

Лабораторна робота №3. Потоки

Тема: Робота з потоками: потокове введення / виведення на консоль та у файл, рядки типу string, stringstream.

Мета: Отримати знання про основи роботи з потоковим введенням / виведенням на мові C++, роботу з файлами та рядками типу string.

ВИМОГИ

1.1 Інформація про розробника:

- Кліщов Б. Р.
- КІТ 102.8а

1.2 Загальне завдання

Поширити попередню лабораторну роботу наступним чином:

- використання функцій printf/scanf замінити на використання cin/cout;
- усі конкатенації рядків замінити на використання stringstream;
- замінити метод виводу інформації про об'єкт на метод, що повертає рядок-інформацію про об'єкт, який далі можна виводити на екран;
- замінити метод вводу інформації про об'єкт на метод, що приймає рядок з інформацією про об'єкт, обробляє його та створює об'єкт на базі цієї інформації;
- поширити клас-список, шляхом реалізації методів роботи з файлами за допомогою файлових потоків (fstream) (якщо використовувалися функції fprintf/fscanf – замінити їх на класи ifstream/ofstream), при цьому сигнатури методів повинні виглядати наступним чином:

- читання: void CList::readFromFile(string fileName); де CList – клас-список об'єктів, при цьому слід пам'ятати, що при повторному читанні з файлу, попередні дані списку повинні бути очищені;

- запис: void CList::writeToFile(string fileName);

1.3 Додаткові умови виконання завдання:

- продемонструвати відсутність витоків пам'яті;
- продемонструвати роботу розроблених 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 C, а також не повинні використовуватися рядки типу char*.

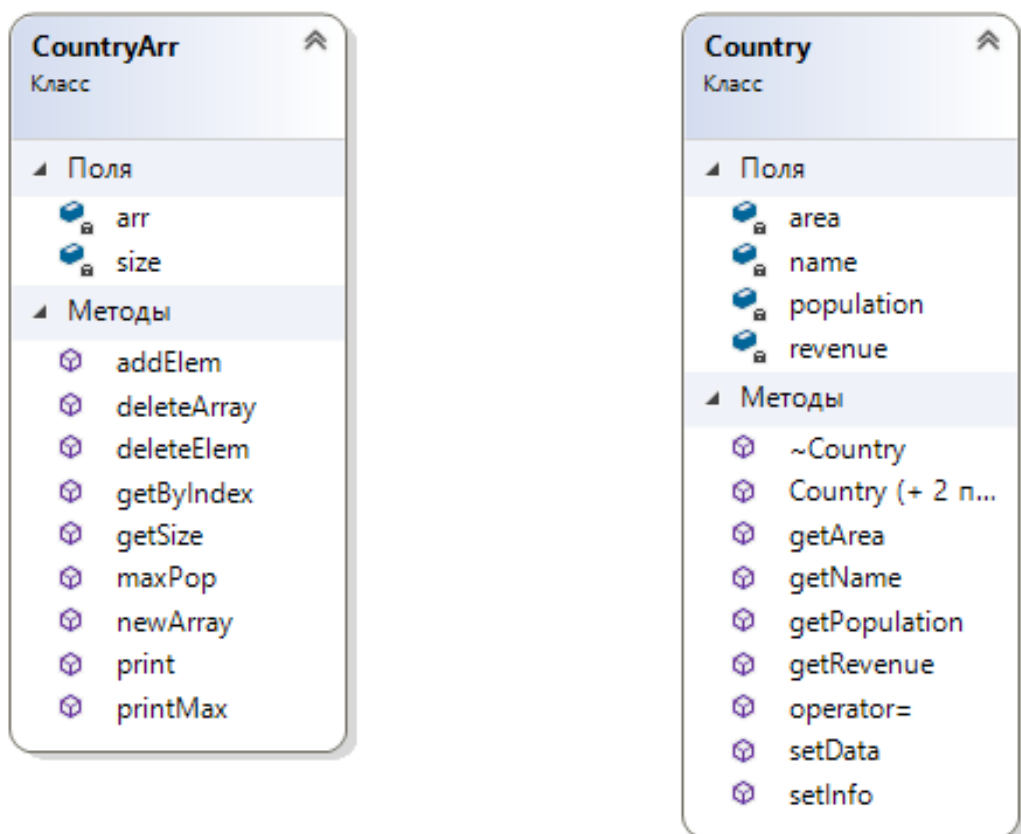
2. ОПИС ПРОГРАМИ

2.1 Функціональне призначення

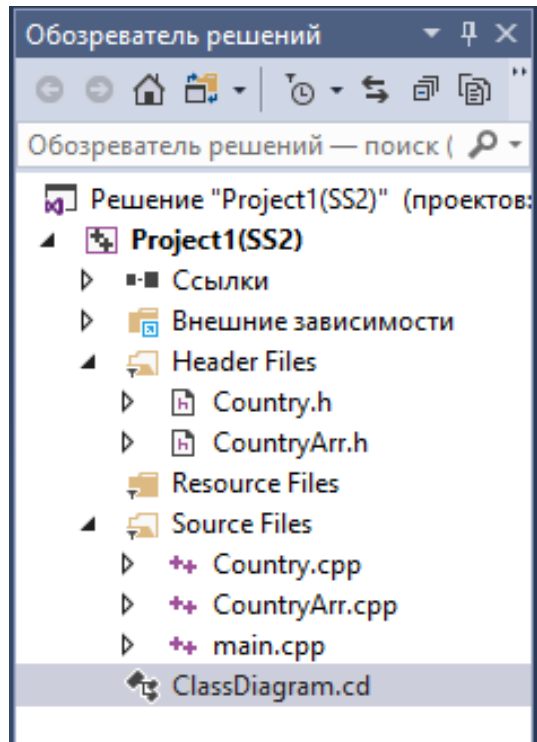
Програма призначена щоб отримувати та зберігати інформацію щодо різних країн світу. Інформацію можна зчитувати з файлу та записувати в нього.

2.2 Опис логічної структури

На рисунку № 1 зображена діаграма класу



На рисунку № 2 зображена структура програми:



Малюнок №2: Структура програми

2.3 Важливі фрагменти програми

Код програми:

Main:

```
int main() {  
  
    CountryArr *country = new CountryArr();  
    Country *Max = new Country();  
  
    int id = 1;  
    country->getSize(id); //set size!!!  
  
    std::ifstream fin("data.txt");  
  
    int population;  
    int area;  
    int revenue;  
  
    std::string name;
```

```

std::cout << "Enter name: ";
std::cin >> name;

country->newArray(name);
fin >> population >> area >> revenue >> name;
country->addElem(population, area, revenue, name);
fin >> population >> area >> revenue >> name;
country->addElem(population, area, revenue, name);
fin >> population >> area >> revenue >> name;
country->addElem(population, area, revenue, name);
country->deleteElem(0);
country->getByIndex(1);
country->print();
*Max = country->maxPop();
country->printMax(*Max);
country->deleteArray();

fin.close();

delete Max;
delete country;

_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;
}

```

Функція для запису результату в файл:

```

void CountryArr::print() {
    std::ofstream fout("result.txt");
    for (int i = 0; i < size; i++) {
        int population = arr[i].getPopulation();
        int area = arr[i].getArea();
        int revenue = arr[i].getRevenue();
        std::string name = arr[i].getName();

        fout << "The name of country: " << name << std::endl;
        fout << "Population: " << population << std::endl;
        fout << "Area: " << area << std::endl;
        fout << "Revenue: " << revenue << std::endl << std::endl;
    }
    fout.close();
}

```

Конструктори:

1. Без параметрів:

```

Country::Country() :population(0), area(0), revenue(0), name() {
    name = new char[24];
};

```





2. З параметрами:

```
Country::Country(int a, int b, int c, char* Name) :population(a), area(b), revenue(c) {  
    name = new char[24];  
    strcpy_s(name, 24, Name);  
};
```

3. Копіювальний:

```
Country::Country(const Country &obj) :population(obj.population), area(obj.area),  
revenue(obj.revenue), name(obj.name) {};
```

Файли:

 data.txt	26.03.2019 22:38	Текстовый докум...	1 КБ
 result.txt	26.03.2019 23:33	Текстовый докум...	1 КБ
 resultID.txt	26.03.2019 23:33	Текстовый докум...	1 КБ
 resultMax.txt	26.03.2019 23:33	Текстовый докум...	1 КБ

Було створено чотири файли формату .txt :

data.txt – для читання інформації про країни

result.txt – сюди записується результат роботи функції `void CountryArr::print()`

resultID.txt – для запису роботи функції `void getByIndex(int index);`

resultMax.txt – для запису результату `Country maxPop();`

Функція для запису в файл країни з максимальною щільністю населення:

```
void CountryArr::printMax(Country min) {  
    std::ofstream fout("resultMax.txt");  
    int population = min.getPopulation();  
    int area = min.getArea();  
    int revenue = min.getRevenue();  
    std::string name = min.getName();  
  
    fout << "The name of country: " << name << std::endl;  
    fout << "Population: " << population << std::endl;  
    fout << "Area: " << area << std::endl;  
    fout << "Revenue: " << revenue << std::endl << std::endl;  
    fout.close();  
}
```

3 ВАРІАНТИ ВИКОРИСТАННЯ

3.1 Результат роботи функцій

На рисунку № 3 зображено результат запису даних в файл



Рисунок № 3. Запис даних в файл

На рисунку № 4 зображено результат запису в файл країни, отриманої по ID

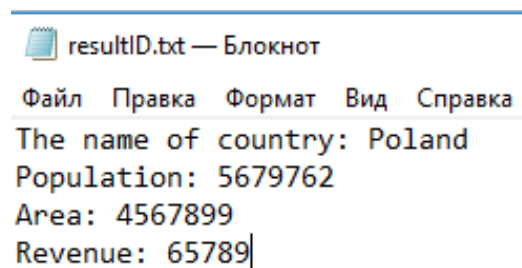


Рисунок № 4. Пошук по ID

На рисунку № 5 зображено результат запису в файл країни, з максимальною кількістю населення

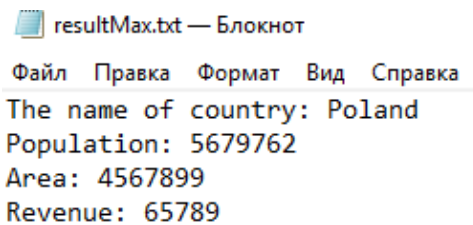


Рисунок № 5. Мінімальна щільність населення

Програма має декілька функцій:

1. Додавання елементу
2. Видалення елементу
3. Пошук по індексу
4. Запис в файл
5. Видалення масиву елементів

Висновок: Я отримав знання про основи роботи з потоковим введенням / виведенням на мові C++, роботу з файлами та рядками типу string. Було перевірено програму на витоки пам'яті: вони відсутні.