

Лабораторна робота № 2: Права доступу, const, покажчики посилання

Мета

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

1.ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Розподілити в [Wheel](#) права доступу private, public. Реалізувати клас [GraphScreen](#) основна задача якого полягає у більш багатому відображенню даних [Wheel](#) 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 [Wheel](#).

2.РОЗРОБКА ПРОГРАМИ

2.1 Засоби ООП

В ході розробки програми були використані такі засоби ООП:

- Абстракція - кожен об'єкт описує свою особливу сутність, яка визначається його полями.
- Інкапсуляція - поля об'єктів закриті для користувача, натомість ми даємо доступ до даних за допомогою геттерів та сеттерів, так користувач має можливість отримати готові дані, а не обробляти їх, для подальшого використання.

2.2 Ієрархія та структура класів

На рис 2.1 дивись ієрархію класів

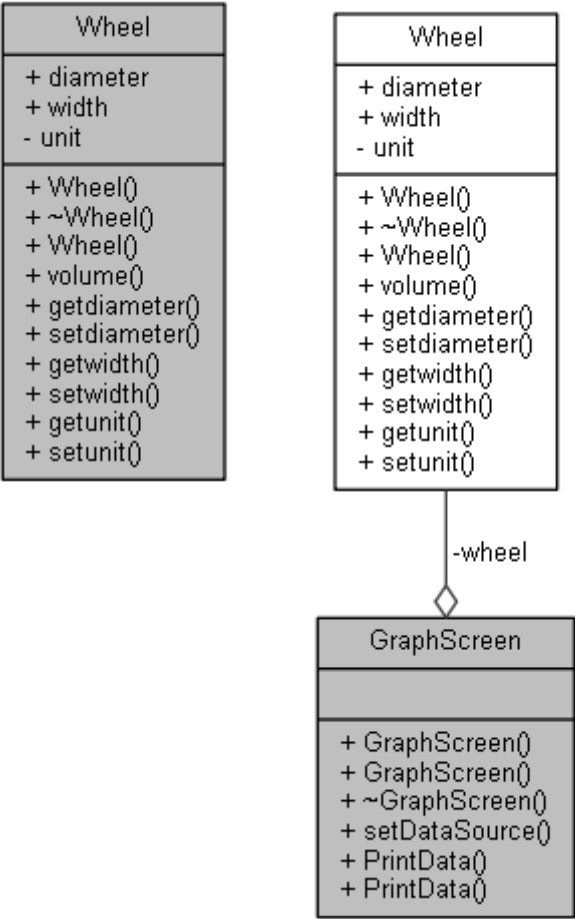


Рисунок 2.1 ієрархія класів

2.3 Опис програми

На рис 2.2 дивись структуру проекту.

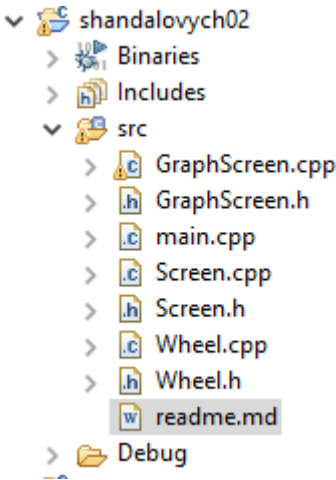


Рисунок 2.2 структура проекту

На рис 2.3 дивись призначення класів.

Here are the classes, structs, unions and interfaces with brief descriptions:

C GraphScreen	Класс который выводит псевдографикой колесо
C Screen	Класс, который выводит информацию про колесо
C Wheel	Класс который содержит описание колеса

Рисунок 2.3 призначення класів

З попередньої лабораторної роботи збереглися класи **Wheel1** та **Screen**. У **Wheel1** змінено модифікатори доступу, додано 2 методи, які виводять ширину та висоту вікна. Методи виводу класу **Screen** змінено, для коректної роботи з `private` та `public` полями. Додано клас **GraphScreen** у якому завдяки псевдографіки ми виводимо наш **Wheel1**.

2.4 Важливі фрагменти програми

У программі слід зауважити увагу на таких моментах:

Клас **GraphScreen** функція виведення даних за допомогою псевдографіки:

```
void GraphScreen::PrintData(const Wheel& wheel) {
cout << "Объем колеса: " << wheel.volume() << "\n";
    cout << "          * * * \n";
    cout << "          * *   * * \n";
    cout << "          * *   * * \n";
    cout << "      *   *   *   *   * \n";
    cout << "    *   *   *   *   *   * \n";
    cout << "    *       * * *   *   * \n";
    cout << " *           * * * * *   * \n";
    cout << "***** \n";
    cout << " *           * * *   *   * \n";
    cout << "    *       *   *   *   *   * \n";
    cout << "      *   *   *   *   *   * \n";
    cout << "          * *   *   *   * \n";
    cout << "          * *   *   *   * \n";
    cout << "          * * * \n";
}
```

Демонстрація роботи програми:

```
int main(void) {  
    //Константы для инициализации первого объекта.  
    const int diameter = 20;  
    const int width = 10;  
    const string unit = "см";  
  
    Wheel data( diameter, width, unit);  
  
    Screen screen(data);  
  
    screen.ViewWheel();  
  
    GraphScreen view2(&data);  
    view2.PrintData();  
  
    return 0;  
}
```

Результат роботи показано на рис 3.1.

```

      * * *
    * *   * * *
  * *     * * *
*         * * *
*       * * * * *
*     * * * * *
*   * * * * *
*****
*   * * * * *
*     * * * * *
*       * * * * *
*         * * * *
          * * *
            * *
              *

```

Вызов деструктора
Вызов деструктора

Рисунок 3.1 результат роботи

В результаті лабораторної роботи було розроблено програму з використанням прав доступу та const методів. Були придбані навички роботи з цими технологіями.