Лабораторна робота №23.
Розумні показчики.

- **Вимоги**
- **Розробник**
  - Гладков Костянтинт Сергіович
  - Студент групи КІТ-320;
  - 01-march-2021.
- **Основне завдання**

Створити STL-контейнер, що містить у собі об'єкти ієрархії класів, використати розумні вказівники:

- auto_ptr;
- unique_ptr;
- shared_ptr;
- weak_ptr.

Додаткове завдання на оцінку "відмінно"

Створити власний розумний вказівник, поданий у вигляді шаблонного класу, який:

- має перевантажений оператор * та -> для отримання фактичного об'єкта та його вказівника;
- дозволяє підраховувати кількість вказівників на об'єкт. Продемонструвати дії, коли виникає інкремент та декремент кількості вказівників;
- контролювати виток пам'яті при виникненні виняткової ситуації.
- продемонструвати відсутність витоків пам'яті при відсутності явного видалення пам'яті за допомогою функцій delete / free.

- **Опис роботи**

Програма працює за тим же принципом, що і колишні, однак використовує ООП, полегшуючи роботу. Також використання розумних покажчиків допомагає змінювати дані

- **Функціональне призначення**

Дана програма може бути використана для маніпуляцій з нашим динамічним масивом, який має в собі класи. Таким чином ми отримуємо

великий функціонал на увазі більше кількості можливих методів для роботи з нашим масивом.

- **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 коструктора

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

void List::change_vals() - змінює дані за допомогою розумних показників

void List::iterate() - итерирование.

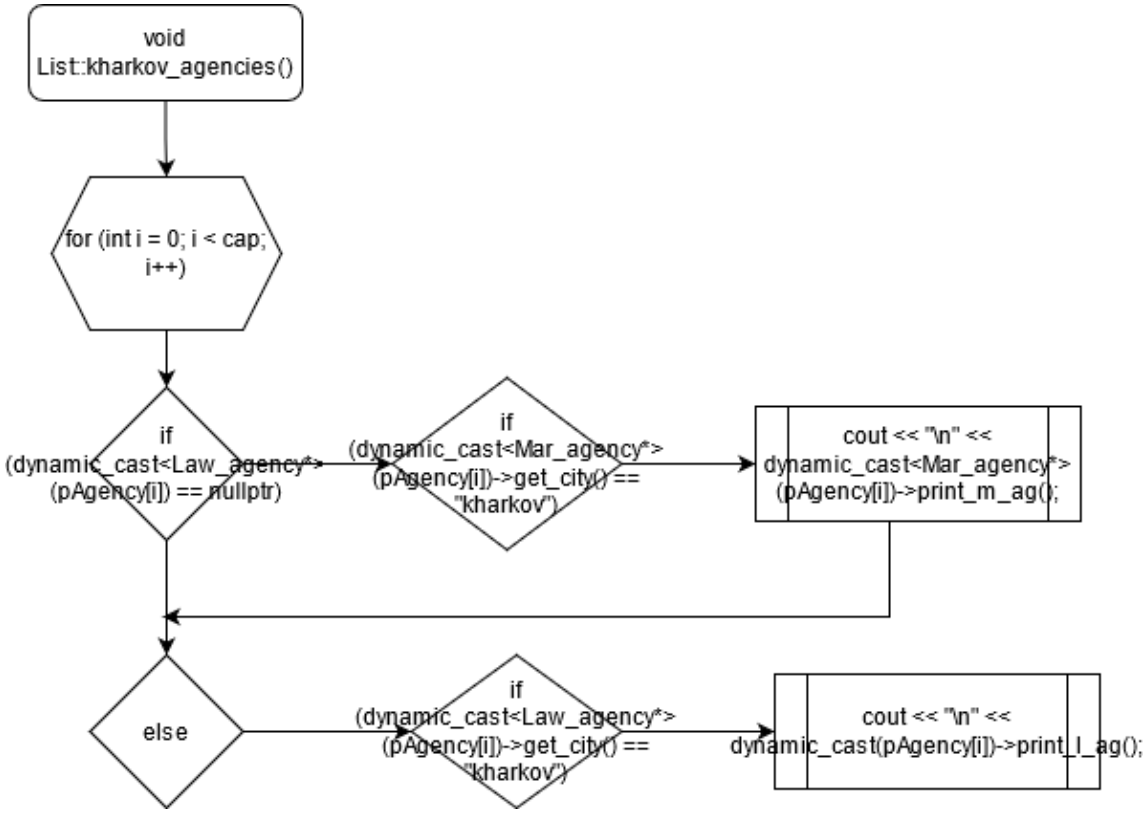

рис. 1 - kharkov_agencies

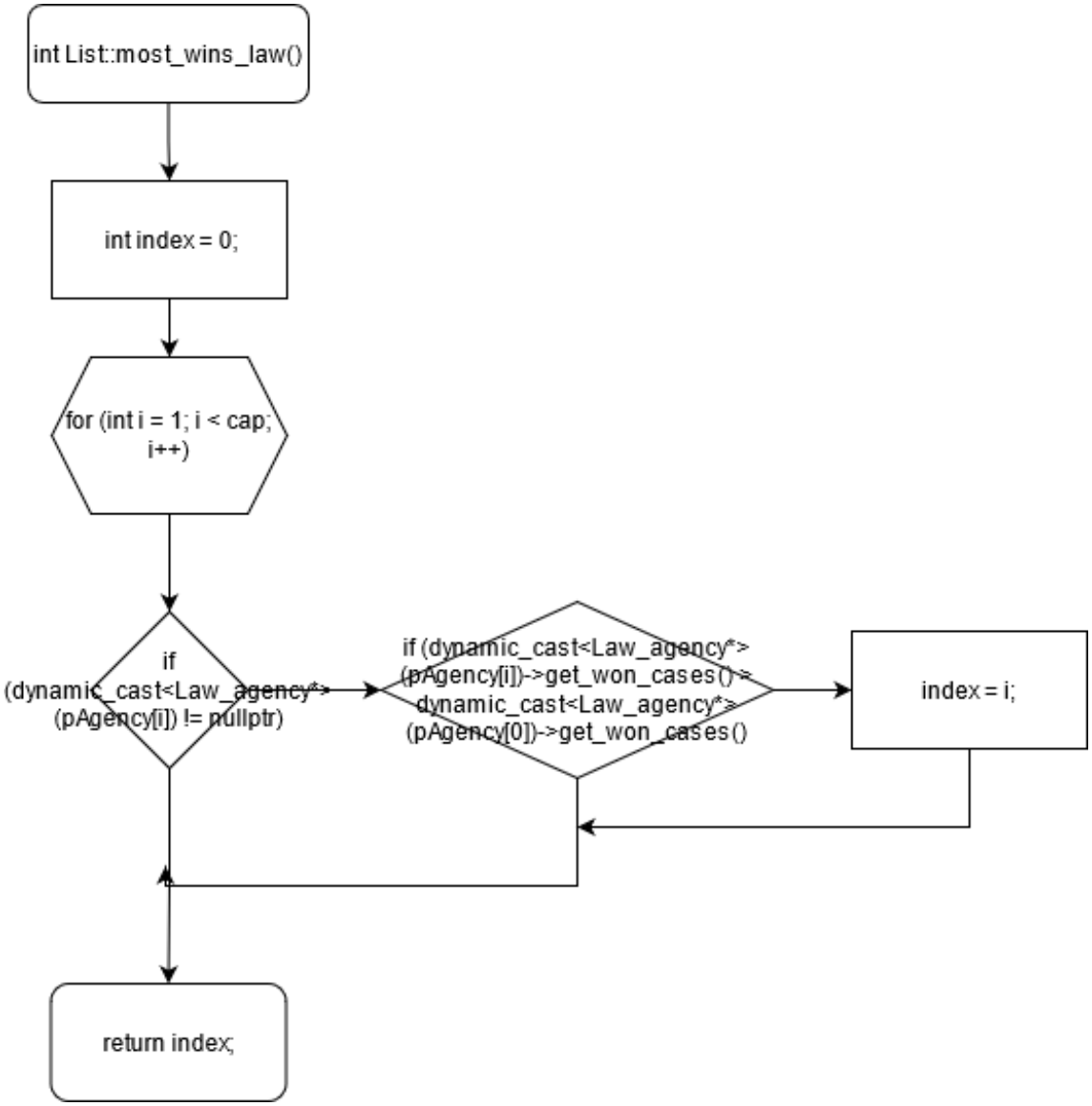

рис. 2 - most_wins_law

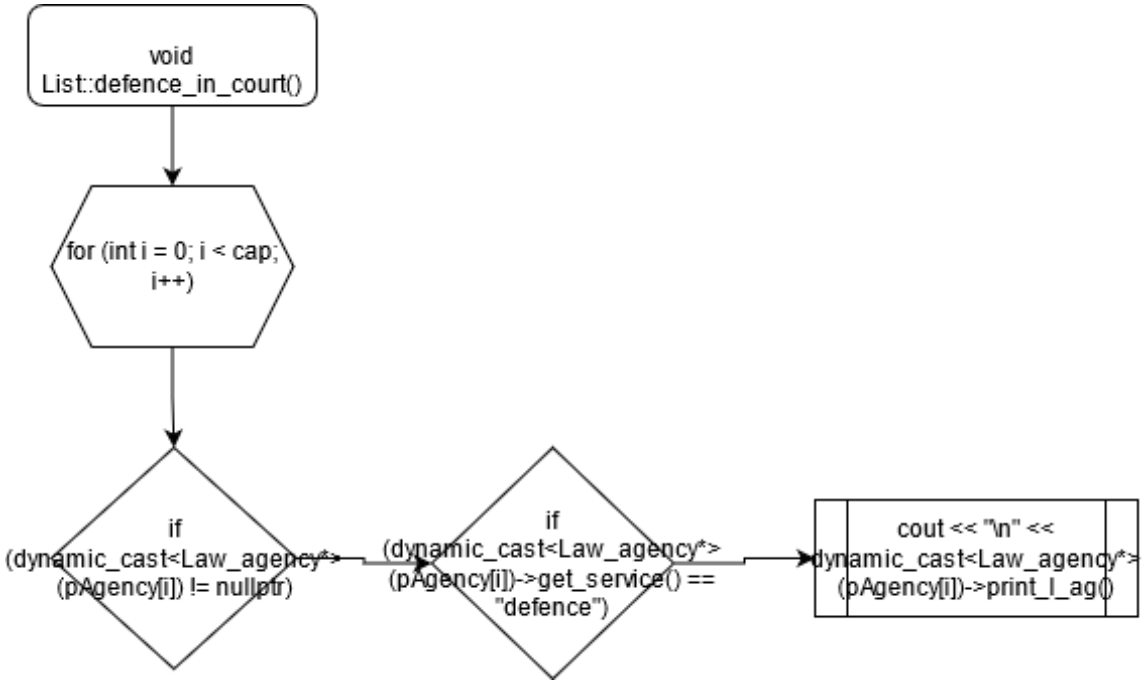

рис. 3 - defence_in_court()

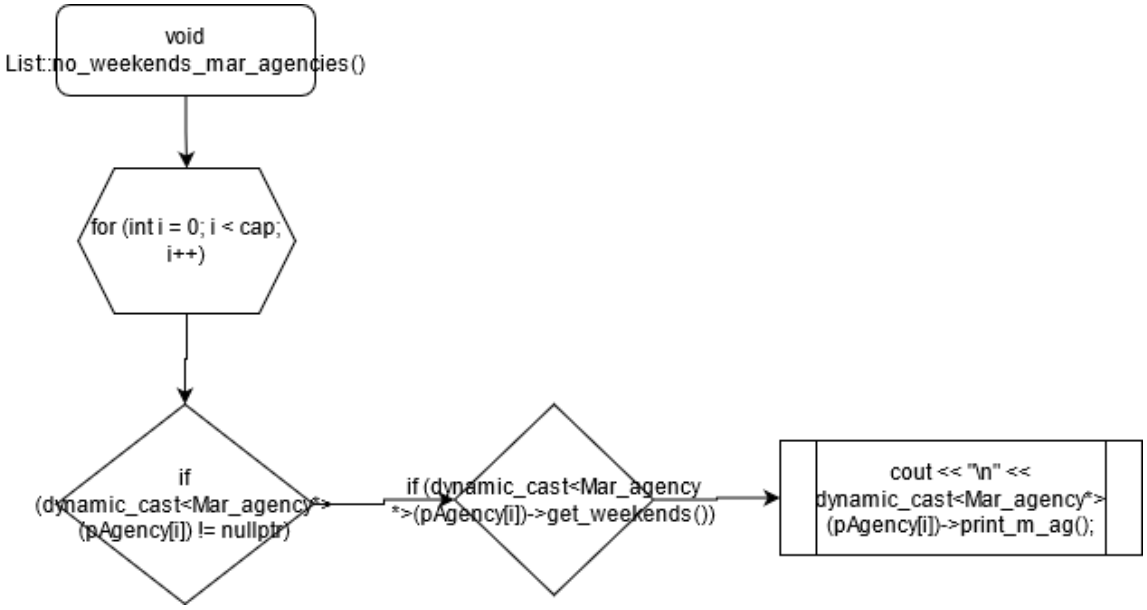

рис. 4 - no_weekends_mar_agencies()

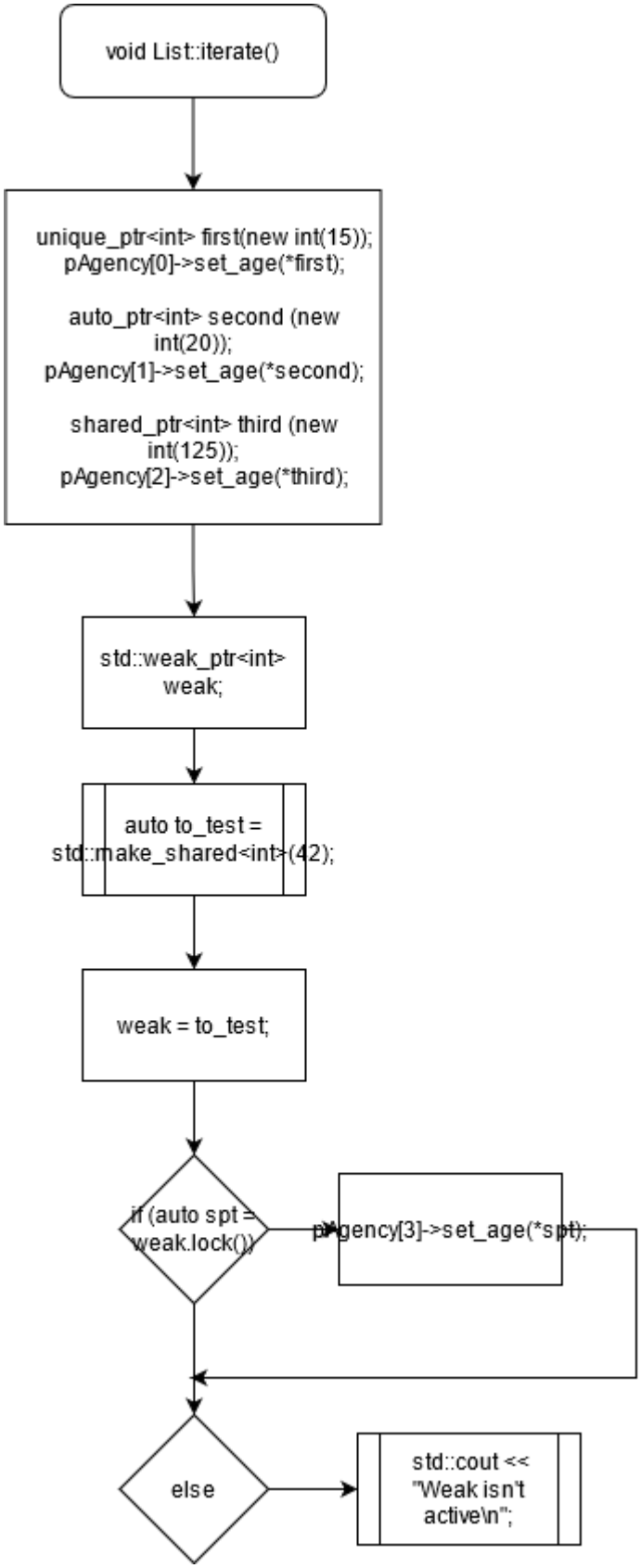

рис. 5 - change_vals

- **Важливі елементи програми**

```
void List::add(Agency *ag){
    pAgency.push_back(ag);
}
```

```
void List::iterate() {

    vector<Agency*>::iterator it;
    for (it = pAgency.begin(); it != pAgency.end(); ++it){
        if (dynamic_cast<Law_agency*>(*it) == nullptr){
            cout << "\n" << dynamic_cast<Mar_agency*>(*it)->print_m_ag();
        }
        else cout << "\n" << dynamic_cast<Law_agency*>(*it)->print_l_ag();
    }
}
```

```
    for (int i = 0; i < pAgency.size(); i++){
        if (dynamic_cast<Law_agency*>(pAgency[i]) == nullptr){
            if (dynamic_cast<Mar_agency*>(pAgency[i])->get_city() == "kharkov"){
                cout << "\n" << dynamic_cast<Mar_agency*>(pAgency[i])->print_m_ag();
            }
        }
        else {
            if (dynamic_cast<Law_agency*>(pAgency[i])->get_city() == "kharkov"){
                cout << "\n" << dynamic_cast<Law_agency*>(pAgency[i])->print_l_ag();
            }
        }
    }
```

```
    for (int i = 0; i < pAgency.size(); i++){
        if (dynamic_cast<Law_agency*>(pAgency[i]) != nullptr){
            if (dynamic_cast<Law_agency*>(pAgency[i])->get_service() == "defence"){
                cout << "\n" << dynamic_cast<Law_agency*>(pAgency[i])->print_l_ag();
            }
        }
        else {
        }
    }
```

```
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 (итератори);