**Практична робота4. Інкапсуляція на основі класів**

**Мета роботи**

Ознайомитись з об'єктно-орієнтованим підходом при розробці програм на мові С++ та отримати практичні навички у застосуванні інкапсуляції на основі класів.

**Хід роботи**

1) Ознайомитись з методичними вказівками до лабораторної роботи та темою "Об'єктно-орієнтований підхід. Класи та їх члени";

2) розробити схеми алгоритмів методів класу згідно із завданням за варіантом;

3) за схемами алгоритмів виконати оголошення та визначення класу і його членів;

4) розробити схему алгоритму програми, що демонструє роботу зі сформованим класом;

5) за схемою алгоритму написати програму на мові С++;

6) зробити висновки;

7) підготувати звіт про виконання лабораторної роботи, що включає наступні пункти: номер, тема, мета та хід лабораторної роботи, завдання, схема алгоритму програми, лістинг програми з коментуванням кожної інструкції, результат роботи програми (скріншот), висновки.

**Теоретичні відомості та рекомендації до виконання**

Основою об'єктно-орієнтованного програмування є поняття класу. Клас визначає природу об'єкта і є основним механізмом інкапсуляції. Клас – це логічна структура, яка поєднує у собі змінні, константи та функції, пов'язані між собою одним напрямком та ідеєю.

Основне призначення класу – спрощення написання програм, завдяки об'єктній орієнтаціі та інкапсуляції змінних і функцій у структурі класу.

При виконанні завдання до лабораторної роботи студент повинен засвоїти основні принципи побудови класів та набути вміння використовувати класи та їх члени у програмних проектах.

Виконання лабораторної роботи починається ознайомленням з теоретичними відомостями про основи побудови класів та методичними вказівками. Засвоївши їх, студент приступає до розробки класу згідно із завданням за варіантом. Необхідно створити основу свого класу та визначитися з кількістю методів, змінних та їх назвами. Назви методів мають відображати сутність дій, що вони виконують у програмі.

Після розробки класу та визначення його методів студент переходить до створення схем алгоритмів програми, що буде використовувати розроблений клас, та методів класу.

У висновках до звіту студент повинен розкрити призначення теми "Інкапсуляція на основі класів". Висновок має бути поданий в формі пояснення сфери застосування вивченого матеріалу згідно із розумінням його студентом.

**Завдання для самостійного виконання**

Розробити схему алгоритму та програму, що демонструє роботу з класом, оголошеним та визначеним згідно із завданням за варіантом. Клас застосовується для роботи з тривимірним масивом. Він повинен містити масив цілого типу даних Arr[x][y][z] та три методи (табл. 11.1):

− метод заповнення масиву за алгоритмом – конструктор класу;

− метод заповнення зрізу масиву значенням;

− метод виводу на консоль зрізу тривимірного масиву.

Тривимірний масив повинен бути побудований згідно з рис. 11.1.

***Рис. 11.1. Схематичне відображення тривимірного масиву***

Таблиця 11.1



\* Метод 2 треба виконати, вважаючи, що масив має розміри 5х5х5, виключаючи ті зрізи (матриці), які відсутні за завданням. В функцію повинна передаватися цифра, за допомогою якої і буде виконано заповнення.

**Вказівки до виконання завдання**

Завдання до лабораторної роботи включає розробку методів згідно з варіантом. Перший метод виконує заповнення тривимірного масиву за алгоритмом, що наведено у вигляді переліку трьох площин.

Наприклад, для алгоритму **XZY** заповнення повинно виконуватися у наступній послідовності: спочатку заповнюється рядок по осі **X**, потім наступний рядок, що входить до матриці **XZ**. Коли заповнення матриці закінчується, виконується перехід до наступної матриці по вісі **Y**.

Наприклад, заповнена 0 матриця по осі осі **У** масиву 5х5х5, буде мати вигляд:

Другий метод виконує заповнення фігури, що формується при поєднанні верхівок тривимірного масиву у послідовності за варіантом, обраною студентом цифрою. Фігура наведена за допомогою номерів верхівок тривимірного масиву згідно з рисунком у завданні. Елементи фігури входять до зрізу масиву, що створюється при формуванні площини, яка включає задану фігуру.

Наприклад, для фігури 136 масиву 3х3х3 (заповненого згідно з алгоритмом **XYZ**) необхідно записати нові значення в елементи, що зазначені на рис. 11.2.

***Рис. 11.2. Заповнення фігури, що задана номерами верхівок 136, цифрою***

Елементи массиву, що позначено жирним, перевизначаються обраною цифрою. На рисунку цифра "0".

Третій метод виконує відображення вертикального чи горизонтального зрізу тривимірного масиву, номер зрізу вказано у таблиці згідно з варіантом.

Наприклад, для зрізу Arr[x][y][2] масиву 3х3х3 (заповненого згідно з алгоритмом **XYZ**) функція повинна виводити 3 матрицю **XY** по осі **Z**. Рахунок елементів масиву ведеться з 0, тому матриця під номером 2 і є третьою (рис. 11.3).

***Рис. 11.3. Відображення третьоїматриці по осі Z***

Метод повинен вивести на консоль матрицю, що вказана у правій частині рисунка.

Для відображення тривимірного масиву можна скористатися наступною функцією:

//*функція виводу елементів тривимірного масиву у вигляді кубу*

**void** ShowCube(**char** x, **char** y, **char** z)

{

**char** i, j, k; //*оголошення змінних лічильників для циклів*

printf("\r\n\r\n\r\n"); //*відступ за рядками*

**for**(k=z-1; k>=0; k--) //*цикл перебору елементів по осі z*

{

**for**(j=0; j<y; j++) //*цикл перебору елементів по осі y*

{

**for**(int l=5-j; l>=1; l--) //*цикл зміщення рядка вздовж х*

printf(" "); //*вивід 3-х символів відступу*

**for**(i=0; i<x; i++) //*цикл перебору елементів по осі x*

//*вивід значення елемента та 9-ти символів відступу*

printf("%0.3i ", A[i][j][k]);

printf("\r\n"); //*перехід на новий рядок*

}

printf("\r\n\r\n\r\n"); //*відступ за рядками*

}

}

Наведену функцію зручно додати у створений клас як четвертий метод.

Основна програма має відображати результати роботи всіх трьох методів, причому спочатку на консоль треба вивести заповнений масив за допомогою методу ShowCube().

**Питання для підготовки до захисту лабораторної роботи**

1) Дайте визначення поняттю об'єкт. Що таке клас?

2) Як оголошується клас?

3) Які є специфікатори доступу до членів класу?

4) Як можуть визначатися методи класу?

5) Як застосувати методи та поля класу з основної програми?

6) Що таке конструктор та деструктор класу?