

СПАДКУВАННЯ ТА ВІРТУАЛЬНІСТЬ

Лабораторна робота №3

Мета:

- отримати навички розробки власних ієрархій класів із використанням принципу розширення та віртуальності.

1 ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

Варіант 10. Створити клас «Банка» використовуючи спадкування від класу «Ємності». Виділити базовий клас BaseView для Screen та GraphScreen із функцією BaseView::Display() та наступними віртуальними функціями, котрі викликаються із неї: showHeader(), showContent(), showFooter(). Створити клас, котрий задає специфіку відображення для об'єктів класу-нащадку. Обрати необхідне місце у ієрархії для нього.

2 РОЗРОБКА ПРОГРАМИ

Для реалізації програми було створено клас Jar, клас для його відображення JarScreen та базовий клас відображення BaseView та оновлено існуючі класи та методи, згідно індивідуального завдання.

2.1 Засоби ООП

У розробленій програмі використані наступні засоби ООП:

- розділення програми на ієрархію класів (інкапсуляція);
- поліморфізм;
- спадкування;
- абстракція (віртуальність);

2.2 Ієрархія та структура класів

На рис.2.2 наведена ієрархія розроблених класів

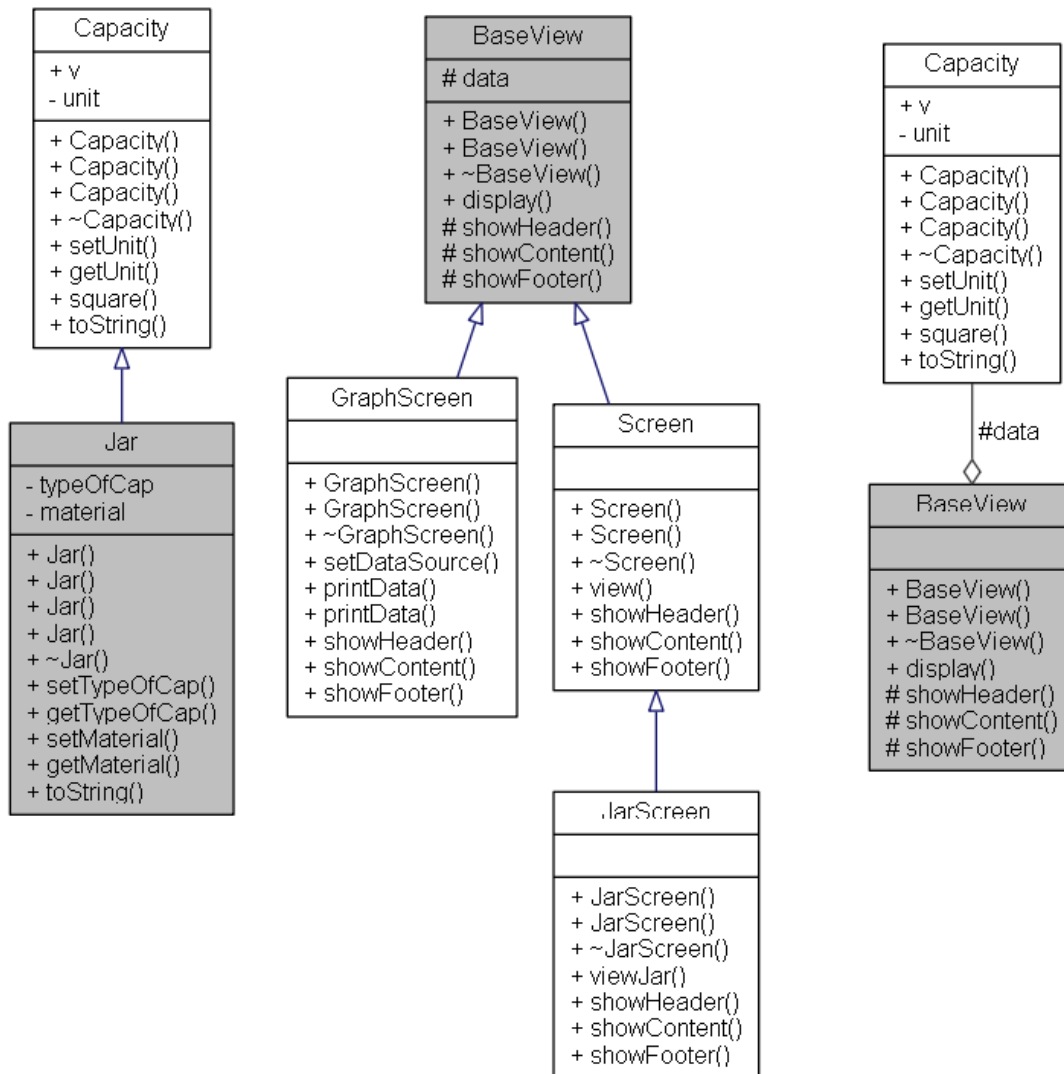


Рисунок 2.2 – Ієрархія класів

2.3 Опис програми

На рис.2.3 наведена структура розробленого проекту

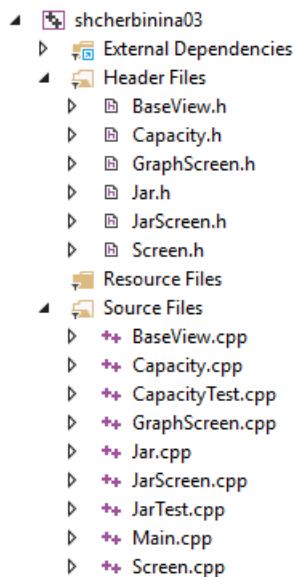


Рисунок 2.3 – Структура проекту

Призначення спроектованих класів наведено на рис.2.4

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

C BaseView	Базовий клас відображення даних
C Capacity	Клас, що містить реалізацію ємності
C GraphScreen	Клас відображення інформації про об'єкт класу Capacity за допомогою псевдографіки
C Jar	Клас, що містить реалізацію банки
C JarScreen	Клас відображення інформації про об'єкт класу Jar
C Screen	Клас відображення інформації про об'єкт класу Capacity

Рисунок 2.4 – Призначення класів

2.4 Важливі фрагменти програми

Клас BaseView:

```
/**
 * Базовий клас відображення даних
 */
class BaseView
{
protected:
    ///Об'єкт відображення
    Capacity* data;
    /**
     * Виводить заголовок відображення
     */
    virtual void showHeader() = 0;
    /**
     * Виводить головну інформацію про об'єкт
     */
    virtual void showContent() = 0;
    /**
     * Виводить нижній колонтитул відображення
     */
}
```

```

        virtual void showFooter() = 0;
public:
    /**
     * Конструктор без параметрів
     */
    BaseView();
    /**
     * Конструктор з параметрами
     * @param встановлює значення об'єкта, що виводиться
     */
    BaseView(Capacity* data);
    /**
     * Деструктор
     */
    virtual ~BaseView();
    /**
     * Виводить об'єкт
     */
    void display();
};

```

Клас Jar

```

/**
 * Клас, що містить реалізацію банки
 */
class Jar :
    public Capacity
{
private:
    ///Тип кришки
    string typeOfCap;
    ///Матеріал
    string material;
public:
    /**
     * Конструктор без параметрів
     */
    Jar();
    /**
     * Конструктор з параметрами для власних полів
     */
    Jar(string typeOfCap, string material);
    /**
     * Конструктор копіювання
     */
    Jar(const Jar& jar);
    /**
     * Конструктор з параметрами для власних полів і тих, що унаслідувались
     */
    Jar(units unit, float v, string typeOfCap, string material);
    /**
     * Деструктор
     */
    virtual ~Jar();
    /**
     * Сеттер для типу кришки
     * @param встановлює значення типу кришки
     */
    void setTypeOfCap(string typeOfCap);
    /**
     * Геттер для типу кришки
     * @return тип кришки
     */
    string getTypeOfCap();
};

```

```

/**
 * Сеттер для поля матеріалу
 * @param встановлює значення матеріалу
 */
void setMaterial(string material);
/**
 * Геттер для поля матеріалу
 * @return матеріал
 */
string getMaterial();
/**
 *
 */
virtual string toString();

```

Функція main():

```

/**
 * Точка входу в програму
 */
int main(int argc, char **argv) {
    setlocale(LC_ALL, "Russian");

    Jar jar1("Закручується", "Скло");
    JarScreen view1(&jar1);
    view1.display();

    Jar jar2;
    Screen view2(&jar2);
    view2.display();

    Capacity data(1,1000);
    GraphScreen view3(&data);
    view3.display();

    ::testing::InitGoogleTest(&argc, argv);
    const int res = RUN_ALL_TESTS();
    getch();
    return res;
}

```

3 РЕЗУЛЬТАТИ РОБОТИ

