

Тема 3: Спадкування, віртуальність.

Мета

Отримати навички розробки власних ієрархій класів із використанням принципу розширення та віртуальності.

1 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Створити клас Data2, використовуючи спадкування від Data1 із додаванням нових полів. Розподілити та обґрунтувати права доступу до полів.

Виділити базовий клас **BaseView** для View1 та View2 із функцією **BaseView::Display()** та наступними віртуальними функціями, котрі викликаються з неї:

```
1 protected:  
2 BaseView::showHeader();  
3 BaseView::showContent();  
4 BaseView::showFooter();
```

Перенести основний функціонал відображення в базовий клас, реалізувавши специфічну поведінку у відповідних віртуальних методах. Після вказаних модифікацій View1 та View2 повинні працювати аналогічно, як у роботі №2 із об'єктами класу Data1 та нащадками.

Створити клас View3, котрий задає специфіку відображення для об'єктів Data2. Вибрати необхідне місце у ієрархії для цього класу.

Показати роботу віртуальності на прикладі використання нащадка через покажчик на базовий клас для об'єкту Data1 та об'єкту Data2 із використанням View-класів.

№	Предметна область	Назва Data2	Поля Data2
3	3-й поверх корпусу ВК	Кімната у навчальному корпусі	Field1_data2 = Назва кмнати(рядок) Field2_data2 = кількість посадочних місць Field3_data2 = призначення кімнати

2 РОЗРОБКА ПРОГРАМИ

2.1 Засоби ООП

В ході розробки програми були використані такі засоби ООП:

- Абстракція.
- Інкапсуляція.
- Спадкування
- Поліморфізм

2.2 Ієрархія та структура класів

Ієрархія класів наведена на рис. 2.1

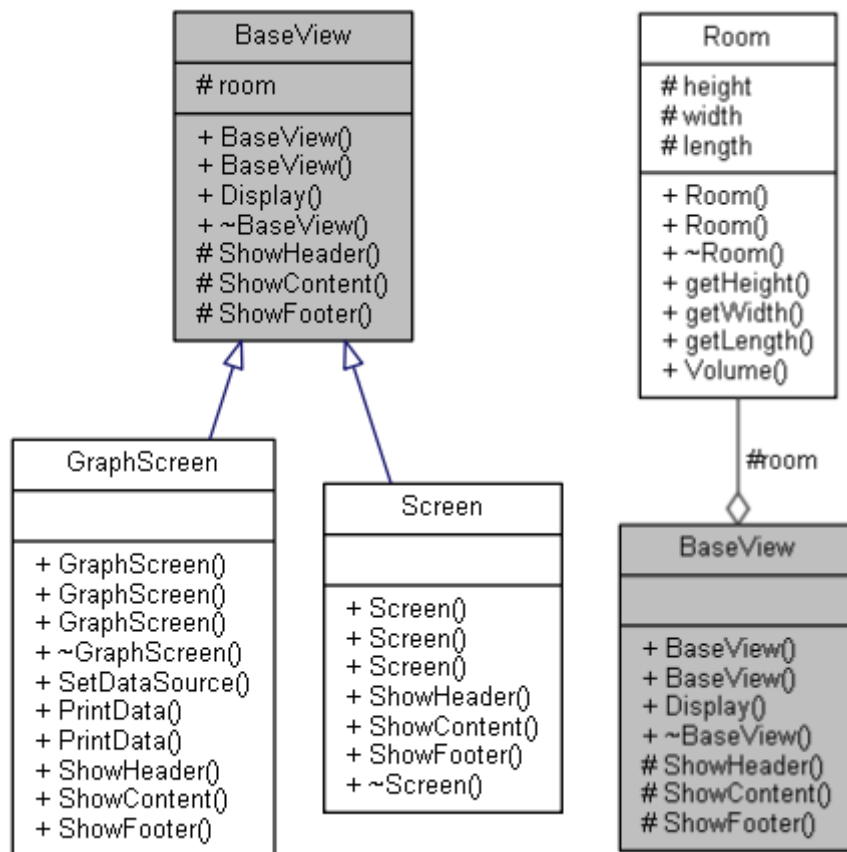


Рисунок 2.1 – Ієрархія класів

2.3 Опис програми

Структура проекту наведена на рис. 2.2.

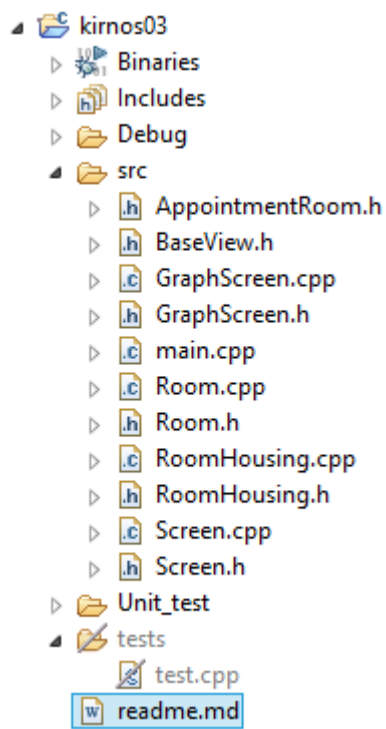


Рисунок 2.2 – Структура проекту

В програмі було створено три класи:

- клас **Room**, який містить значення висоти, ширини і довжини кімнати;

- класи **Screen** та **GraphScreen** для виведення полів об'єкту типу **Room**. Призначення спроектованих класів наведено на рис. 2.3.

C BaseView	Базовый класс, задает способ вывода объекта класса Room и его наследников
C GraphScreen	Класс предназначен для вывода объекта класса Screen используя псевдографику
C Room	Класс описания размеров комнат 3 этажа ВК корпусу
C RoomHousing	Класс расширяющий комнату к комната в корпусе
C Screen	Класс предназначен для вывода объекта класса Screen

Рисунок 2.3 – Призначення класів

2.4 Важливі фрагменти програми

Демонстрація роботи (функція **main()**). Використання нащадків відбувається через покажчик на базовий улас.

```

1 int main() {
2     Room **r = new Room*[4];
3     r[0] = new RoomHousing(4,8,8,"k 202", 28, PRACTICAL);
4     r[1] = new RoomHousing(4,12,10,"k 313", 25, LABORATORY);
5     r[2] = new Room(4,10,10);
6     r[3] = new Room(4,8,10);
7
8     BaseView **bw = new BaseView*[4];
9     bw[0] = new GraphScreen(*r[0]);
10    bw[1] = new Screen(*r[1]);
11    bw[2] = new GraphScreen(*r[2]);
12    bw[3] = new Screen(*r[3]);
13
14    for(int i = 0; i < 4; i++)
15        bw[i]->Display();
16
17    for(int i = 0; i < 4; i++){
18        delete r[i];
19        delete bw[i];
20    }
21    delete []r;
22    delete []bw;
23
24    getchar();
25    return 0;
26 }
```

Динамічне визначення типу:

```

1 const RoomHousing *temp = dynamic_cast<const RoomHousing*>(room);
2 if (temp != NULL) {
3     ...
4 }
```

Результати роботи

Результати роботи показано на рис. 3.1.

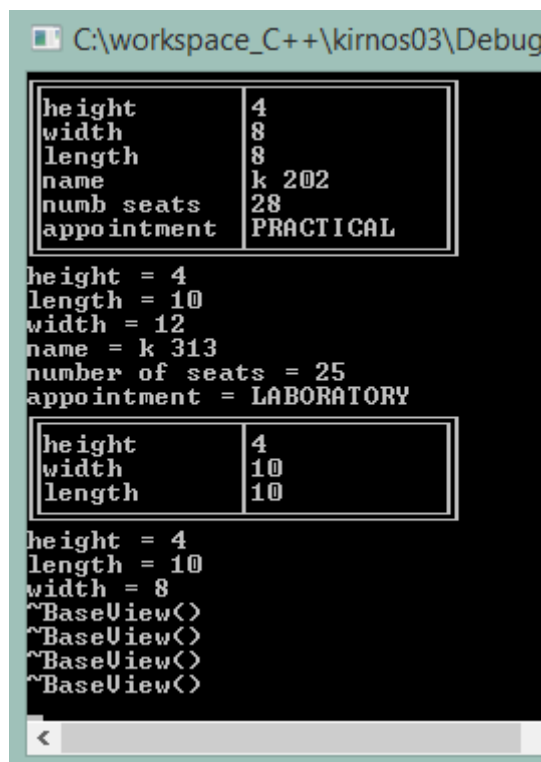


Рисунок 3.1 – Результати роботи

Результат виконання всіх модульних тестів зображений на рисунку 3.2.

```

Running main() from gtest_main.cc
[=====] Running 3 tests from 2 test cases.
[-----] Global test environment set-up.
[-----] 2 tests from classRoom
[ RUN      ] classRoom.constructorsGetters
[          OK ] classRoom.constructorsGetters (0 ms)
[ RUN      ] classRoom.Volume
[          OK ] classRoom.Volume (0 ms)
[-----] 2 tests from classRoom (0 ms total)

[-----] 1 test from classRoomHousing
[ RUN      ] classRoomHousing.constructorsGetters
[          OK ] classRoomHousing.constructorsGetters (0 ms)
[-----] 1 test from classRoomHousing (50 ms total)

[-----] Global test environment tear-down
[=====] 3 tests from 2 test cases ran. (50 ms total)
[ PASSED  ] 3 tests.

```

Рисунок 3.2 – Результати тестування

Висновок

В ході виконання лабораторної роботи були отримані навички розробки власних ієрархій класів із використанням принципу розширення та віртуальності.