

Об'єктно орієнтоване програмування

Лабораторна робота №8

Шаблоні класи



Теоретичні відомості

Шаблон (template) — це механізм параметризації визначення класу чи функції довільним типом. Шаблони дозволяють застосовувати простий спосіб введення різного роду загальних концепцій та методів їх спільного використання. Результуючий код за ефективністю не поступається написаному вручну спеціалізованому коду. Клас, що утворюється із шаблону класу у разі конкретного задання значення параметра, є цілком нормальним класом, тому шаблон класу називають генератором типів. Такий процес генерації оголошення класу чи функції за шаблоном та аргументом шаблону називають інстанціюванням. Якщо має місце генерація різних версій шаблону для різних значень параметра, то кожен з цих версій називають спеціалізацією.

Синтаксис визначення шаблону класу наступний:

```
template <class Type> class class_id
{    // визначення класу
};
```

Функції-члени класу автоматично стають шаблонами функцій, хоча для них і не вказується явно службове слово template. У випадку, коли така функція лише декларується у класі, а визначається поза ним, використовується достатньо складний синтаксис:

```
template <class Type> тип_результату_функції
class_id <Type> :: func_id (параметри_функції)
{    // тіло функції
}
```

При створенні екземпляру шаблону класу конкретні значення аргументів шаблону вказуються в кутових дужках:

```
class_id <int> c_i;
class_id <double> c_d;
```

Для даного прикладу буде згенеровано екземпляр c_i класу class_id, в якому в ролі типу Type виступатиме тип int, та екземпляр c_d того самого класу class_id, але із типом double замість типу Type.

Якщо шаблон має не один параметр, а більше, значення аргументів шаблону при створенні екземпляру мають однозначно відповідати списку параметрів шаблону.

Замість службового слова class, що використовується у списку параметрів шаблону, так само можна використовувати його синонім typename.

Варіанти завдань

Завдання: Виконати завдання до лабораторної роботи №4 (зв'язаний список) згідно варіанту з наступними уточненнями:

1. Клас, що описує зв'язаний список має бути реалізовано як шаблонний клас;
2. Реалізувати оператор [] для отримання значення довільного елементу списку.

Додаткове завдання п.3,п.4. За його реалізацію будуть надані додаткові бали до екзамену. Кількість балів залежить від реалізації. Виконання цього пункту не є обов'язковим.

3. Реалізувати оператор «!» для інверсії зв'язаного списку (перший елемент має стати останнім, а останній – першим);
4. Реалізувати оператор «+» для додавання двох зв'язаних списків (елементи другого списку додаються у кінець першого). В результаті операції «+» отримуємо список, який містить всі елементи з першого та другого списку.