

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

1.1 Розробник

Інформація про розробника:

- Соколенко Д.Г.;
- НТУ "ХПІ", кафедра "ОТП", група 1.KIT 102.8(a);

1.2 Загальне 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.

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

Опис програми дивитись у документації до коду.

2.1 Функціональне призначення

Програма призначена для створення та обробки шаблонних об'єктів.

2.2 Опис логічної структури

Нижче продемонстрована діаграма класів, використаних у програмі з їх полями й методами (рис 2.2.1)

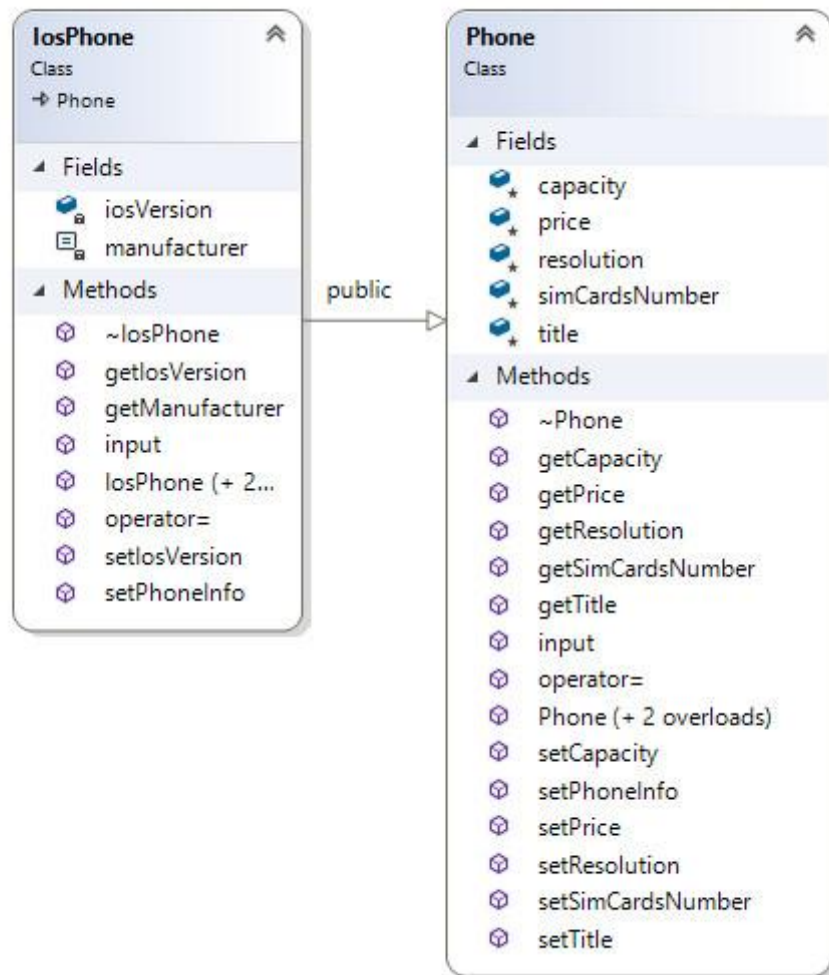


Рис.2.2.1 - Діаграма класів

3 ВАРІАНТИ ВИКОРИСТАННЯ

Нижче (рис.3.1) показані варіанти використання програми.

```

C:\Users\Degster\source\repos\c++ labs\64\D
Index (int) = -1
Lowest (int) = -842150451
#0 -842150451
#1 -842150451
#2 -842150451
Index (Phone) = -1
Lowest (Phone) = | 0 | 0 | 0 | 0
#0 | 0 | 0 | 0 | 0
#1 | 0 | 0 | 0 | 0
#2 | 0 | 0 | 0 | 0
  
```

Рис.3.1 - Робота програми з двома об'єктами-масивами: int та **Phone**.

ВИСНОВКИ

На лабораторній роботі отримані базові знання про STL контейнери, освоєні основні механізми роботи з STL контейнерами.