Спадкування та віртуальність

Мета

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

1. Індивідуальне завдання

Створити клас Mouse використовуючи спадкування від Manipulator з додаванням нових полів та класс відображення даних - MouseScreen.

Виділити базовий клас BaseView для Screen та GraphScreen із функцією відображення та віртуальними методами ShowHeader(), ShowContent(),ShowFooter(). Перенести основний функціонал у базовий клас, реалізувавши спецефічну поведінку у відповідних віртуальних методах. Створити клас MouseScreen та вибрати необхідне місце у ієрархії відображень для цього класу. Показати роботу віртуальності на прикладі викоритсання нащадка через покажчик на базовий клас для об'екту Manipulator та Mouse.

2. Розробка програми

2.1 Засоби ООП

В ході розробки програми були використані так засоби ООП:

- Абсракція – кожен об’єкт описує свою сутність, яка визначається його полями.

- Спадкування - механізм утворення нових класів на основі використання вже існуючих

- 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 подпльшого вікористання.

2.2 Ієрархія та структура класів

На рис 2.1 дивись ієрархію класів.

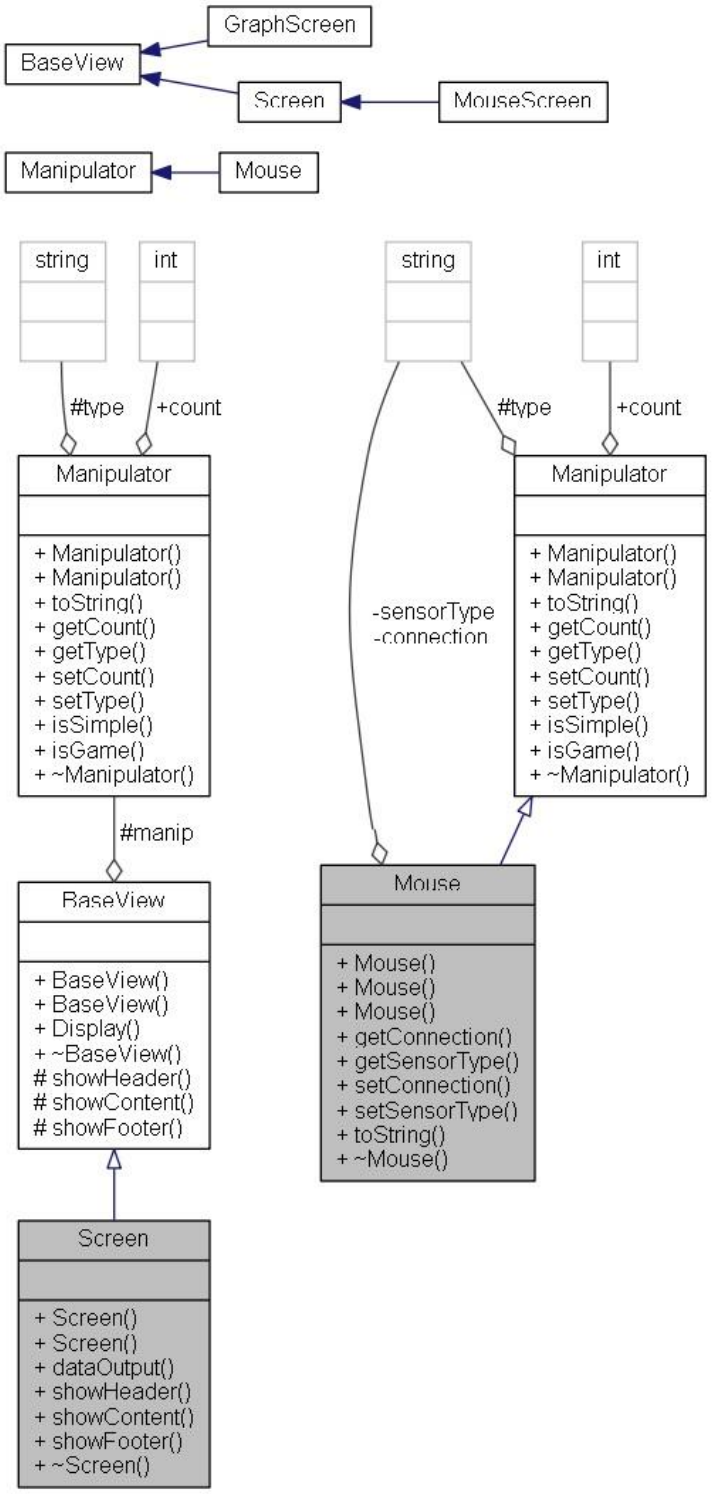


Рисунок 2.1 – Ієрархія класів

Опис програми

На рис. 2.2 дивись структуру проекту.

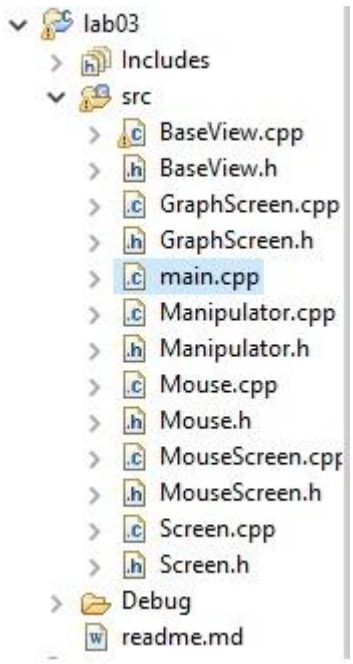


Рисунок 2.2 – Структура проекту

На рис. 2.3 дивись призначення класів.

Класи, структури, об'єднання та інтерфейси з коротким описом.

BaseView	Базовий клас для **Manipulator** та **Mouse** із функцією відображення
GraphScreen	Клас для відображення псевдографіки
Manipulator	Клас опису маніпулятору
Mouse	Клас опису мишки
MouseScreen	Клас відображення інформації **Mouse** та **Manipulator**
Screen	Класс для відображення даних

Рисунок 2.3 – Призначення класів

Маємо 2 класи. Маніпулятор Manipulator та відображувач Screen. Маніпулятор(Manipulator) описує сутність програмного маніпулятору, тобто його тип та кількість кнопок. Відображувач(Screen) використовується для виводу даних з об’єкту Manipulator у консоль. У функції main() відображена робота програми.

2.4 Важливі фрагменті програми

У программі слід зауважити увагу на таких моментах:

Клас `BaseView.h` функція відображення та віртуальні методи:

```
protected:
        Manipulator *manip;
        virtual void showHeader() = 0;
        virtual void showContent() = 0;
        virtual void showFooter() = 0;
public:
        BaseView();
        BaseView(Manipulator *manip);
        void Display();
        virtual ~BaseView();
```

Демонстрація роботи програми:

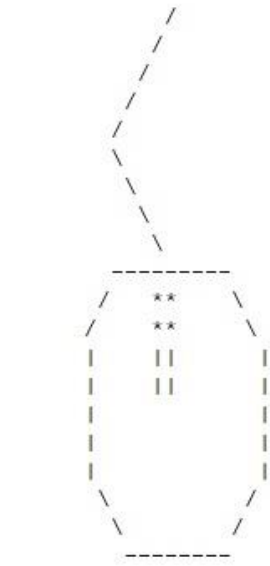
```
        int main() {

                const int COUNT_MANIP1 = 5;
                        const string TYPE_MANIP1 = "Joystick";
                        Manipulator manip1(COUNT_MANIP1, TYPE_MANIP1);

                        const int COUNT_MANIP2 = 2;
                        const string TYPE_MANIP2 = "Mouse";
                        Manipulator manip2(COUNT_MANIP2, TYPE_MANIP2);

                        const int COUNT_MANIP3 = 4;
                        const string TYPE_MANIP3 = "Mouse";
                        const string CONNECTION = "Wireless";
```

```
const string SENSOR_TYPE = "Laser";
Mouse mouse(COUNT_MANIP3, TYPE_MANIP3, CONNECTION, SENSOR_TYPE);

cout << "From Screen " << endl;

cout << "manip1: \n";
Screen view(&manip1);
view.Display();

cout << "manip2: \n";
Screen view2(&manip2);
view2.Display();

GraphScreen gscreen;
cout << "From Graph Screen \n";
gscreen.setDataSource(&manip1);
gscreen.Display();

MouseScreen mscreen(&mouse);
cout << "From mouse: \n";
mscreen.Display();

return 0;
}
```

3. Результати роботи

Результати роботи показано на рис.3.1

```
From Screen
manip1:

Screen constructor
Screen::showHeader()
Количество кнопок: 5
Тип устройства : Joystick

 Screen::showFooter()
manip2:

Screen constructor
Screen::showHeader()
Количество кнопок: 2
Тип устройства : Mouse

 Screen::showFooter()

GraphScreen constructor
From Graph Screen
GraphScreen::showHeader()
```



```
 Тип девайса: Joystick
Количество кнопок: 5


 GraphScreen::showFooter()

Screen constructor
From mouse:
MouseScreen::showHeader()
Количество кнопок: 4
Тип устройства : Mouse
Тип соединения: Wireless
Тип сенсора: Laser

Количество кнопок: 4
Тип устройства : Mouse
Тип соединения: Wireless
Тип сенсора: Laser

 MouseScreen::showFooter())

Screen destructor
GraphScreen destructor

Screen destructor

Screen destructor
```

Рисунок 3.1 – Результати роботи

Висновки

Отримав основні навики розробки власних ієрархій класів із використанням принципу розширення та віртуальності.