

Лабораторна робота №2. Перевантаження методів

Тема: Класи. Конструктори та деструктори. Перевантаження методів.

Мета: Отримати базові знання про класи. Дослідити механізм інкапсуляції.

ВИМОГИ

1.1 Інформація про розробника:

Кліщов Б. Р.

КІТ 102.8а

1.2 Загальне завдання

Поширити попередню лабораторну роботу наступним чином:

- в базовому класі необхідно додати:
- мінімум одне поле типу `char*`;
- конструктор за замовчуванням, копіювання та конструктор з аргументами;
- деструктор;
- в клас-список потрібно додати метод обходу масиву для виконання індивідуального завдання.

1.3 Додаткові умови виконання завдання:

- реалізація конструкторів повинна бути продемонстрована за допомогою списків ініціалізацій;
- конструктори та деструктори повинні мати логіруючі повідомлення.
- студент повинен продемонструвати виклик деструктора та кожного типу конструктора, а також пояснити, коли вони викликаються;
- продемонструвати відсутність витоків пам'яті;
- продемонструвати роботу розроблених методів класу-списку за допомогою модульних тестів.

Індивідуальні завдання:

В табл. 1.2 обрати прикладну галузь по варіанту у відповідності до номера у журналі групи.

Варіант 8: Визначити, яка країна має найбільшу щільність населення.

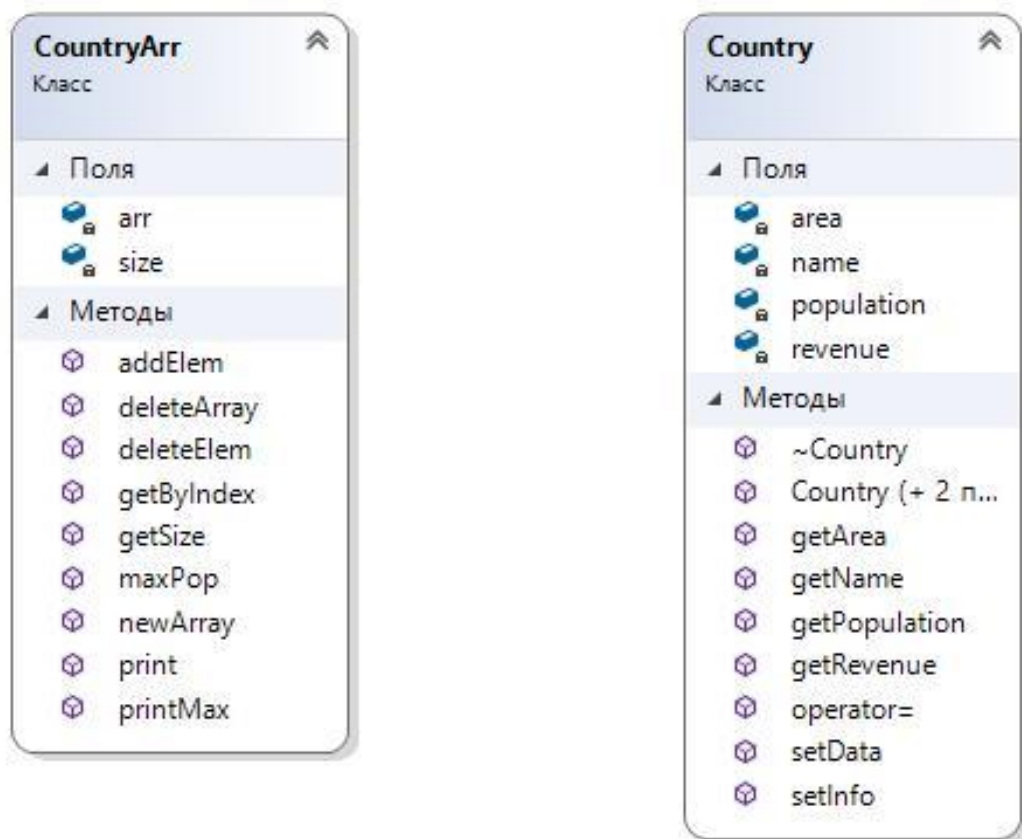
2. ОПИС ПРОГРАМИ

2.1 Функціональне призначення

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

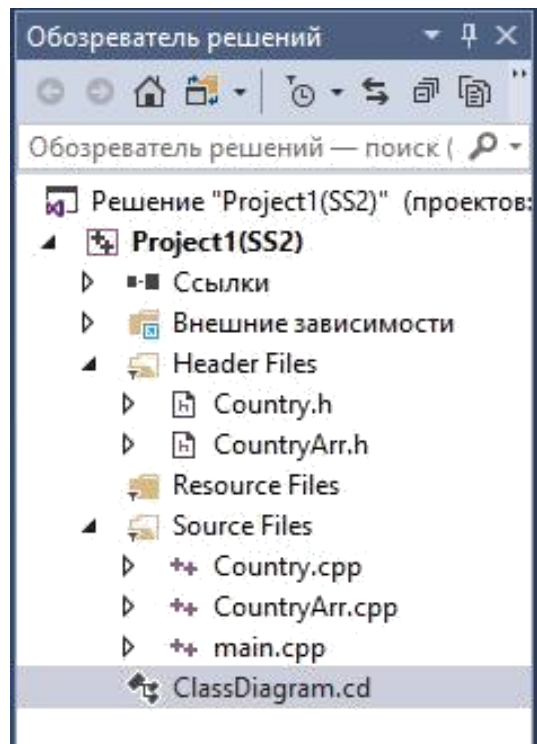
2.2 Опис логічної структури

На рисунку № 1 зображена діаграма класу



Малюнок №1. Діаграма класу

На рисунку № 2 зображена структура програми:



Малюнок №2: Структура програми

2.3 Важливі фрагменти програми

Код програми:

Main:

```
int main() {

    CountryArr country;
    Country Max;

    int id = 1;
    country.getSize(id); //set size!!!

    std::string name;

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

    country.newArray(name);
    country.addElem(232142, 53423, 654321, "Ukraine");
    country.addElem(32142, 5323, 23321, "Poland");
    country.addElem(2422, 523423, 5654321, "Japan");
    country.deleteElem(1);
    country.getByIndex(3);
    country.print();
    Max = country.maxPop();
    country.printMax(Max);
    country.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();

        getchar();

        return 0;
}

```

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) {};

```

Функція для знаходження країни з максимальною щільністю населення:

```

Country CountryArr::maxPop() {
    int max = arr[0].getPopulation();
    int index = 0;
    for (int i = 0; i < size; i++) {
        if (max < arr[i].getPopulation()) {
            max = arr[i].getPopulation();
            index = i;
        }
    }
    return arr[index];
}

```

Функція для виводу в консоль країни з максимальною щільністю населення:

```

void CountryArr::printMax(Country min) { {
    int population = min.getPopulation();
    int area = min.getArea();
}
}

```

```

int revenue = min.getRevenue();
char *name = min.getName();

std::cout << "The name of country: " << name << std::endl;
std::cout << "Population: " << population << std::endl;
std::cout << "Area: " << area << std::endl;
std::cout << "Revenue: " << revenue << std::endl << std::endl;
}

```

3 ВАРІАНТИ ВИКОРИСТАННЯ

На рисунку № 3 зображено результат роботи програми.



```

Enter name:  USA

The name of country: Ukraine
Population: 232142
Area: 53423
Revenue: 654321

The name of country: Ukraine
Population: 232142
Area: 53423
Revenue: 654321

The name of country: Poland
Population: 32142
Area: 5323
Revenue: 23321

The name of country: Japan
Population: 2422
Area: 523423
Revenue: 5654321

The name of country: Ukraine
Population: 232142
Area: 53423
Revenue: 654321

```

Програма має декілька функцій:

1. Додавання елементу
2. Видалення елементу
3. Пошук по індексу
4. Вивід в консоль
5. Видалення масиву елементів

Використовувати програму можна для збереження даних про певних країн.

Наприклад:

```
The name of country: Bolgar
Population: 40041
Area: 48467
Revenue: 46334

The name of country: USA
Population: 2322
Area: 53423
Revenue: 654321

The name of country: France
Population: 32142
Area: 5323
Revenue: 23321

The name of country: Spain
Population: 2422
Area: 52423
Revenue: 654321

The name of country: Spain
Population: 222
Area: 523423
Revenue: 5654321

The name of country: Spain
Population: 242
Area: 423
Revenue: 4321
```

Висновок: Я придбав практичні навички щодо розроблення програм із використанням структурованих типів даних. Програму перевірено на витоки пам'яті. Вони відсутні.