

Тема 2: Права доступу, const, покажчики, посилання.

Мета

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

1. Загальне завдання

Розподілити в Data1 права доступу private, public.

Реалізувати клас View2 основна задача котрого полягає у більш багатому відображенню даних Data1 із застосуванням псевдографіки для наочного відображення пов'язаного об'єкта. View2 повинен містити поля згідно опису в індивідуальному завданні та наступні обов'язкові методи:

- *SetDataSource()* - для зміни об'єкта джереладаних. Сигнатура:
`void SetDataSource(const CMyData* aData);`
- *PrintData()* - виводитиме на екран інформацію про отриманий об'єкт у якості аргументу. Сигнатура:
`void View2::PrintData(const CMyData* aData);`
Оновити View1 для збереження функціональності цього класу при роботі із оновленням Data1.
Згідно індивідуального завдання реалізувати нову пограму, базуючись на класах №1, та вибравши своє завдання за варіантами.

1. Назва Data1: **Manipulator**;
2. Поля Data1:
 - private: amountOfButtons;
 - public typeOfManipulator;
3. View2 - GraphScreen;
4. const-Methods - Data1::isSimple() const , Data1::IsGame() const

2. Розробка програми

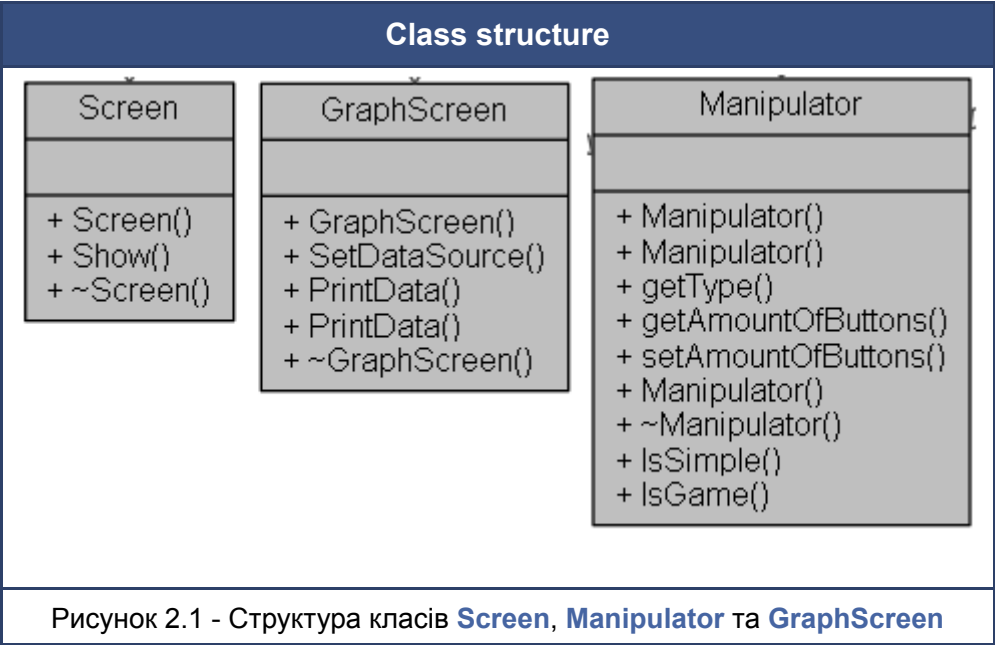
2.1. Засоби ООП

У розробленій програмі були використані наступні засоби ООП:

- інкапсуляція;

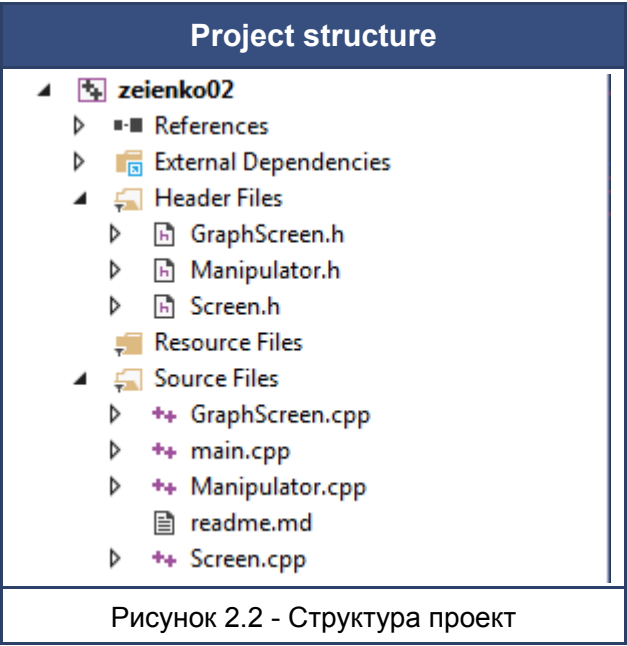
2.2. Ієрархія та структура класів

На рис. 2.1 наведена структура класів **Screen**, **Manipulator** та **GraphScreen**.



2.3. Опис програми

На рис. 2.2 наведена структура розробленого проекту:



2.4. Важливі фрагменти програми

У розробленій програмі слід зауважити увагу на таких класах: Manipulator, Screen, GraphScreen. У класі GraphScreen реалізована псевдографіка, тому роботу цього класу і було перевірено у функції launch() (див. рис. 2.3).

Function launch()
img_launch_func
Рисунок 2.3. – Фрагмент функції launch()

Призначення спроектованих класів наведено на рис. 2.4.




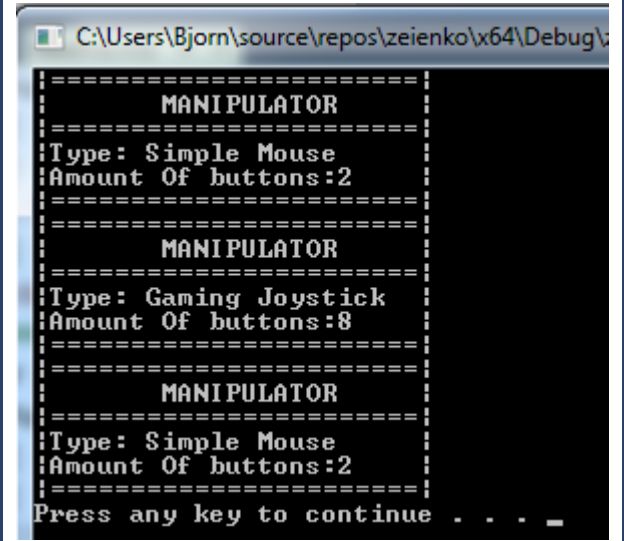
Predestination structure	
Класи	
Класи, структури, об'єднання та інтерфейси з коротким описом.	
 GraphScreen	Representation of text-pseudo-graphic display of data
 Manipulator	Represents abstraction of mouse periphery
 Screen	This class provides an output information about manipulator

Рисунок 2.4. – Призначення спроектованих класів

3. Результат работы

Результат роботи програми зображений на рисунку 3.1.

Output to the console

Рисунок 3.1. – Приклад роботи програми

Висновок

В ході виконання лабораторної роботи ми отримали навички при передаванні об'єктів у класи із застосуванням прав доступу та const-модифікаторів.