Тема 5: Абстрктні класт, інтерфейси, серіалізація.

Мета

Навчитись застосовувати інтерфейси для роботи класів на прикладі задачі серіалізації.

1. Загальне завдання

Реалізувати для кожного із класів даних своєї ієрархії можливість збереження та завантаження даних за допомогою класу FileStorage, який видається до лабораторної роботи у вигляді бібліотеки. Показати у звіті бінарний дамп збереженого файлу та відмітити дані із власних об'єктів.

2. Розробка пограми

2.1. Засоби ООП

У розробленій програмі були використані наступні засоби ООП:

- інкапсуляція
- спадкування
- поліморфізм

2.2. Ієрархія та структура класів

На рис. 2.1 наведена ієрархія зв'язків класу BaseView, на рис. 2.2 - **Manipulator**.

Hierarchy of BaseView class

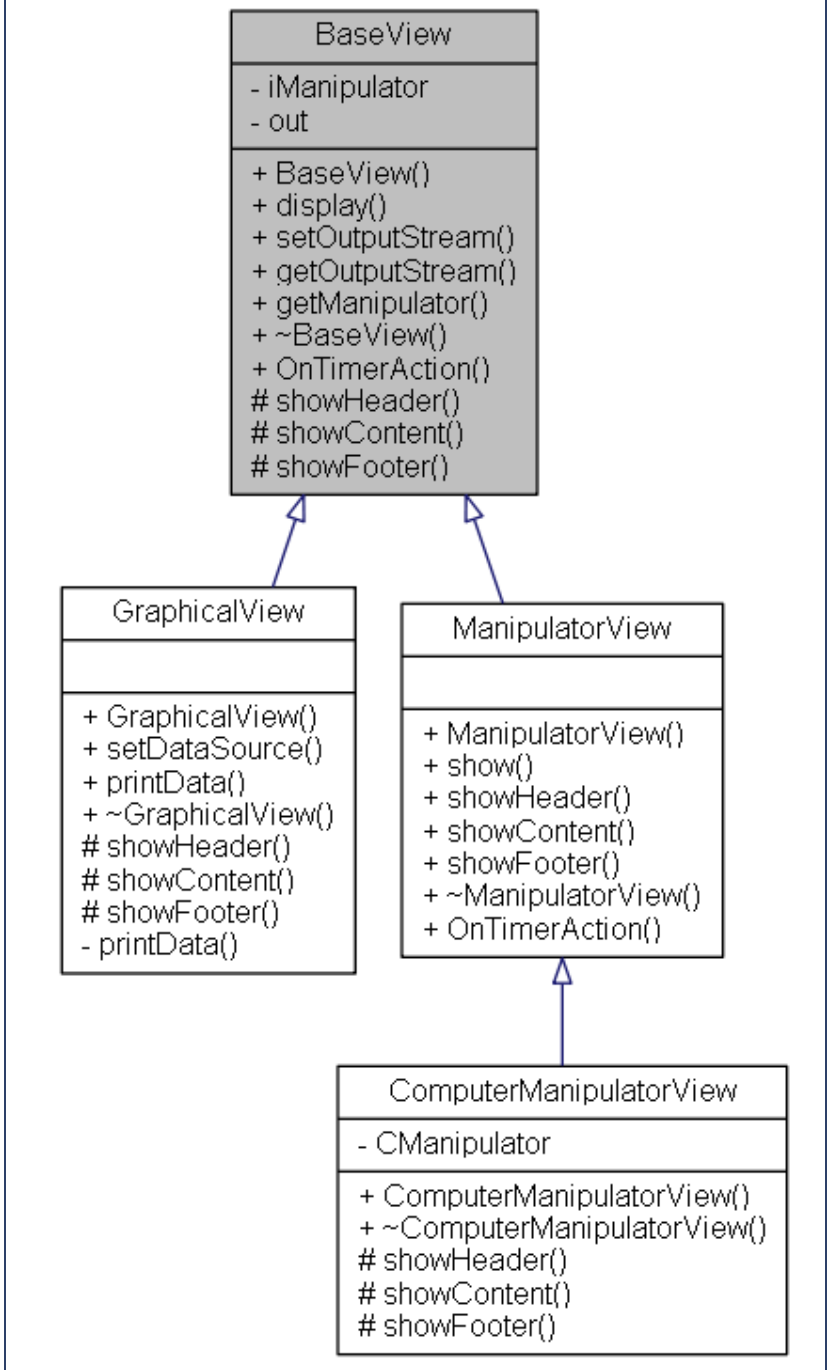

Рисунок 2.1 - Ієрархія зв'язків класу BaseView

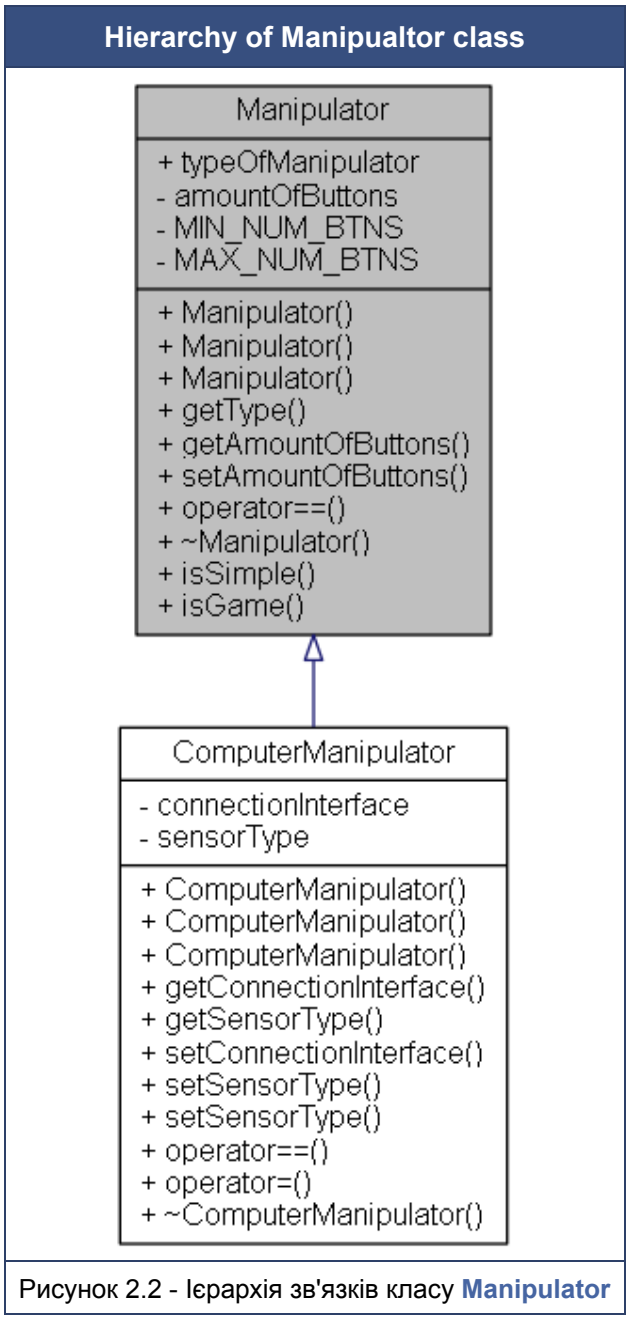

Рисунок 2.2 - Ієрархія зв'язків класу **Manipulator**

2.3. Опис програми

На рисунку 2.4 наведена структура розробленого проекту:

Project structure

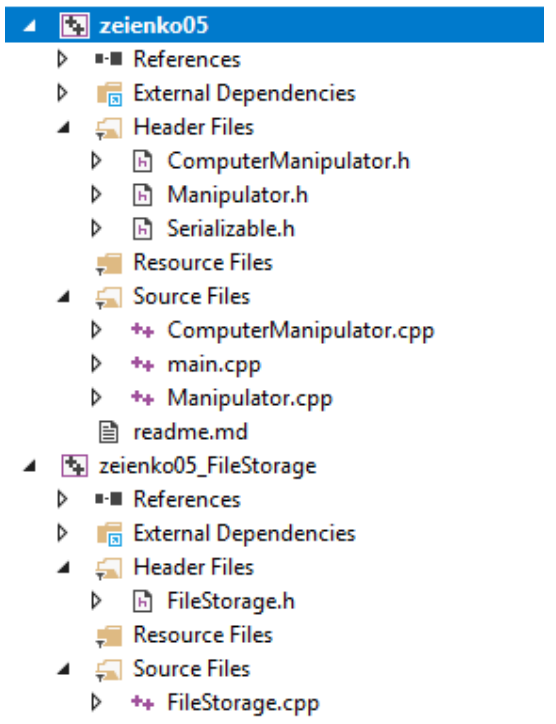

Рисунок 2.4 - Структура проект

2.4. Важливі фрагменти програми

Розроблений інтерфейс **Serializable** зображений на рисунку 2.5. Фрагементи функцій збереження та відновлення класу **Manipulator** зображені на відповідних рисунках 2.5 та 2.6.

Функція збереження об'єкта

```
88    void Manipulator::toSave(std::ofstream& os) {
89        os << "Type: " << this->getType() << std::endl;
90        os << "Amount_of_buttons: " << this->getAmountOfButtons() << std::endl;
91    }
```

Рисунок 2.5. – Функція toSave()

Функція відновлення об'єкта

```
93  void Manipulator::toLoad(std::ifstream& in) {
94      string line;
95      string label;
96      while (!in.eof()) {
97          getline(in, line);
98          std::stringstream ss(line);
99          ss >> label;
100         if (label.compare("Type:") == 0) {
101             char* restoreType = new char[12];
102             ss >> restoreType;
103             this->setType(restoreType);
104             delete restoreType;
105         }
106         else if (label.compare("Amount_of_buttons:") == 0) {
107             unsigned restoreAmountOfButtons;
108             ss >> restoreAmountOfButtons;
109             this->setAmountOfButtons(restoreAmountOfButtons);
110         }
111     }
112 }
```

Рисунок 2.6. – Функція toLoad()

Призначення спроектованих класів наведено на рис. 2.7. На цьому рисунку також зображено клас FileStorage, але файли цього класу знаходяться в іншому проекті.

Predestination structure

Класи

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

C **ComputerManipulator**	This class represents computer`s manipulators
C **FileStorage**	This class represent file storage
C **Manipulator**	Represents abstraction of mouse periphery
C **Serializable**	This interace is used to safe and restore data of an object that implements it

Рисунок 2.7. – Призначення спроектованих класів

3. Результат работи

Результат роботи програми зображений на рисунку 3.1.

Output to the console

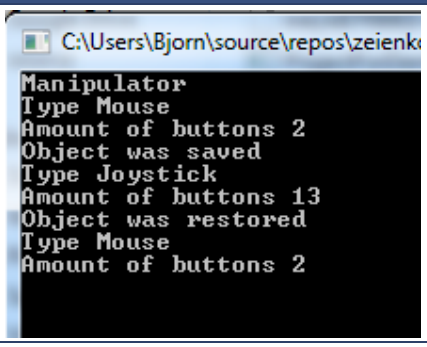

Рисунок 3.1. – Результат виконання програми

Дані, котрі збережені у файлі зображені на риснку 3.2.

Дані з файлу

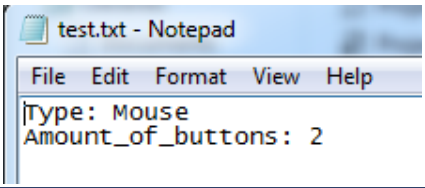

Рисунок 3.2. – Результат тестування

Висновок

В ході виконання лабораторної роботи були отримані навики застосовування інтерфейсів для роботи класів на прикладі задачі серіалізації.