

## ЛАБОРАТОРНА РОБОТА №14. СОРТУВАННЯ

*Тема.* STL. Алгоритми зміни послідовності. Сорткування. Функтори.

*Мета.* На практиці порівняти STL алгоритми, що модифікують послідовність.  
Отримати навички роботи з STL функторами.

### 1 ВИМОГИ

#### 1.1 Розробник

- Котенко Сергій Миколайович;
- Студент групи КІТ 102.8(а);
- 09-06-2019р..

#### 1.2 Загальне завдання

Поширити попередню лабораторну роботу, додаючи наступні можливості діалогового меню:

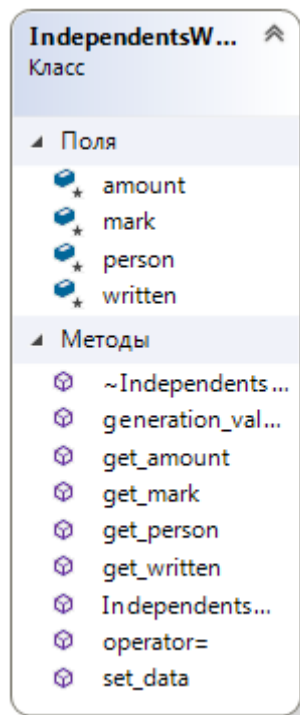
- об'єднання двох STL контейнерів типу vector;
- сортувати заданий контейнер з використання функтора.

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 Опис логічної структури



*Діаграма класу `IndependentsWork`:*

- ✓ ~IndependentsWork - Деструктор класу;
- ✓ generation\_values – Генерація випадкових значень;
- ✓ get\_amount, get\_mark, get\_person, get\_written - Отримання даних;
- ✓ InfoIndependentsWork - Конструктор класу;
- ✓ operator= - Перевантаження оператора;
- ✓ set\_data - Встановлення значень .

## 2.2 Фрагменти коду

```
case 5: {
    sort(container.begin(), container.end(), sort_cont);
    break;
}
case 6: {
    container = ContainerInsert(container, container2);
    break;
}
```

Рисунок 2.1 – Функція сортування та об'єднання контейнерів

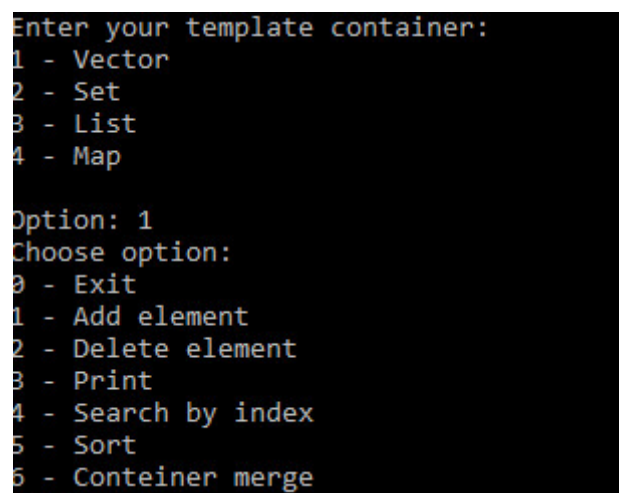
```
template <class T1, class T2>
vector<T1> ContainerInsert(vector<T1> t1, vector<T2> t2) {
    for (const auto& el : t2) {
        t1.insert(t1.end(), el);
    }
    return t1;
}

template<class T>
class Sort_container {
public:
    bool operator() (const T &a, const T &b) {
        return a < b;
    }
};
```

Рисунок 2.2 – Реалізація сортування та об'єднання контейнерів

## 3 ВАРІАНТИ ВИКОРИСТАННЯ

### 3.1 Ілюстрація роботи програми



```
Enter your template container:
1 - Vector
2 - Set
3 - List
4 - Map

Option: 1
Choose option:
0 - Exit
1 - Add element
2 - Delete element
3 - Print
4 - Search by index
5 - Sort
6 - Container merge
```

Рисунок 3.1 – Вибір роботи з контейнером та можливі функції

## ВИСНОВОК

В інтегрованому середовищі *Visual Studio* розроблена програма мовою C++. Виконання програми дозволяє продемонструвати коректність роботи контейнерів.