**Розрахункове завдання з дисципліни «Програмування»**

Тема: Розробка інформаційно-довідкової системи.

Мета: Закріпити отримані знання з дисципліни «Програмування» шляхом виконання типового комплексного завдання.

**1. ВИМОГИ**

**1.1 Інформація про розробника:**

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

Кожний студент отримує індивідуальне завдання. Варіант завдання обирається за номером прізвища студента у журналі групи. При виконанні завдання з розробки інформаційно-довідкової системи необхідно виконати наступне:

1) з табл. 1, відповідно до варіанта завдання, обрати прикладну галузь;

2) дослідити літературу стосовно прикладної галузі. За результатами аналізу літератури оформити перший, аналітичний розділ пояснювальної записки обсягом 2–3 сторінки;

3) для прикладної галузі розробити розгалужену ієрархію класів, яка складається з не менш ніж трьох класів, один з яких є «батьком» для інших (класів-спадкоємців). Класи повинні мати перевантажені оператори введення-виведення даних та порівняння;

4) розробити клас-контролер, що буде включати колекцію розроблених класів, та наступні методи роботи з цією колекцією:

а) читання даних з файлу та їх запис у контейнер;

б) запис даних з контейнера у файл;

в) сортування елементів у контейнері за вказаними критеріями: поле та напрям сортування, які задаються користувачем з клавіатури;

г) пошук елементів за вказаним критерієм (див. «Завдання для обходу колекції» в табл. 1);

5) розробити клас, який має відображати діалогове меню для демонстрації реалізованих функцій класу контролера;

6) оформити схеми алгоритмів функцій класів контролера та діалогового меню;

7) оформити документацію: пояснювальну записку (див. розділ 2 даних методичних вказівок). Увага. Текст програми та результати роботи програми мають бути подані в додатках.

* 1. **Вимоги**

− усі класи повинні мати конструктори та деструктори;

− якщо функція не змінює поля класу, вона має бути декларована як константна;

− рядки повинні бути типу string;

− при перевантаженні функції треба використовувати ключове слово override; − програмний код усіх класів має бути 100 % doxygen- документований;

− у звіті текст програми слід оформляти стилем Courier new 8 пт, інтервал

– одиничний; довжина рядка не повинна перевищувати 80 символів.

* 1. **Додаткові вимоги**

Додаткові вимоги на оцінку «добре»:

− виконання основного завдання та додаткових наступних вимог:

− додати обробку помилок; при цьому функція, що генерує виключення, при її декларуванні повинна мати ключове слово throw;

− виконати перевірку вхідних даних за допомогою регулярних виразів.

Додаткові вимоги на оцінку «відмінно»:

− виконати завдання відповідно до вимог на оцінку «добре» та додаткові наступні вимоги:

− критерій для пошуку та сортування задавати у вигляді функтора;

− розробити клас-тестер, основною метою якого буде перевірка коректності роботи класу-контролера.

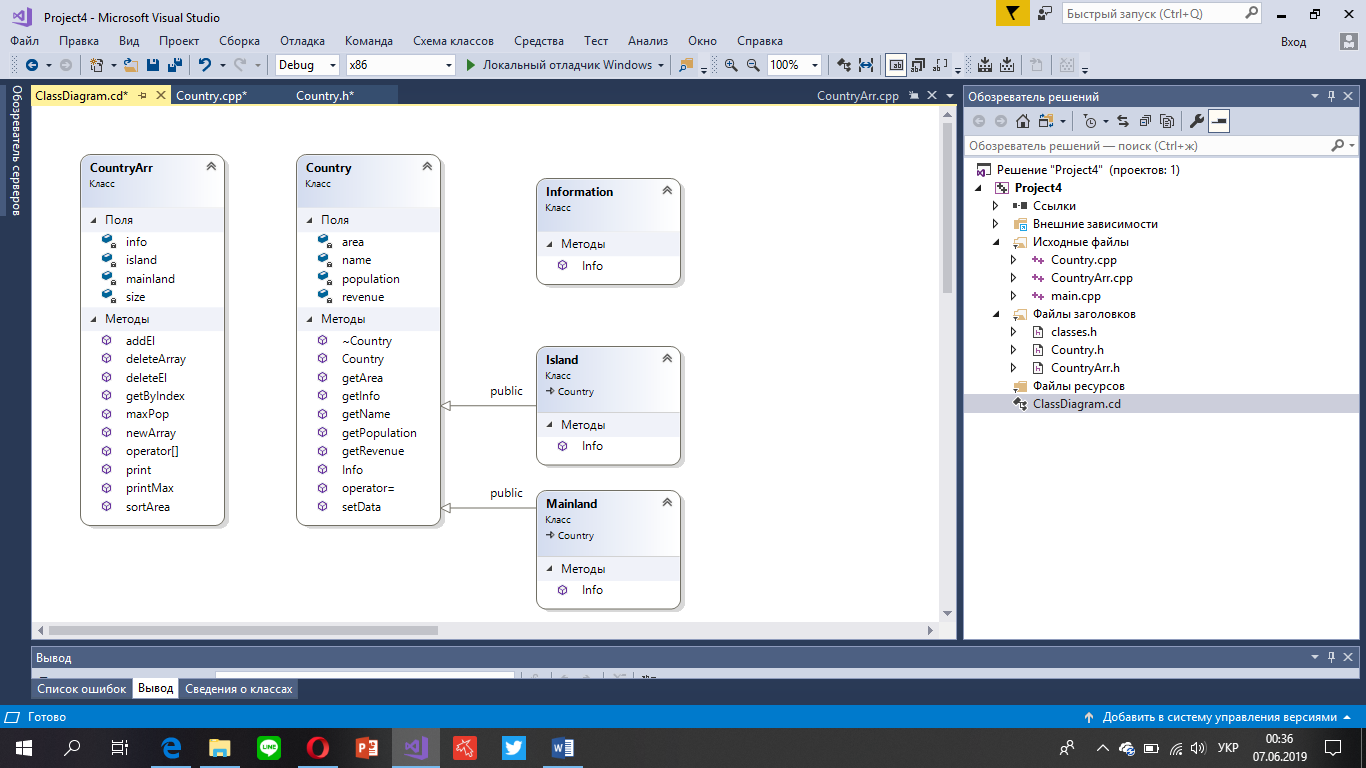
1. **ОПИС ПРОГРАМИ**

**2.1 Функціональне призначення**

Програма призначена щоб отримувати та зберігати інформацію щодо різних країн світу та пошуку країни з максимальним населенням.

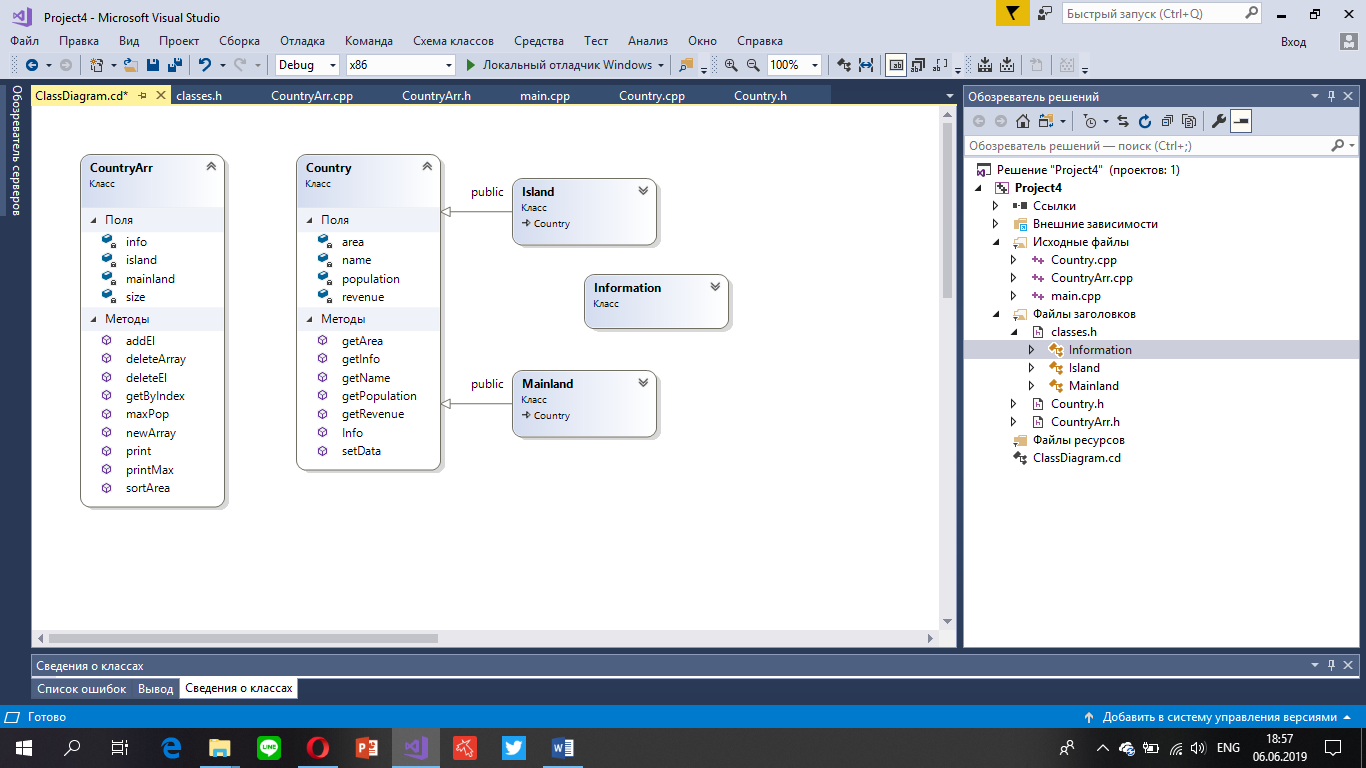
* 1. **Опис логічної структури**

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 базового класу:

class Country {

private:

int population;

int area;

int revenue;

string name;

public:

Country& operator= (const Country &obj);

friend std::ostream& operator<< (std::ostream &out, const Country &obj);

friend std::istream& operator>> (std::istream &in, Country &obj);

Country();

int getPopulation();

int getArea();

int getRevenue();

string getName();

void getInfo(std::string s);

void setData(int population, int area, int revenue, string name);

virtual void Info() = 0;

~Country();

};

Код програми класів нащадків:

class Island : public Country {

public:

void Info() override {

cout << "This country is washed by the ocean." << endl;

}

};

class Mainland : public Country {

public:

void Info() override {

cout << "This country is on the mainland" << endl;

}

};

Код програми класу контролеру:

class CountryArr {

private:

int size = 0;

Island \*island;

Mainland \*mainland;

Information info;

public:

void newArray(std::string name);

void print();

void addEl(int population, int area, int revenue, std::string s);

void deleteEl(int l);

void getByIndex(int index);

void deleteArray();

Island maxPop();

void printMax(Island max);

void sortArea(bool(\*comp)(int x, int y));

friend std::ostream& operator<< (std::ostream &out, const CountryArr &obj);

friend std::istream& operator>> (std::istream &in, CountryArr &obj);

Country &operator[] (const int index);

};

Код головної функції:

int main() {

CountryArr Countr;

std::regex regex\_spaces("[\\s]{2,}");

std::regex regex\_firstSymbol("^[A-Z]");

std::string name;

std::cout << "Please, the name of country: ";

getline(std::cin, name);

if (!(regex\_search(name, regex\_firstSymbol)) || regex\_search(name, regex\_spaces)) {

std::cout << "Incorrect entry, writing with large letters(A - Z) and without double spaces : " << std::endl;

std::cout << "Please, the name of country: ";

getline(std::cin, name);

}

Countr.newArray(name);

Countr.print();

bool(\*p)(int a, int b);

int option = 0;

do {

std::cout << "Choose option:" << std::endl << "0 - Exit " << std::endl << "1 - Add element" << std::endl << "2 - Delete element" << std::endl << "3 - Get by index" << std::endl << "4 - Search by max population" << std::endl << "5 - Sort by area" << std::endl;

std::cout << std::endl;

std::cin >> option;

switch (option) {

case 1: {

int population, area, revenue;

std::string name;

std::ifstream fin("data.txt");

fin >> population >> area >> revenue >> name;

Countr.addEl(population, area, revenue, name);

fin >> population >> area >> revenue >> name;

Countr.addEl(population, area, revenue, name);

fin >> population >> area >> revenue >> name;

Countr.addEl(population, area, revenue, name);

system("cls");

Countr.print();

break;

}

case 2: {

auto id = 0;

std::cout << std::endl << "Enter index: ";

std::cin >> id;

std::cout << std::endl;

Countr.deleteEl(id);

system("cls");

Countr.print();

break;

}

case 3: {

auto index = 0;

std::cout << std::endl << "Enter index : ";

std::cin >> index;

std::cout << std::endl;

system("cls");

Countr.getByIndex(index);

break;

}

case 4: {

Island Max = Countr.maxPop();

Countr.printMax(Max);

break;

}

case 5: {

int ch;

std::cout << "Please, enter the type of sort" << std::endl << "1 - up, 0 - down: ";

std::cin >> ch;

std::cout << std::endl;

if (ch == 1) {

p = comp;

}

else if (ch == 0) {

p = comp2;

}

else {

std::cout << "You enter false variant" << std::endl;

break;

}

Countr.sortArea(p);

break;

}

default: {

break;

}

}

} while (option != 0);

Countr.deleteArray();

\_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 роботи функції ініціалізації масиву

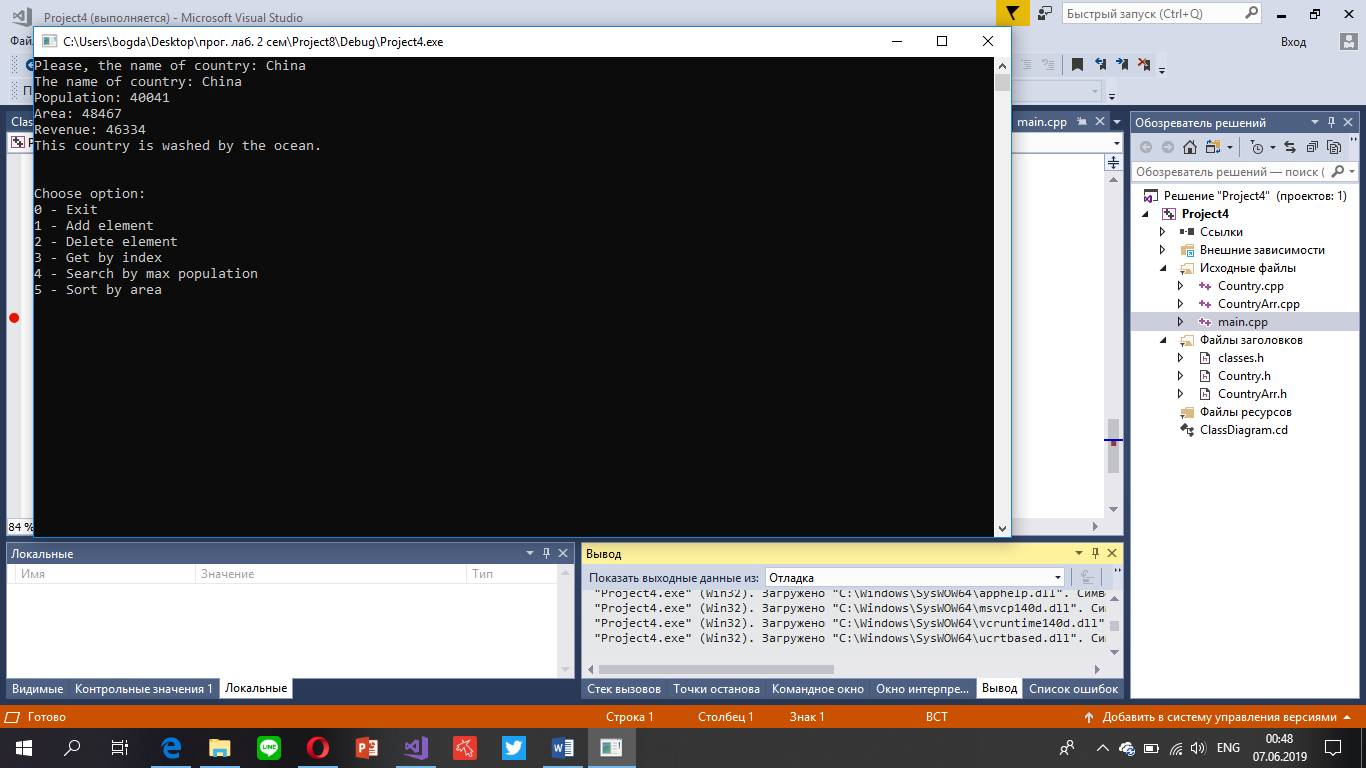


Рисунок № 3. Ініціалізація масиву

На рисунку № 4 зоображено результат роботи функції додавання нового елемонту до масиву

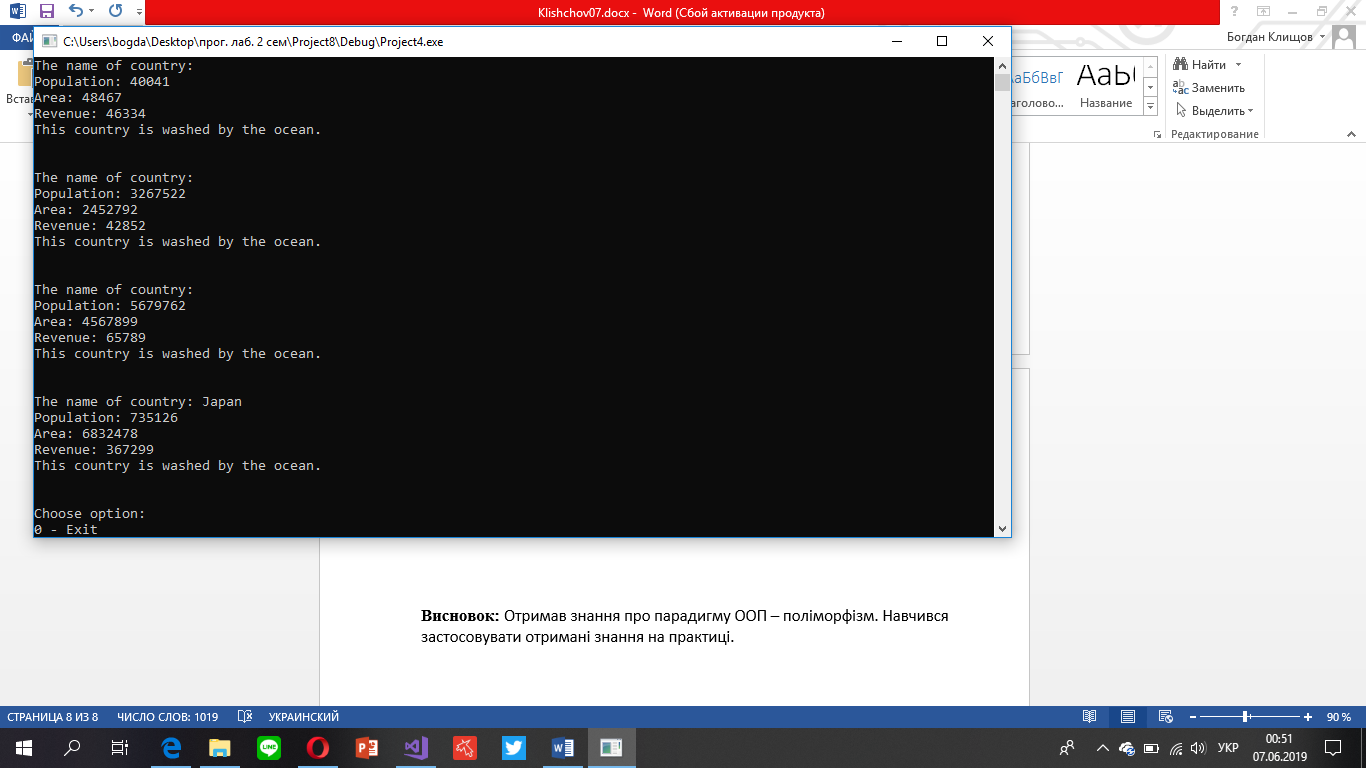


Рисунок № 4. Додавання нових елементів

На рисунку № 5 зоображено результат роботи функції видалення елементу масиву:

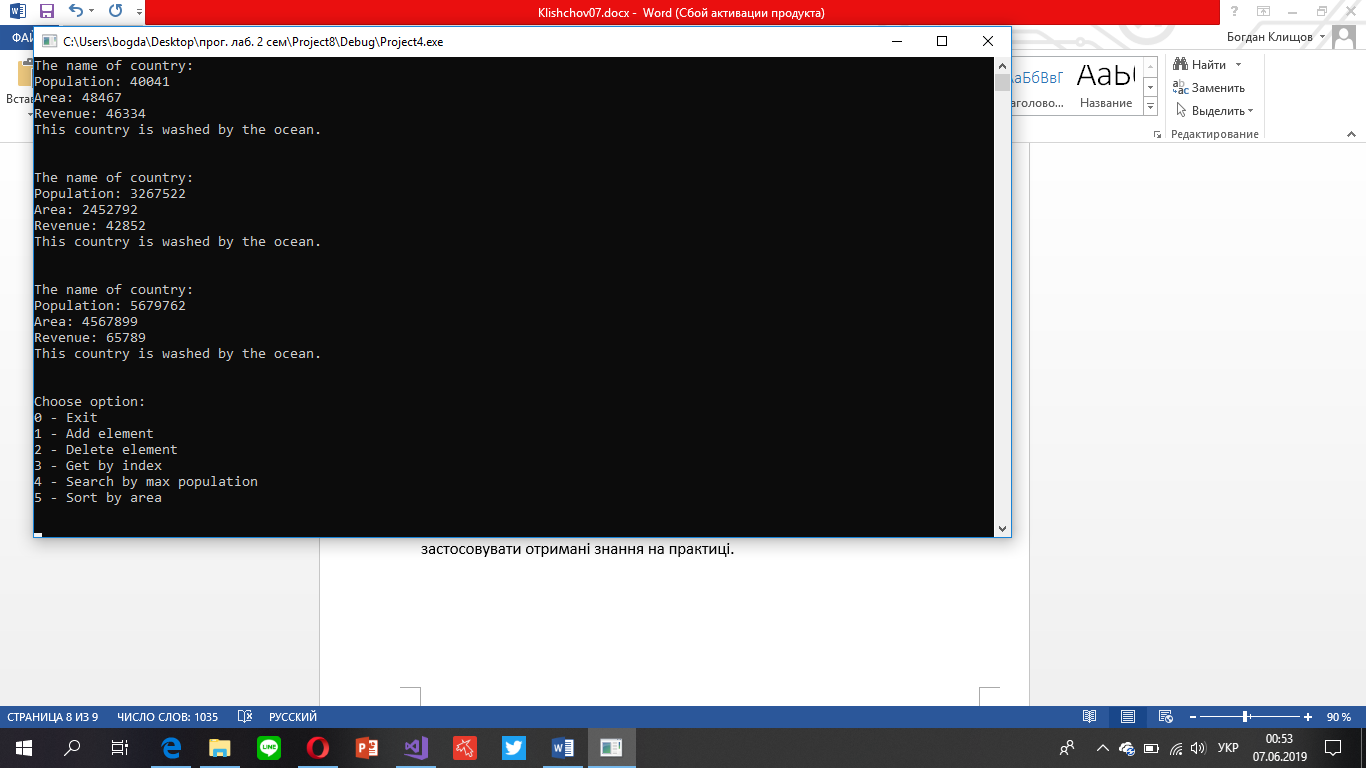


Рисунок № 5. Видалення елементу

На рисунку № 6 зоображено результат роботи функції отримання по індексу:

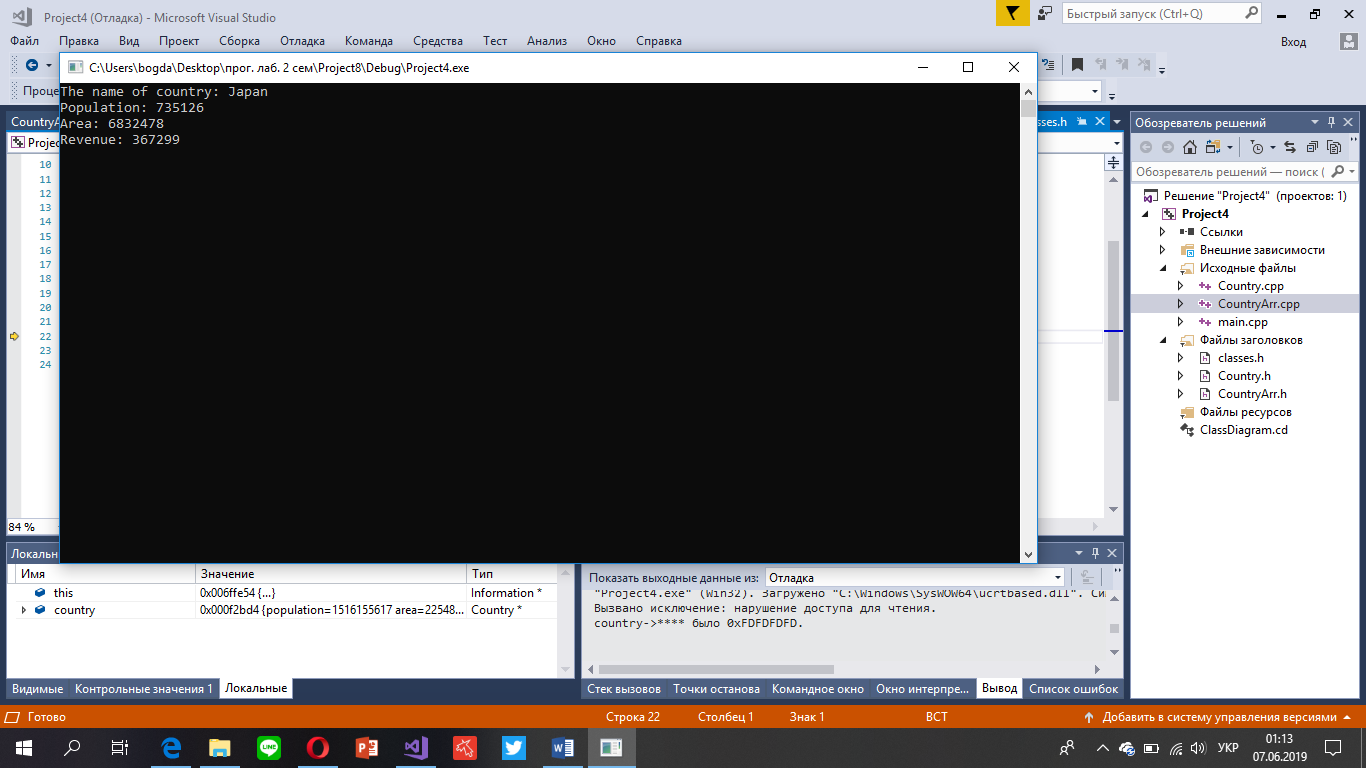


Рисунок № 6. Отримання по індексу

На рисунку № 7 зоображено результат роботи регулярних виразів:

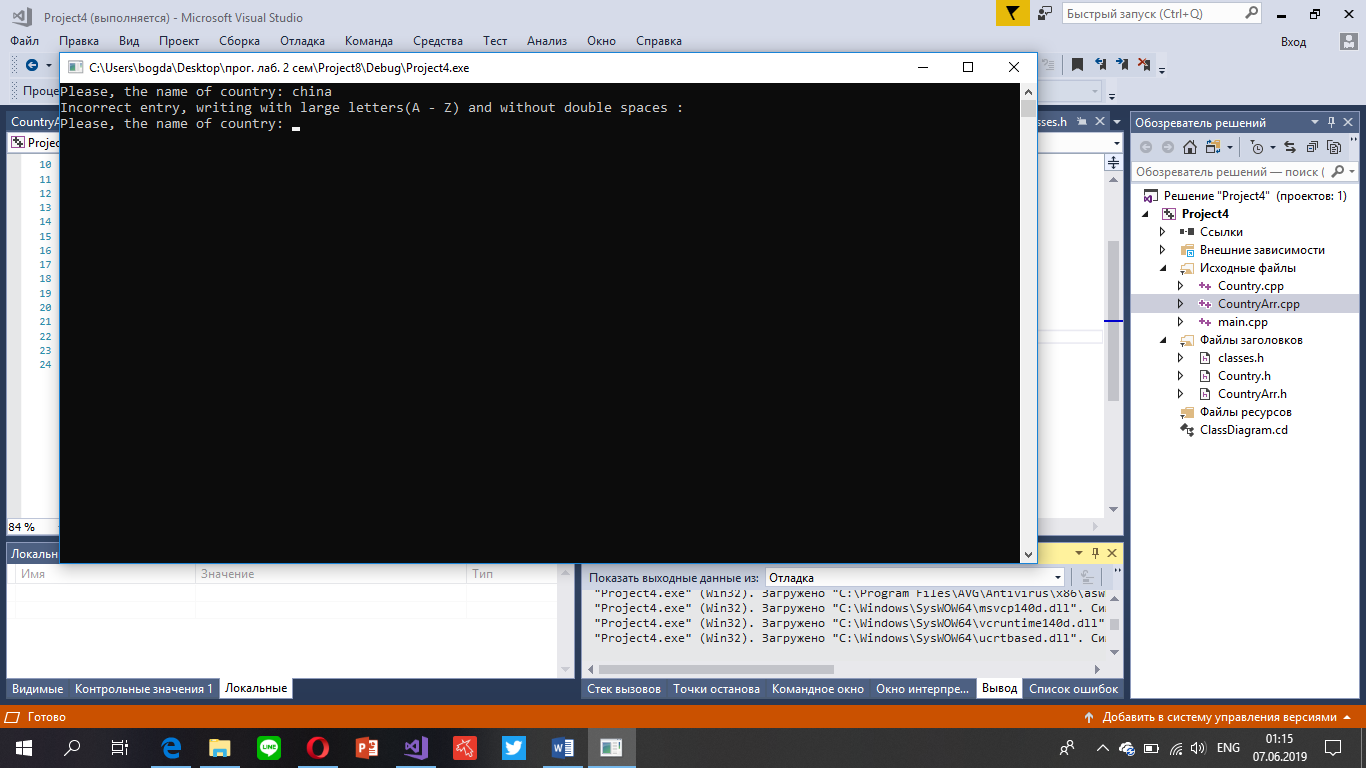


Рисунок № 7. Регулярні вирази

**Висновок:** Закріпив отримані знання з дисципліни «Програмування» шляхом виконання типового комплексного завдання.