**Лабораторна робота №7. Поліморфізм**

Тема: Класи. Поліморфізм. Абстрактні класи.

Мета: Отримати знання про парадигму ООП – поліморфізм. Навчитися застосовувати отримані знання на практиці.

**ВИМОГИ**

**1.1 Інформація про розробника:**

* Кліщов Б. Р.
* КІТ 102.8а
  1. **Загальне завдання**

Модернізувати попередню лабораторну роботу шляхом:

- додавання ще одного класу-спадкоємця до базового класу. Поля обрати самостійно;

- базовий клас зробити абстрактним. Додати абстрактні поля;

- розроблені класи-списки поєднуються до одного таким чином, щоб він міг працювати як з базовим класом, так і з його спадкоємцями. При цьому, серед полів класу-списку повен бути лише один масив, що містить усі типи класів ієрархії. Оновити методи, що працюють з цим масивом.

**1.3 Додаткові умови виконання завдання:**

- продемонструвати відсутність витоків пам’яті;

- продемонструвати роботу розроблений методів за допомогою модульних тестів;

- не використовувати конструкцію «using namespace std;» , замість цього слід роботи «using» кожного необхідного класу:using std::string, using std::cout;

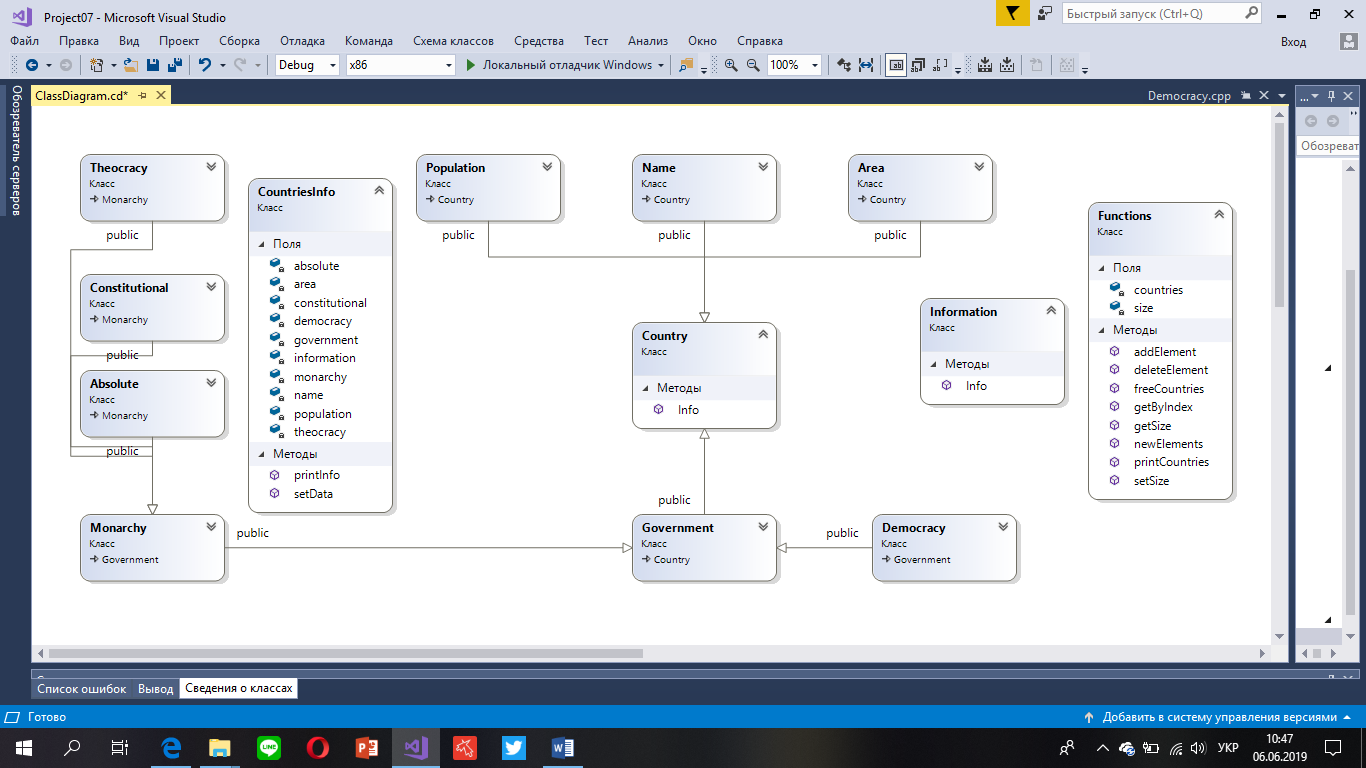
- в проекті не повинні використовуватися бібліотеки введення / виведення мови С, а також не повинні використовуватися рядки типу char\*.

1. **ОПИС ПРОГРАМИ**

**2.1 Функціональне призначення**

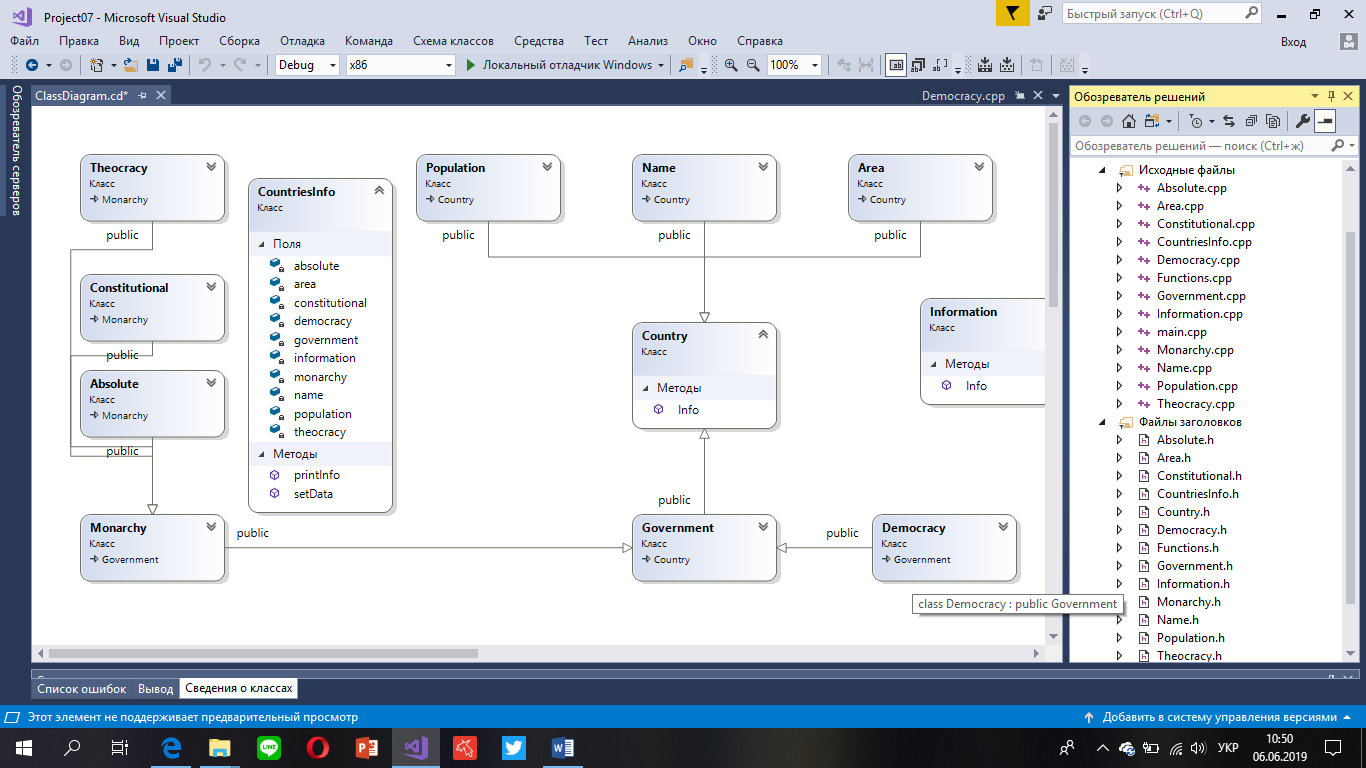
Програма призначена щоб отримувати та зберігати інформацію щодо різних країн світу.

* 1. **Опис логічної структури**

На рисунку № 1 зображена діаграма класу

Малюнок №1. Діаграма класу

На рисунку № 2 зоображена структура програми:



Малюнок №2: Cтруктура програми

**2.3 Важливі фрагменти програми**

Код програми:

#include "Functions.h"

int main()

{

srand(time(NULL));

string \*gover = new string[2];

gover[0] = "Monarchy";

gover[1] = "Democracy";

string \*monarchy = new string[3];

monarchy[0] = "Absolute";

monarchy[1] = "Constitutional";

monarchy[2] = "Theocracy";

Functions functions;

int size;

cout << "Please, enter the size: ";

cin >> size;

cout << endl;

functions.setSize(size);

functions.newElements(gover, monarchy);

functions.addElement(gover, monarchy);

int index;

cout << "Please, enter the index for deleting: ";

cin >> index;

cout << endl;

functions.deleteElement(index);

functions.printCountries();

cout << "Please, enter the index for geting country: ";

cin >> index;

cout << endl;

functions.getByIndex(index);

functions.freeCountries();

delete[]gover;

delete[]monarchy;

\_CrtSetReportMode(\_CRT\_WARN, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_WARN, \_CRTDBG\_FILE\_STDERR);

\_CrtSetReportMode(\_CRT\_ERROR, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ERROR, \_CRTDBG\_FILE\_STDERR);

\_CrtSetReportMode(\_CRT\_ASSERT, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ASSERT, \_CRTDBG\_FILE\_STDERR);

\_CrtDumpMemoryLeaks();

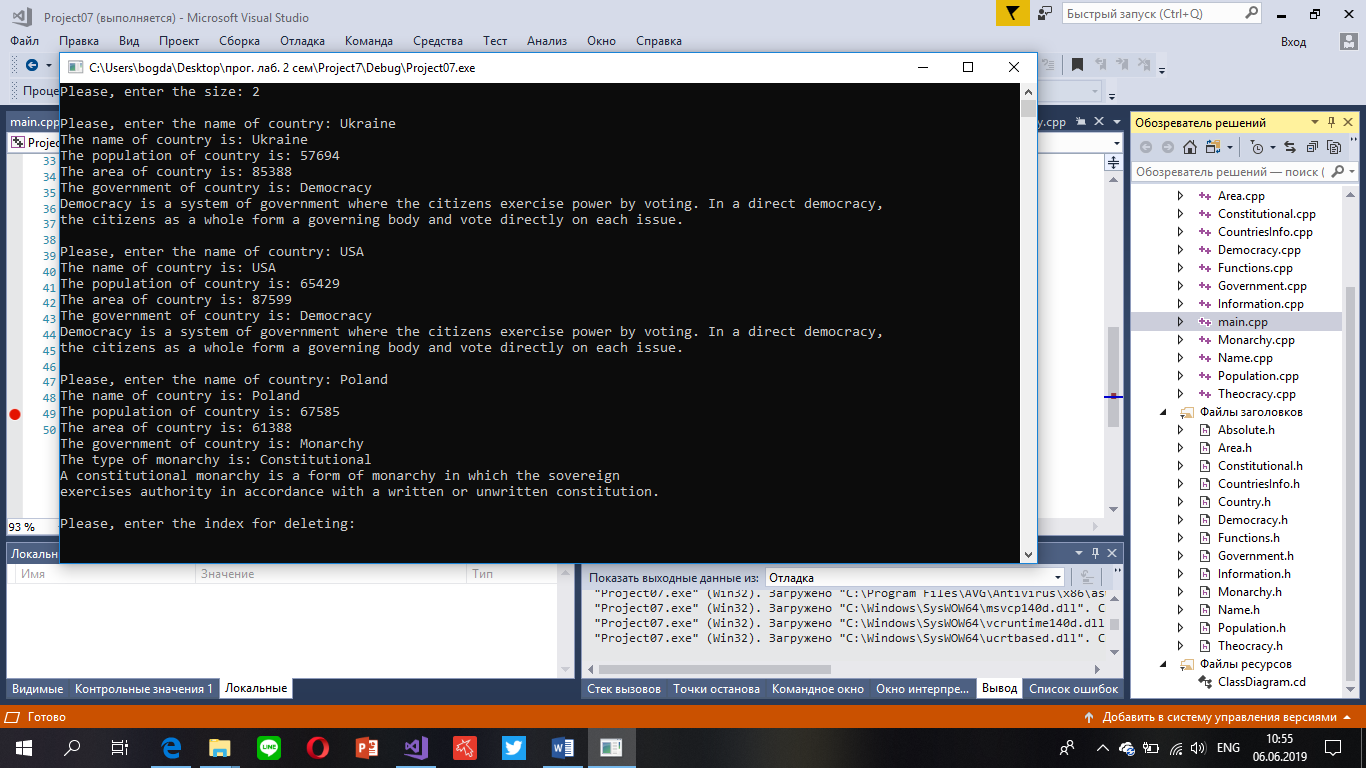
return 0;

}

1. **ВАРІАНТИ ВИКОРИСТАННЯ**

**3.1 Результат роботи функцій**

На рисунку № 3 зоображено результат робрти програми



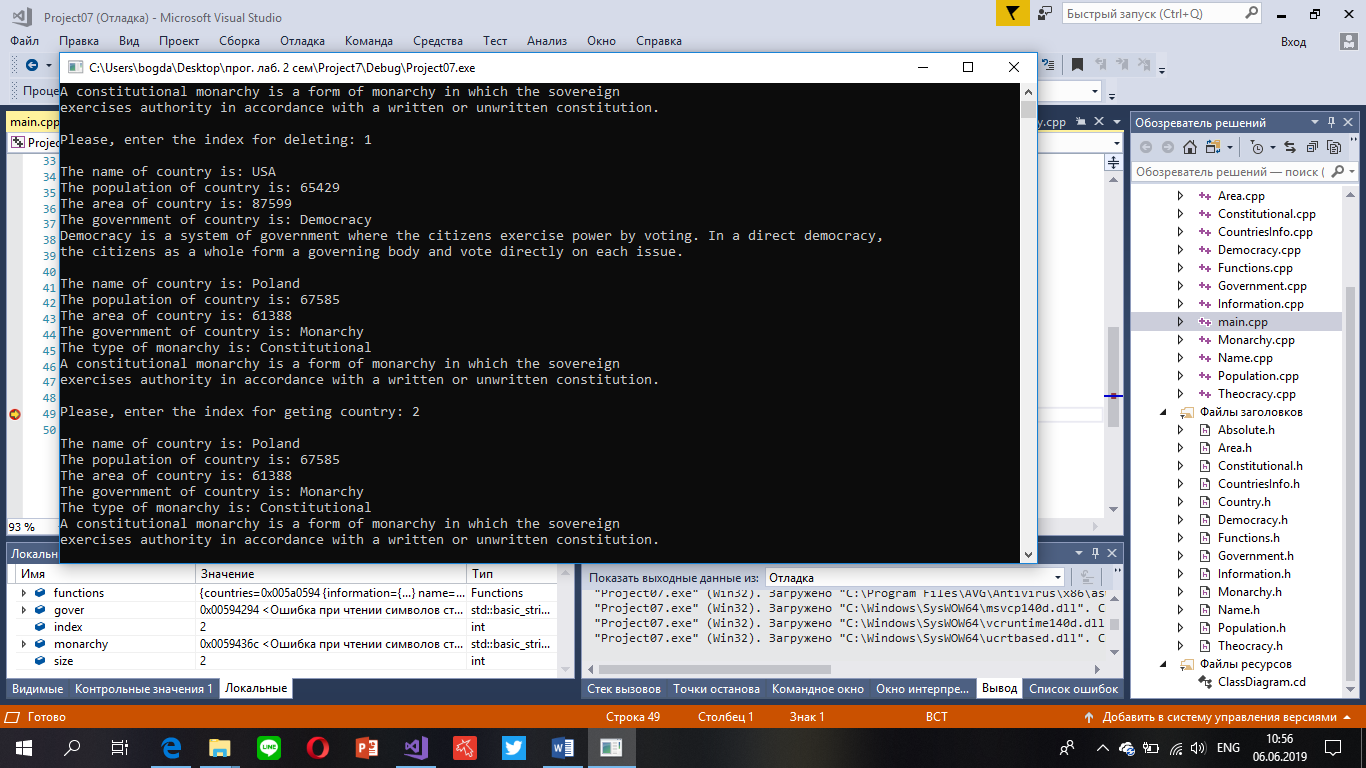


Рисунок № 3. Результат роботи програми

**Функції:**

1. Створення масиву
2. Додавання елементу
3. Видалення елементу
4. Пошук по індексу
5. Видалення масиву елементів

**Висновок:** Отримав знання про парадигму ООП – поліморфізм. Навчився застосовувати отримані знання на практиці.