

Тема 2: Права доступу, покажчики, const, посилання

Мета

Отримати навички при передаванні об'єктів у класи із застосуванням прав доступу та const-модифікаторів.

1 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Розподілити в **Room** права доступу private, public. Реалізувати клас **GraphScreen** основна задача якого полягає у більш багатому відображенні даних Window із застосуванням псевдографіки для наочного відображення пов'язаного об'єкта. **GraphScreen** повинен містити поля згідно опису в індивідуальному завданні та наступні методи:

- SetDataSource() - для зміни об'єкта-джерела даних.
- PrintData() - виводитиме інформацію про отриманий об'єкт у якості аргументу. Оновити **Screen** для збереження функціональності цього класу при роботі з оновленим Window.

2 РОЗРОБКА ПРОГРАМИ

2.1 Засоби ООП

В ході розробки програми були використані такі засоби ООП:

- Абстракція.
- Інкапсуляція.

2.2 Ієрархія та структура класів

Ієрархія класів наведена на рис. 2.1

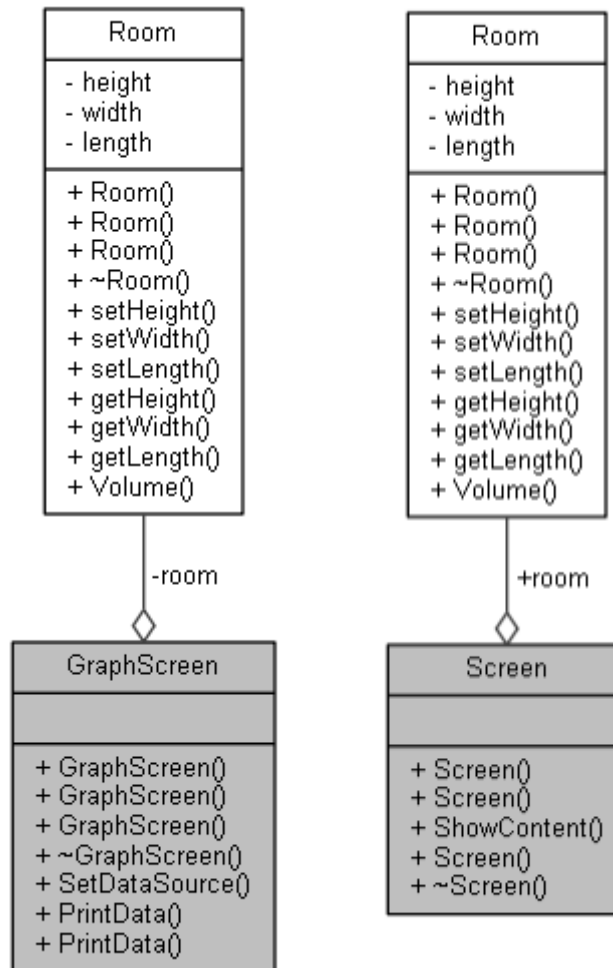


Рисунок 2.1 – Ієрархія класів

2.3 Опис програми

Структура проекту наведена на рис. 2.2.

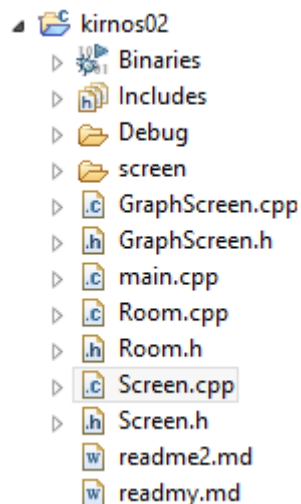


Рисунок 2.2 – Структура проекту

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.




 GraphScreen	Клас призначений для вивода об'єкта класу Screen використовуючи псевдографіку
 Room	Клас описує розміри кімнат 3-го поверху ВК корпусу
 Screen	Клас призначений для вивода об'єкта класу Screen

Рисунок 2.3 – Призначення класів

2.4 Важливі фрагменти програми

Демонстрація роботи (функція **main()**)

```

1  int main() {
2      cout << "Begin." << endl;
3      Room r(10, 11, 12);
4      GraphScreen gs(r);
5      gs.PrintData();
6      Room r2(1, 2, 3);
7      gs.PrintData(r2);
8      cout << "done." << endl;
9      system("pause");
10     return 0;
11 }
```

Функція **PrintData** класу **GraphScreen** для виведення полів змінної типу **Room**

```

1  void GraphScreen::PrintData(const Room& R) const{
2      char lh = 201, rh = 187, h = 205, hd = 209, l = 186, r = 186, c = 179, ld =
3          200, rd = 188, dh = 207;
4      cout << lh;
5      for (int i = 0; i < 10; i++, cout << h)
6          ;
7      cout << hd;
8      for (int i = 0; i < 10; i++, cout << h)
9          ;
10     cout << rh << endl;
11     cout << l << setw(10) << left << "height" << c << setw(10) << R.getHeight()
12         << r << endl;
13     cout << l << setw(10) << left << "width" << c << setw(10) << R.getWidth()
14         << r << endl;
15     cout << l << setw(10) << left << "length" << c << setw(10) << R.getLength()
16         << r << endl;
17     cout << ld;
18     for (int i = 0; i < 10; i++, cout << h)
19         ;
20     cout << dh;
21     for (int i = 0; i < 10; i++, cout << h)
22         ;
23     cout << rd << endl;
24 }
```

Результати роботи

Результати роботи показано на рис. 3.1.

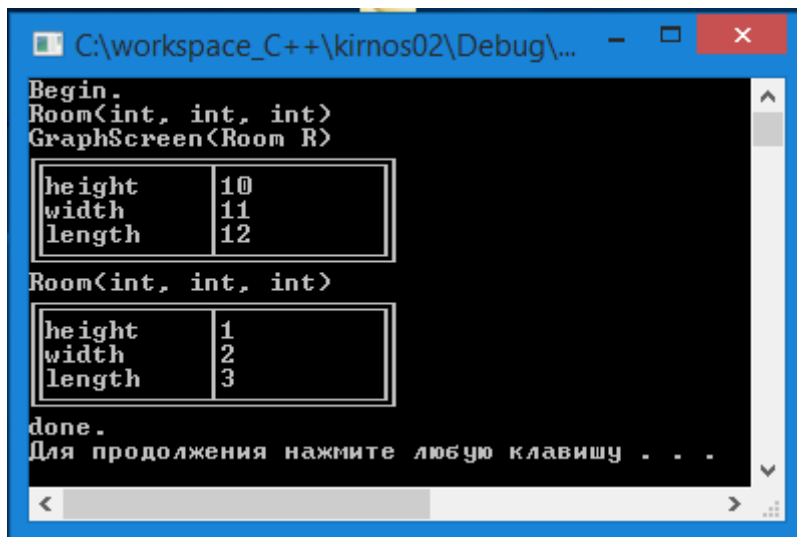


Рисунок 3.1 – Результати роботи

Висновок

В ході виконання лабораторної роботи було створено клас для відображення даних за допомогою псевдографіки із застосуванням прав доступу та const-модифікаторів при передачі об'єктів у клас.