

# Лабораторна робота №8. Перевантаження операторів

Тема: Перевантаження операторів. Сериалізація.

Мета: Отримати знання про призначення операторів, визначити їх ролі в житті об'єкта та можливість перевизначення.

## 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, та класі/класах-спадкоємцях перевантажити:
  - оператор присвоювання;
  - оператор порівняння (на вибір: `==` , `<` , `>` , `>=` , `<=` , `!=` );
  - оператор введення/виведення;
  - в класі-списку перевантажити:
  - оператор індексування ( `[ ]` );
  - оператор введення/виведення з акцентом роботи в тому числі і з файлами.
- При цьому продовжувати використовувати регулярні вирази для валідації введених даних.

### 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 робити `using` кожного необхідного класу: `using std::string` , `using std::cout`.

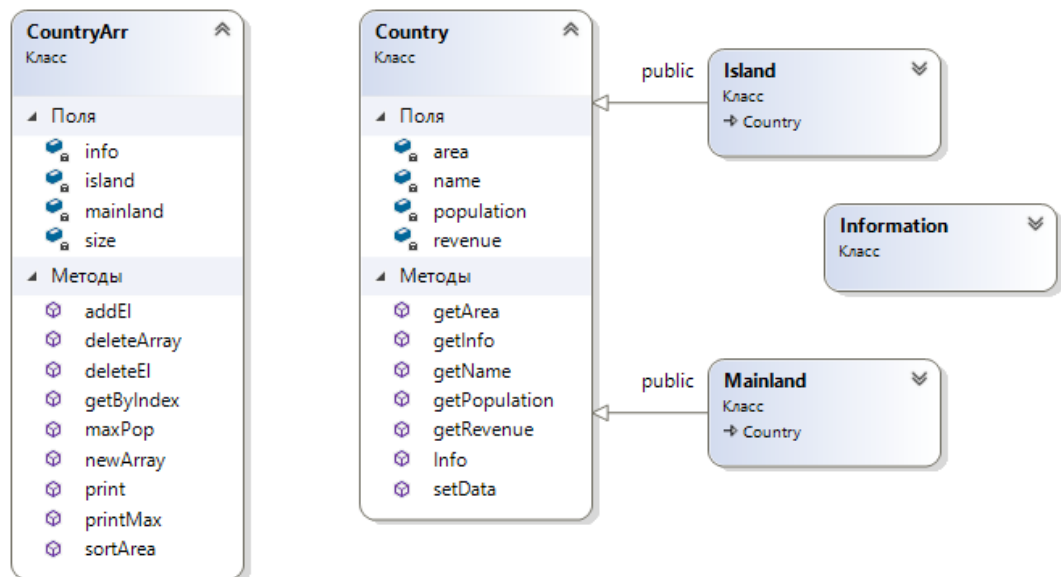
## 2. ОПИС ПРОГРАМИ

### 2.1 Функціональне призначення

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

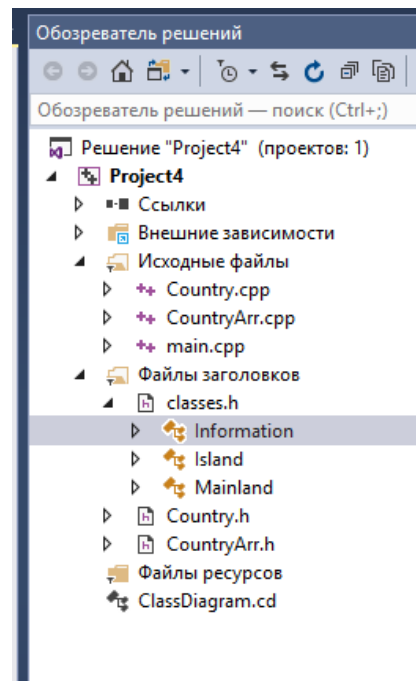
### 2.2 Опис логічної структури

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 базового класу та класів спадкоємців:

```
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;
    }
};

class Information {
public:
    void Info(Country *country)
    {
        country->Info();
    }
};
```

```

class Country {
private:
    int population;
    int area;
    int revenue;
    string name;
public:
    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;
};

```

## Перевантажені оператори:

```

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

    out << obj.name << ": " << obj.area << std::endl;
    out << obj.name << ": " << obj.population << std::endl;
    return out;
}

std::istream& operator>> (std::istream &in, Country &obj) {
    in >> obj.name;
    in >> obj.population;
    in >> obj.area;
    in >> obj.revenue;
    return in;
}

Country& Country::operator= (const Country &obj) {
    area = obj.area;
    population = obj.population;
    revenue = obj.revenue;
    return *this;
}

```

```

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

    out << obj.size << std::endl;
    for (int i = 0; i < obj.size; i++) {
        out << obj.island[i];
    }
    return out;
}

```

```

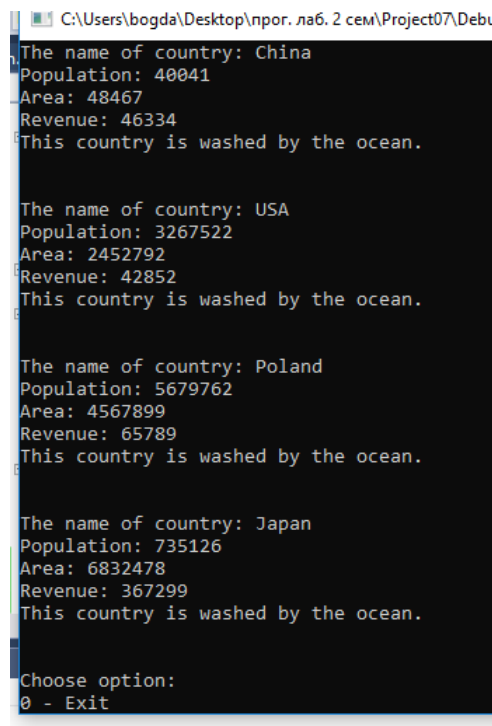
std::istream& operator>> (std::istream &in, CountryArr &obj) {
    in >> obj.size;
    for (int i = 0; i < obj.size; i++) {
        in >> obj.island[i];
    }
    return in;
}
Country& CountryArr::operator[] (const int index) {
    return island[index];
}

```

## 3 ВАРІАНТИ ВИКОРИСТАННЯ

### 3.1 Результат роботи функцій

На рисунку № 3 зображено результат роботи програми



```

C:\Users\bogda\Desktop\mpor. лаб. 2 сем\Project07\Debu
The name of country: China
Population: 40041
Area: 48467
Revenue: 46334
This country is washed by the ocean.

The name of country: USA
Population: 3267522
Area: 2452792
Revenue: 42852
This country is washed by the ocean.

The name of country: Poland
Population: 5679762
Area: 4567899
Revenue: 65789
This country is washed by the ocean.

The name of country: Japan
Population: 735126
Area: 6832478
Revenue: 367299
This country is washed by the ocean.

Choose option:
0 - Exit

```

Рисунок № 3. Результат роботи програми

**Висновок:** Отримав знання про призначення операторів, визначити їх ролі в житті об'єкта та можливість перевизначення.

