

## Лабораторна робота №12. STL

*Тема.* STL. Ітератори. Послідовні контейнери. Цикл range-for. Асоціативні контейнери.

*Мета.* Отримати базові знання про STL контейнери. Освоїти основні механізми роботи з STL контейнерами.

### 1. Вимоги

*Розробник :* Кабак О.Р., НТУ “ХПІ”, КІТ102.8а

#### 1.1 Основне завдання

Маючи класи з прикладної області РГЗ (тільки базовий клас та клас/класи спадкоємці), створити діалогове меню, що дозволяє продемонструвати роботу STL контейнерів (додавання / видалення / отримання даних, показ всіх елементів) та показати їх принципову різницю:

vector                  set                  list                  map .

При цьому врахувати, що контейнери містять елементи одного типу, наприклад, базового.

Прохід по всьому контейнеру повинен виконуватися за допомогою циклу мови C++11 – range-for.

*Додаткове завдання на оцінку «відмінно»:*

*контейнери повинні оперувати даними не тільки базового класу, а ще даними класів-спадкоємців.*

#### 1.2 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 для створення та обробки масивів.

```
Quantity of objects in container: 0
What function do you want to cause the list?
0.Exit from program
1.Print Array
2.Delete object from array
3.Add new object to array
4.Access to element by index
->3
```

Рис 1. Початкове меню масивом)

```
Enter type of container
1.Vector
2.Set
3.List
4.Map
1
```

Рис.2 Вигляд вибору наступного кроку(робота з масивом)

```
All elements:
ID:2 Surname:Kabak Works:8 Pages:5 Graphs:3 Arguments:3 Mark5
ID:1 Surname:Someone Works:10 Pages:3 Graphs:3 Arguments:3 Mark5
```

Рис 3. Вивід у консоль усіх студентів котрих я додав у список(двох)

```
Enter index of element for output: 1
ID:1 Surname:Someone Works:10 Pages:3 Graphs:3 Arguments:3 Mark5
```

Рис 4. Результат пошуку за ID

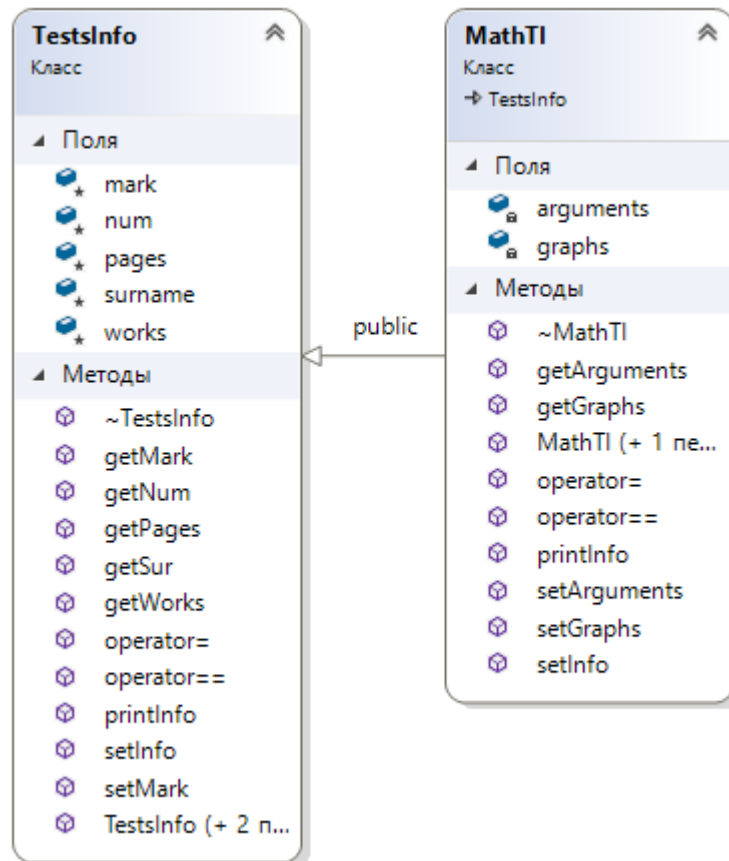


Рис 5. Діаграма класів

```

vector<MathTI> arr;
while (true) {
    cout << "Quantity of objects in container: " << arr.size() << endl;
    cout << "What function do you want to cause the list?" << endl;
    cout << "0.Exit from program" << endl;
    cout << "1.Print Array" << endl;
    cout << "2.Delete object from array" << endl;
    cout << "3.Add new object to array" << endl;
    cout << "4.Access to element by index" << endl;
    cin >> num;
    system("cls");
    switch (num) {
    case 1:
        cout << "All elements: " << endl;
        for (auto it : arr) {
            cout << it;
        }
        system("pause");
        system("cls");
        break;
    case 2:
        cout << "Enter index of element for delete: ";
        cin >> index;
        if (index >= 0 && index < arr.size()) {
            arr.erase(arr.begin() + index);
            system("pause");
        }
        else { ... }
        system("cls");
        break;
    case 3:
        cin >> newObj;
        arr.push_back(newObj);
        system("pause");
        system("cls");
        break;
    case 4:
        cout << "Enter index of element for output: ";
        cin >> index;
        if (index >= 0 && index < arr.size()) {
            newObj = arr[index];
            cout << newObj;
            system("pause");
        }
        else { ... }
        system("cls");
        break;
    case 0:
        arr.clear();
        system("cls");
        return 0;
    }
}

```

Рис 6. Работа з вектором

```

set<MathT> arr;
while (true) {
    cout << "Quantity of objects in container: " << arr.size() << endl;
    cout << "What function do you want to cause the list?" << endl;
    cout << "0.Exit from program" << endl;
    cout << "1.Print Array" << endl;
    cout << "2.Delete object from array" << endl;
    cout << "3.Add new object to array" << endl;
    cin >> num;
    system("cls");
    switch (num) {
    case 1:
        cout << "Elements in container: " << endl;
        for (auto it : arr) {
            cout << it;
        }
        system("pause");
        system("cls");
        break;
    case 2:
        cin >> newObj;
        if (arr.find(newObj) != arr.end()) {
            arr.erase(newObj);
        }
        else {
            cout << "Doesn't exist: " << newObj << endl;
        }
        break;
    case 3:
        cin >> newObj;
        arr.insert(newObj);
        system("pause");
        system("cls");
        break;
    case 0:
        arr.clear();
        system("cls");
        return 0;
    }
}

```

Рис 7. Работа з контейнером Set

```

list<MathTI> arr;
while (true) {
    cout << "Quantity of objects in container: " << arr.size() << endl;
    cout << "What function do you want to cause the list?" << endl;
    cout << "0.Exit from program" << endl;
    cout << "1.Print Array" << endl;
    cout << "2.Delete object from array" << endl;
    cout << "3.Add new object to array" << endl;
    cin >> num;
    system("cls");
    switch (num) {
        case 1:
            cout << "Elements in container: " << endl;
            for (auto it : arr) {
                cout << it;
            }
            system("pause");
            system("cls");
            break;
        case 2:
            if (arr.size() > 0) {
                arr.pop_back();
            }
            system("cls");
            break;
        case 3:
            cin >> newObj;
            arr.push_back(newObj);
            system("pause");
            system("cls");
            break;
        case 0:
            arr.clear();
            system("cls");
            return 0;
    }
}

```

Рис 8. Робота зі списком

```

map<int, MathTI> arr;
while (true) {
    cout << "Quantity of objects in container: " << arr.size() << endl;
    cout << "What function do you want to cause the list?" << endl;
    cout << "0.Exit from program" << endl;
    cout << "1.Print Array" << endl;
    cout << "2.Delete object from array" << endl;
    cout << "3.Add new object to array" << endl;
    cin >> num;
    system("cls");
    switch (num) {
    case 1:
        cout << "Elements in container: " << endl;
        for (const auto& it : arr) {
            cout << "index: " << it.first << endl << it.second << endl;
        }
        system("pause");
        system("cls");
        break;
    case 2:
        cout << "Enter index of element for delete: ";
        cin >> index;
        if (index >= 0 && index < arr.size()) {
            arr.erase(index);
        }
        else {
            cout << "There is no element with this index" << endl;
            system("pause");
        }
        system("cls");
        break;
    case 3:
        cin >> newObj;
        index = arr.size();
        arr.emplace(index, newObj);
        system("pause");
        system("cls");
        break;
    case 0:
        arr.clear();
        system("cls");
        return 0;
    }
}

```

Рис 9. Робота з контейнером Map

## Висновки

В даній лабораторній роботі отримано базові знання про STL контейнери. Освоєно основні механізми роботи з STL контейнерами (vector set list map).