предмети: АСД: 2.3; ОП: 2.5; МАТАН: 7.8; АНГЛ: 2.5;);
ТВ-22: 4. Кривой Володимир Ігорович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5;);
ТВ-23: 3. Семенчук Олександр Ігорович(Предмети: КПІ: 4.4;);
=====
--->:

```

Рисунок 3.10 – Пошук студентів

Ввівши команду refactor програма пропонує виконати 3 дії: додати, видалити, або редагувати студента в групі(команди: add, remove, refactor), на рисунках 3.11, 3.12, 3.13 зображено операції додавання, видалення та редагування студентів у групах відповідно.

```

--->:refactor
=====
-> add - додати студента
-> remove - видалити студента
-> refactor - змінити дані студента
>:add
Введіть назву групи:ТВ-21
-----Введіть дані-----
Ім'я:Павло
Прізвище:Павленок
По-батькові:Павлович
Скільки у нього буде предметів?(максимум 5):3
Назва предмету:АСД
Оцінка за цей предмет(1 - 10):8.65
Назва предмету:
Ав
Оцінка за цей предмет(1 - 10):5
Назва предмету:ОП
Оцінка за цей предмет(1 - 10):10
=====
--->:

```

Рисунок 3.11 – Додавання студента в задану групу

```

--->:refactor
=====
-> add - додати студента
-> remove - видалити студента
-> refactor - змінити дані студента
>:remove
Введіть назву групи:ТВ-22
Введіть номер студента:4
4. Кривой Володимир Ігорович(Предмети: ОКСМ: 6.7; ЛАТАГ: 3.8; АНГЛ: 5.5;);
Ви впевнені що хочете видалити студента зі списку?:так
Видалено.
=====
--->:

```

Рисунок 3.12 – Видалення студента з групи

```

--->:refactor
=====
-> add - додати студента
-> remove - видалити студента
-> refactor - змінити дані студента
>:refactor
Введіть назву групи:ТВ-23
Введіть номер студента:6
6. Турніцький Владислав Анатолійович(Предмети: КПІ: 4.4; АНГЛ: 5.5; ЗОЖ: 7.3;);
Введіть параметр який хочете змінити:прізвище
Введіть новий варіант:Пилпенко
Змінено.=====
--->:

```

Рисунок 3.13 – Зміна даних студента в групі

Для того щоб вивести список команд потрібно ввести команду `command`, програма виведе інформацію, як на рисунку 3.14.

```
--->:command
=====
=====Команди=====
--> Створити групу - createGroup
--> Видалити групу - removeGroup
--> Переглянути всі групи - printAll
--> Переглянути певну групу - print
--> Вивести назви груп - groups
--> Сформувати відомість - info
--> Редагувати групу - refactor
--> Пошук студента по всіх групах - find
--> Вивести список команд - commands
--> Вийти - exit <--
=====
=====
--->:
```

Рисунок 3.14 – Вивід спису команд

Для того щоб очистити консоль потрібно ввести команду `clear`. Приклад зображено на рисунку 3.15.

```
=====
--->:
```

Рисунок 3.15 – Очищення консолі

Для того щоб завершити роботу потрібно ввести команду `exit`, на рисунку 3.16 зображено кінцевий екран консолі.

```
--->:exit
=====
Бувай!!!!
z8999@DESKTOP-0BND354 /cygdrive/d/G/Course_work1
```

Рисунок 3.16 – Завершення роботи

Кожного разу програма починає роботу з однаковими даним, незалежно від того що було зроблено попереднього прогону програми.

ВИСНОВКИ

В ході виконання курсової роботи, було закріплено знання мов програмування C++. В особливості навички об'єктно-орієнтованого програмування для створення класів, які описують навчальний предмет, студента, групи, та самого університету. Було розроблено можливість зміни даних а саме:

- створення групи;
- видалення групи;
- редагування групи(редагування, створення та видалення студента в групі);
- пошук студента за деякими даними;
- виведення інформації та створення відомості на основі.

Роботу над курсовою можна розділити на три етапи.

Перший етап – вибір теми для роботи, аналіз поставлених завдань та створення алгоритмів для рішення задач

Другий етап – це програмна реалізація завдання, а саме розробка програмного коду, використовуючи знання об'єктно-орієнтованого та процедурного програмування.

Третій етап – це заключний етап, в процесі якого проводимо тестування, вдосконалюємо код на основі виявлених багів та помилок.

Мова програмування C++ є дуже популярною та широко використовується в індустрії програмного забезпечення. Вона має кілька переваг, які роблять її хорошим вибором для багатьох проектів. Вона дозволяє писати достатньо ефективний код з високою швидкодією та рівнем абстракції. Вона має прямий доступ до пам'яті та можливість оптимізувати код для вимогливих за ресурсами додатків. C++ підтримує ООП, що дозволяє створювати модульний та легко розширювальний код.

Вона може бути використана для розробки різних платформ, серед яких Windows, Linux та багато інших. C++ використовується у багатьох сферах, а саме розробка вбудованих систем, графічні програми, ігри, високопродуктивні програми, та багато чого іншого.

Однак, варто враховувати, що C++ є мовою з високим рівнем складності та має велику кількість можливостей, що може призвести до складнощів для початківців.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ІТ-автоматизація: нові тренди 2021 року [Електронний ресурс] – Режим доступу: <https://litiko.com/uk/info-czentr-uk/statti/it-avtomatyzacziya-novi-trendy-2021-roku/>
2. С++. Основи програмування. Теорія та практика. : підручник / О. Трофименко та ін. ; ред. О. Трофименко. Одеса : Фенікс, 2010. 544 с.
3. Інформатика – теорія – мова програмування С++. Гімназія Євшан. [Електронний ресурс] – Режим доступу: <https://yevshan.com.ua/info/007/content/content3.html>
4. Уроки програмування на С++. aCode. року [Електронний ресурс] – Режим доступу: <https://acode.com.ua/uroki-po-cpp/>
5. JetBrains Clion. І.Т.Про. [Електронний ресурс] – Режим доступу: <https://itpro.ua/product/jetbrains-clion/?tab=description>