**Лабораторна робота 5**

**Тема роботи:** Композиція класів. Дружні функції

**Мета роботи:** Навчитися створювати композиції класів та проектувати структуру програми.

**Завдання лабораторної роботи:**

**Етап 1. Створення композиції класів**

1. Клас, що було розроблено в ЛР№3 розділити на 2 таких окремих класа, щоб було можливо створити композицію цих класів. (Якщо це не є можливим, створити новий клас, який має відношення «has-a» до базового).
2. Один з класів має містити поле, що є покажчиком на об’єкт іншого класу.

**Етап 2. Проектування структури програми.**

1. Програма повинна складатися з 5 файлів: main.cpp() – основна програма, кожен клас повинен бути розбитий на файл заголовку(інтерфейс) в якому описана основна структура класу без реалізації з розширенням .h приклад Class1.h та файл в якому реалізовано всі методи класу з розширенням .cpp приклад Class1.cpp. Для підключення файлів один до одного використовувати конструкцію #include
2. Оголосити у класах конструктор з параметрами за замовчуванням, методи get(), set() за необхідності і загальний showAll(), що виводить на екран повну інформацію про об’єкт класу.

**Етап 3.Функція main().**

1. Додати у цей файл необхідні бібліотеки й файли заголовків.
2. Створити й вивести на екран об’єкти похідного класу (5 екземплярів мінімум), потім створити об’єкти базового класу (5 екземплярів мінімум).
3. Створити масив для цих об’єктів.

**Етап 4. Дружня функція**

1. Вигадати функцію, що викониватиме пошук у масиві, що було створено у п.3.3. Ця функція ОБОВ’ЯЗКОВО має:
   * + в якості параметрів брати масив об’єктів, розмір масиву й дані від користувача.
2. Оголосити вигадану фунцію як friend у ,базовому класі.
3. Викликати цю фунцію у main().

**Етап 5. Звичайна функція**

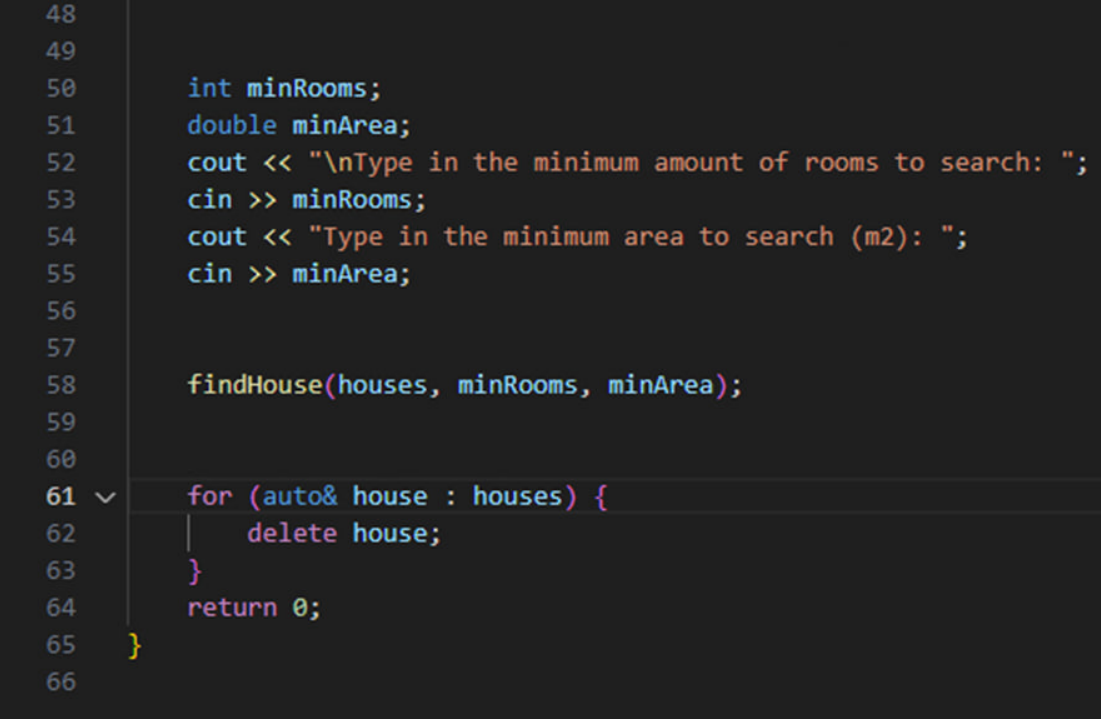
1. Створити функію, котра викониє туж саму дію, що й friend-функція, але не є friend.
2. Викликати цю фунцію у main().

**Виконання роботи:**

**main.cpp:**

**Зображення, що містить текст, знімок екрана, програмне забезпечення

Автоматично згенерований опис**

****

**House.h:**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис**

**House.cpp:**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис**

**Cottage.h:**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис**

**Cottage.cpp:**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис**

**Apartment.h:**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис**

**Apartment.cpp:**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Автоматично згенерований опис**

**Результати роботи:**

**Зображення, що містить текст, меню, знімок екрана, Шрифт

Автоматично згенерований опис**

**Зображення, що містить текст, знімок екрана, меню

Автоматично згенерований опис**

**Зображення, що містить текст, Шрифт, знімок екрана, дизайн

Автоматично згенерований опис**

**Контрольні питання:**

1. Що таке композиція?

Композиція – це створення складних функцій або об’єктів шляхом комбінування та об’єднання простіших функцій або об’єктів

1. Яка різниця між відношеннями «is-a» та «has-a»?

Відношення типу is-a означає що один клас є підвидом іншого класу, і має у собі принципи спадковості.

Відношення типу has-a означає що клас містить один або декілька екземплярів іншого класу.

1. Що таке дружні функції?

Це функція, яка має доступ до закритих членів класу, наче вона сама є членом цього класу.

1. Наведіть приклад оголошення дружньої функції та дружнього класу.

Дружня функція:

friend void output(string WelcomingMessage);

Дружній клас:

friend class Cars

1. Чому дружні функції не бажано використовувати(зловживати)?

Дружні функції мають доступ до приватних та захищених членів класу. Вони ускладнюють тестування, адже можуть змінювати приватні або захищені дані класу. А також клас що використовує багато дружніх функцій, стає менш гнучким.