

Лабораторна робота №13. Алгоритми переміщення та пошуку

Тема: STL. Алгоритми переміщення та пошуку.

Мета: На практиці порівняти STL алгоритми, що не модифікують послідовність.

ВИМОГИ

1.1 Інформація про розробника:

- Кліщов Б. Р.
- КІТ 102.8а

1.2 Загальне завдання

Поширити попередню лабораторну роботу, додаючи наступні можливості діалогового меню:

- вивід всіх елементів масиву за допомогою STL функції `for_each`;
- визначення кількості елементів за заданим критерієм;
- пошук елемента за заданим критерієм.

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.`

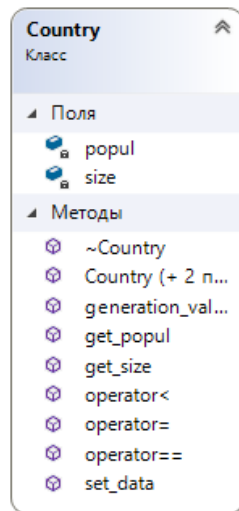
2. ОПИС ПРОГРАМИ

2.1 Функціональне призначення

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

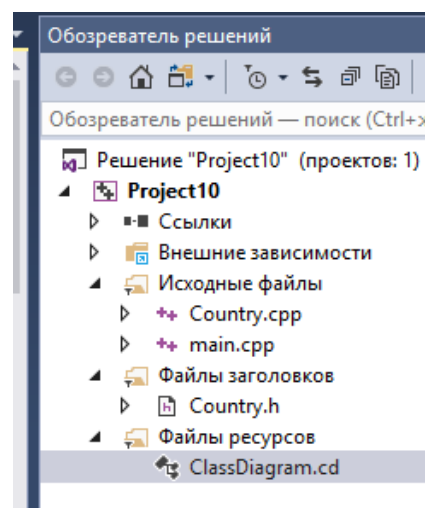
2.2 Опис логічної структури

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 :

```
#include
<iostream>

#include <vector>
#include <set>
#include <map>
#include <list>
#include <iterator>
#include "Country.h"

using namespace std;

struct Print {
    Print(){}
    void operator()(Country printCountry) { cout << printCountry << " ";
    cout << endl; }

};

void Vector() {
    vector<Country> myVector;
    int choose, id;
    Country country;
    string name;
    cout << "Vector" << endl;
    do {
        cout << "0 - Exit" << endl;
        cout << "1 - Print vector" << endl;
        cout << "2 - Find element by index" << endl;
        cout << "3 - Add element" << endl;
        cout << "4 - Delete element" << endl;
        cout << "5 - Number of elements" << endl;
        cout << "Choose: ";
        cin >> choose;

        switch (choose)
        {
            case 1:
```

```

        for_each(myVector.begin(), myVector.end(), Print());
        break;
    case 2:
        cout << "input index: ";
        cin >> id;
        cout << myVector.at(id);
        break;
    case 3: {
        cout << "Please, enter the name of country: ";
        cin >> name;
        cout << endl;
        country.setData(name);
        myVector.push_back(country);
        break;
    }
    case 4:
        myVector.clear();
        break;
    case 5:
        cout << "Number of elements is: " << myVector.size() <<
endl;

        break;
    default:
        break;
    }
} while (choose != 0);
}

```

```

void Set() {
    int choose, id;
    set<Country> mySet;
    Country country;
    string name;
    cout << "Set" << endl;

    do {
        cout << "0 - Exit" << endl;
        cout << "1 - Print" << endl;
        cout << "2 - Add element" << endl;
        cout << "3 - Delete element" << endl;
        cout << "4 - Number of elements" << endl;
        cout << "Choose: ";
        cin >> choose;

        switch (choose)

```

```

{
    case 1:
        for (auto &it : mySet) {
            cout << it << endl;
        }
        break;
    case 2: {
        cout << "Please, enter the name of country: ";
        cin >> name;
        cout << endl;
        country.setData(name);
        mySet.insert(country);
        break;
    }
    case 3:
        mySet.erase(country);
        break;
    case 4:
        cout << "Number of elements is: " << mySet.size() <<
endl;
        break;
    default:
        break;
}
} while (choose != 0);
}

```

```

void List() {
    int choose, id;
    list<Country> myList;
    Country country;
    string name;
    cout << "List" << endl;
    do {
        cout << "0 - Exit" << endl;
        cout << "1 - Print" << endl;
        cout << "2 - Find element by index" << endl;
        cout << "3 - Add element" << endl;
        cout << "4 - Delete element" << endl;
        cout << "5 - Number of elements" << endl;
        cout << "Choose: ";
        cin >> choose;

        auto it = myList.begin();
    }
}

```

```

switch (choose)
{
case 1:
    for (auto i = myList.cbegin(); i != myList.cend(); i++) {
        cout << *i;
    }
    break;
case 2:
    cout << "input index: ";
    cin >> id;

    advance(it, id);

    cout << id;
    break;
case 3: {
    it = myList.begin();
    cout << "Please, enter the name of country: ";
    cin >> name;
    cout << endl;
    country.setData(name);
    myList.insert(it, country);

    break;
}
case 4:
    cout << "input delete point: ";
    cin >> id;
    it = myList.begin();

    advance(it, id);
    myList.erase(it);
    break;
case 5:
    cout << "Number of elements is: " << myList.size() <<
endl;

    break;
default:
    break;
}
} while (choose != 0);
}

```

```

void Map() {
    int choose, id;
    map<int, Country> myMap;
    Country country;
    string name;
    cout << "Map" << endl;
    do {
        cout << "0 - Exit" << endl;
        cout << "1 - Print" << endl;
        cout << "2 - Find element by index" << endl;
        cout << "3 - Add element" << endl;
        cout << "4 - Delete element" << endl;
        cout << "5 - Number of elements" << endl;
        cout << "Choose: ";
        cin >> choose;

        auto it = myMap.begin();
        int k;

        switch (choose)
        {
            case 1:
                for (const auto& el : myMap) {
                    std::cout << el.second << endl;
                }
                break;
            case 2:
                cout << "Input index: ";
                cin >> id;
                cout << myMap.at(id);
                break;
            case 3: {
                cout << "Please, enter key: ";
                cin >> k;
                cout << endl;
                cout << "Please, enter the name of country: ";
                cin >> name;
                cout << endl;
                country.setData(name);
                myMap.emplace(k, country);
                break;
            }
            case 4:
                cout << "input delete point: ";
                cin >> id;
                myMap.erase(id);

```

```

        break;
    case 5:
        cout << "Number of elements is: " << myMap.size() <<
endl;
        break;
    default:
        break;
    }
} while (choose != 0);
}

```

```

int main() {
    int choose;

    do {
        cout << "0 - Exit" << endl;
        cout << "1 - Work with VECTOR" << endl;
        cout << "2 - Work with SET" << endl;
        cout << "3 - Work with LIST" << endl;
        cout << "4 - Work with MAP" << endl;
        cout << "Choose: ";
        cin >> choose;

        switch (choose)
        {
            case 1:
                Vector();
                break;
            case 2:
                Set();
                break;
            case 3: {
                List();
                break;
            }
            case 4:
                Map();
                break;
            default:
                break;
        }
    } while (choose != 0);
}

```

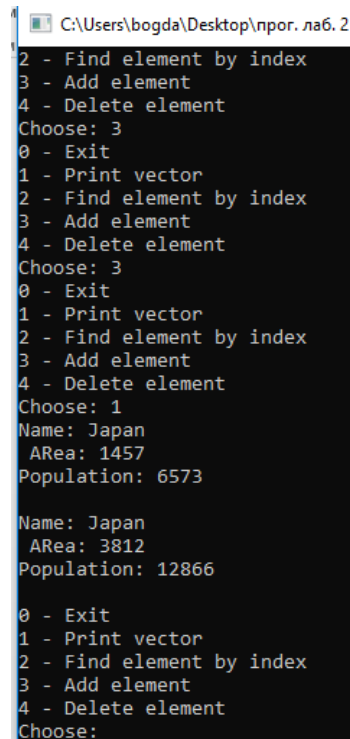


```
        return 0;  
    }
```

3 ВАРІАНТИ ВИКОРИСТАННЯ

3.1 Результат роботи функцій

На рисунку № 3 зображено результат роботи програми



```
C:\Users\bogda\Desktop\пpor. лаб. 2  
2 - Find element by index  
3 - Add element  
4 - Delete element  
Choose: 3  
0 - Exit  
1 - Print vector  
2 - Find element by index  
3 - Add element  
4 - Delete element  
Choose: 3  
0 - Exit  
1 - Print vector  
2 - Find element by index  
3 - Add element  
4 - Delete element  
Choose: 1  
Name: Japan  
Area: 1457  
Population: 6573  
  
Name: Japan  
Area: 3812  
Population: 12866  
  
0 - Exit  
1 - Print vector  
2 - Find element by index  
3 - Add element  
4 - Delete element  
Choose:
```

Рисунок № 3. Результат роботи програми

Висновок: На практиці порівняти STL алгоритми, що не модифікують послідовність.

