

# Лабораторна робота №23. STL.

- **Вимоги**
- **Розробник**
  - Гладков Костянтин Сергійович
  - Студент групи КІТ-320;
  - 03-march-2021.
- **Основне завдання**

## **Загальне завдання**

Поширити реалізацію лабораторної роботи “Поліморфізм” наступним шляхом:

- замінити масив та CRUD (create/read/update/delete) методи роботи з ним на використання STL
- для предметної галузі з розділу “Розрахункове завдання / Індивідуальні завдання” реалізувати/оновити всі методи роботи з колекцією на використання функцій STL
- додати функцію сортування колекції з використанням функтора
- додати функцію об’єднання двох класів-списків

- **Опис роботи**

Програма працює за принципом успадкування та поліморфізму об'єкта. Таким чином маємо великий функціонал з маленькою навантаженням.

- **Функціональне призначення**

Програма може бути використана для створення сховища з об'єктом, який потрібен вам, а також його подальшим користуванням.

- **Опис логічної структури**

Данна програма працює за принципом створення класу, в якому зберігається динамічний масив іншого класу, а також за допомогою методів, які контролюється цей масив.

Функція main містить в собі виклик функцій, а також корисних властивостей для функцій

class List - клас, який зберігає в собі динамічний масив іншого класу, а також які всередині себе має всі методи роботи з даним класом.

class Agency - базовий клас.

void List::add\_ag - метод, який використовується для розширення нашого масиву, а точніше додавання в даній оновлений масив нового елемента за рахунок спеціального конструктора

void kharkov\_agencies() - Функція знаходить все агенції в харкові

int most\_wins\_law() - метод знаходить агенство з Найбільший числом перемог

void defence\_in\_court() - метод знаходить все агенції із захистом в суді

void no\_weekends\_mar\_agencies() - метод знаходжу агенства, які працюють без вихідних

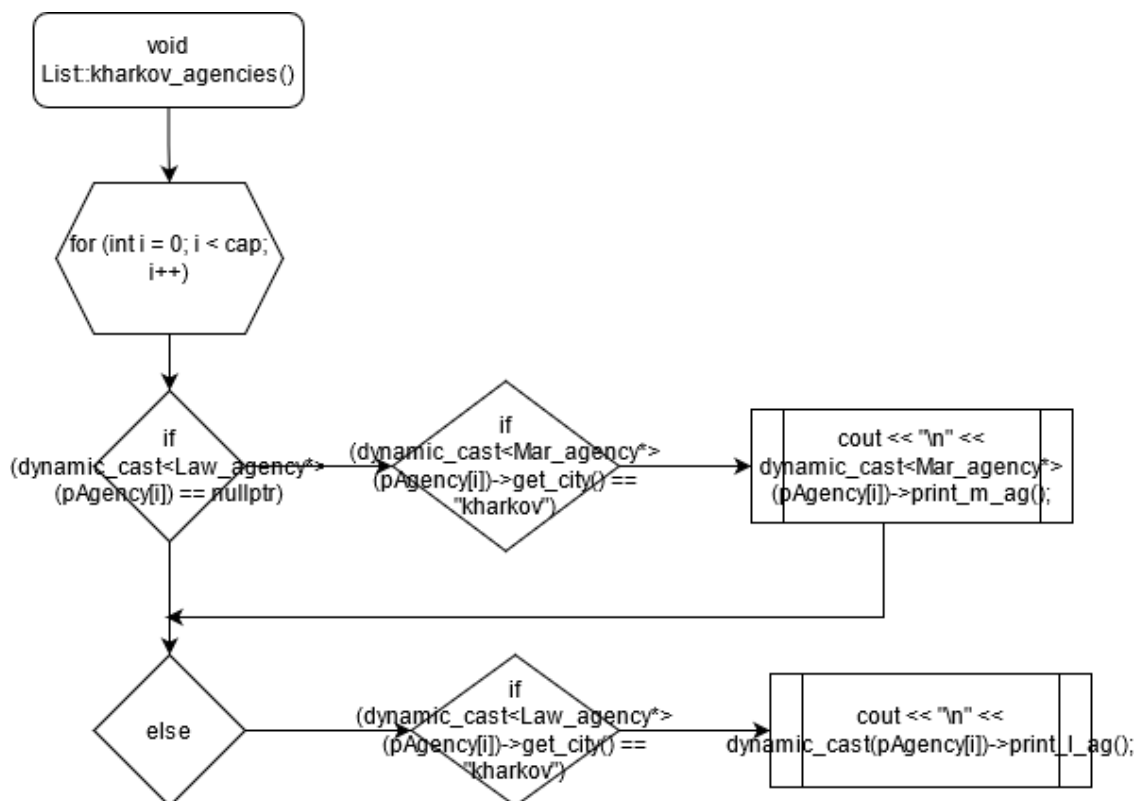


рис. 1 - kharkov\_agencies

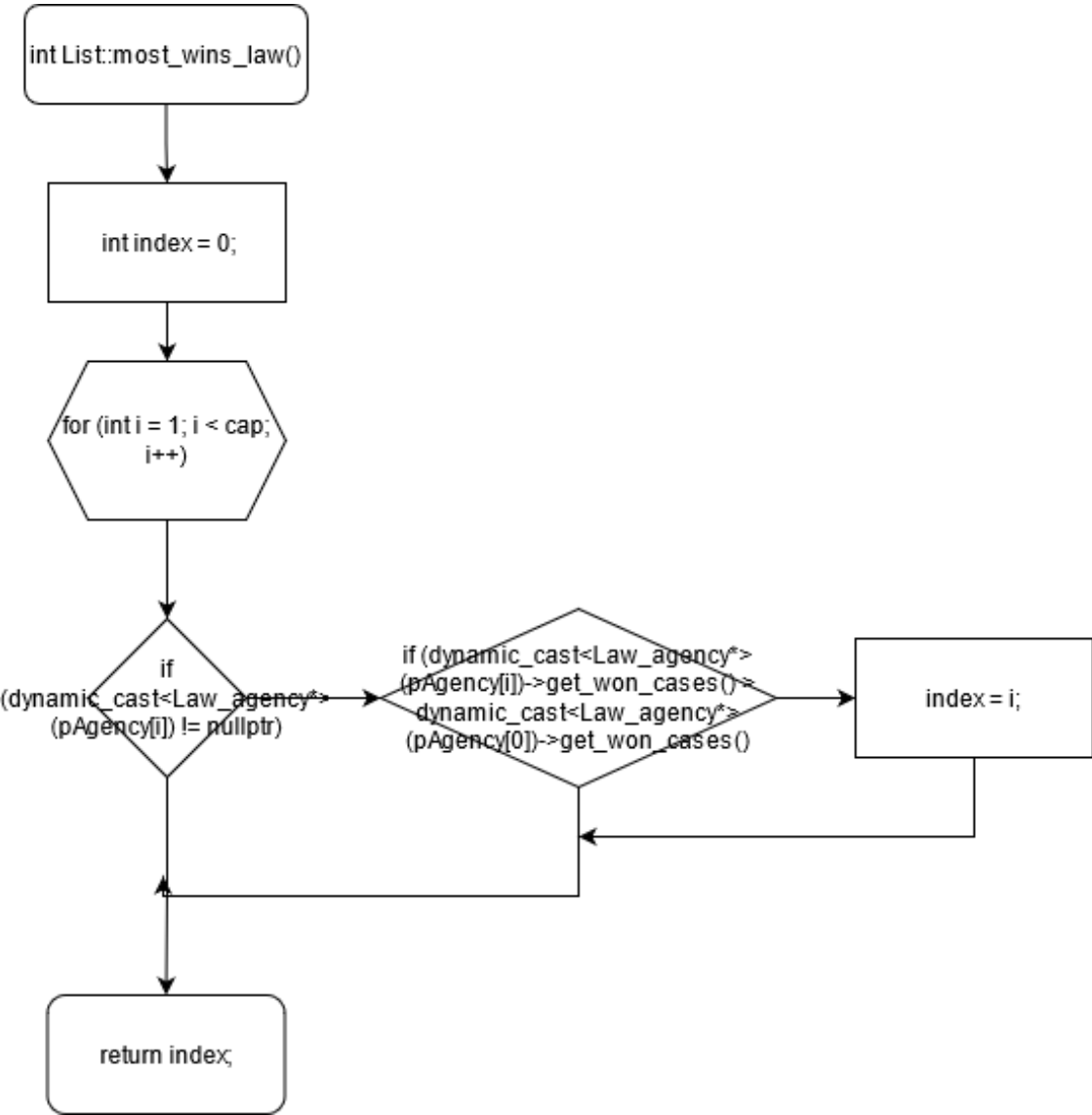


рис. 2 - most\_wins\_law

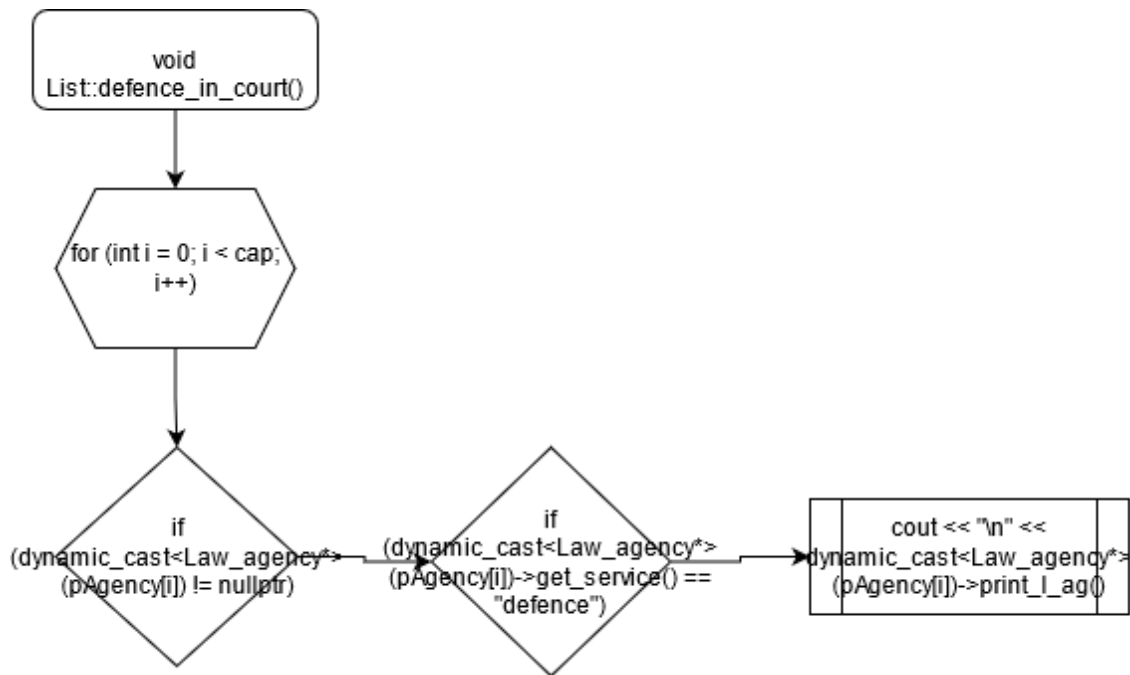


рис. 3 - defence\_in\_court()

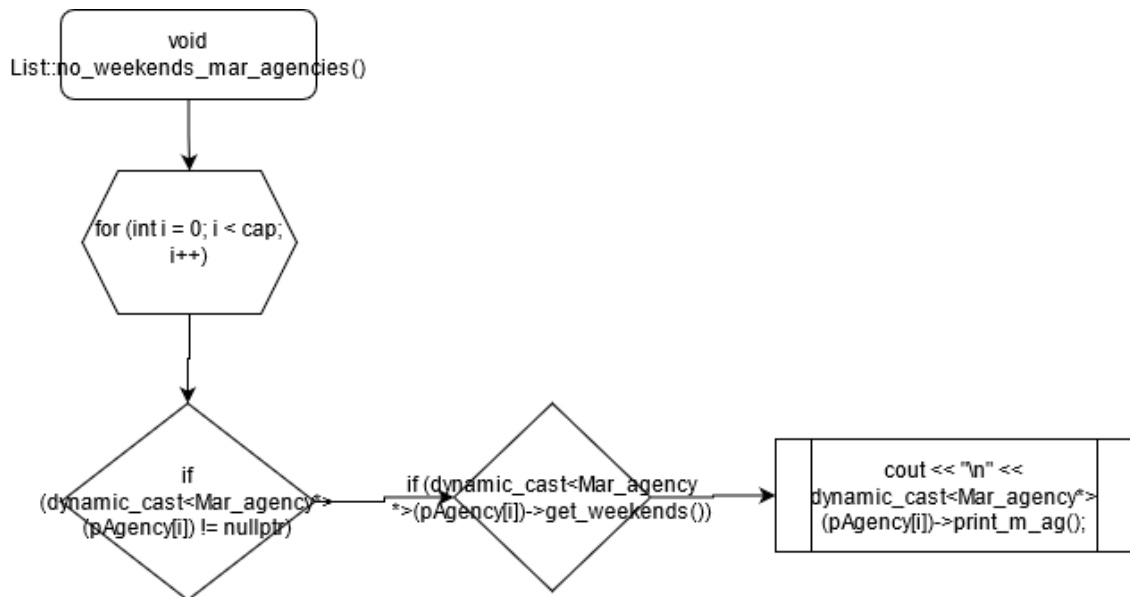


рис. 4 - no\_weekends\_mar\_agencies()

- **Важливі елементи програми**

```

void List::add(Agency *ag){
    this->pAgency[index++] = ag;
}

Agency& List::get_ag(int index) const {

    if (index < 0 || index >= this->num_of_elem){
        cout << "Out of range!";
        exit(1);
    }

    return *this->pAgency[index];
}

int List::get_cap() {
    return cap;
}

class Functor{
public:
    bool operator()(int a, int b){
        return (a < b);
    }
};

class List{
private:
    vector<Agency*>pAgency;

```

- **Варіанти використання**

Дана програма може бути використана для зберігання класів і роботи з ними.

## **Висновки**

У даній лабораторній роботі були придбані знання роботи з STL;