Московский Авиационный Институт
(Национальный Исследовательский Университет)

Кафедра: 806 "Вычислительная математика и программирование"
Факультет: "Информационные технологии и прикладная математика"
Дисциплина: "Объектно-ориентированное программирование"

Группа:
Студент: Пашкевич Андрей Романович
Преподаватель: Поповкин Александр Викторович

Москва, 2017

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

Вариант №17

ЦЕЛЬ РАБОТЫ
Целью лабораторной работы является:

- Знакомство с шаблонами классов.
- Построение шаблонов динамических структур данных.

Задание:
Необходимо спроектировать и запрограммировать на языке C++ шаблон класса-контейнера первого уровня, содержащий все три фигуры класса фигуры, согласно вариантов задания (реализованную в ЛР1). Классы должны удовлетворять следующим правилам:

- Требования к классам фигуры аналогичны требованиям из лабораторной работы 1.
- Шаблон класса-контейнера должен соджержать объекты используя std:shared_ptr.
- Шаблон класса-контейнера должен иметь метод по добавлению фигуры в контейнер.
- Шаблон класса-контейнера должен иметь методы по получению фигуры из контейнера (опередеряется структурой контейнера).
- Шаблон класса-контейнера должен иметь метод по удалению фигуры из контейнера (опередеряется структурой контейнера).
- Шаблон класса-контейнера должен иметь перегруженный оператор по выводу контейнера в поток std::ostream (<<)
- Шаблон класса-контейнера должен иметь деструктор, удаляющий все элементы контейнера.
- Классы должны быть расположенны в раздельных файлах: отдельно заголовки (.h), отдельно описание методов (.cpp).

Нельзя использовать:

- Стандартные контейнеры std.

Программа должна позволять:

- Вводить произвольное количество фигур и добавлять их в контейнер.
- Распечатывать содержимое контейнера.
- Удалять фигуры из контейнера.

Описание структуры классов и алгоритма работы программы:

Tree.cpp	
Tree();	Конструктор класса
Tree(std::sharedptr<T> node);	Конструктор класса с заданным узлом
void add(std::sharedptr<T> figure);	Добавление фигуры в дерево
Std::sharedptr<TreeNode>del(std::sharedptr<T> triangle);	Удаление из дерева по параметрам фигуры
bool empty();	Проверка дерева на пустоту
friendstd::ostream& operator<<(std::ostream& os, const Tree& tree);	Перегруженный оператор вывода
virtual ~Tree();	Деструктор класса
TreeNode.cpp	
TreeNode(const std::sharedptr<T>>& figure);	Конструктор Класса
friend std::ostream& operator<<(std::ostream& os, const TreeNode& obj);	Перегруженный оператор ввода
Std::sharedptr< T >SetLeft(std::sharedptr< T > ptr)	Установка ссылки на левый узел
Std::sharedptr<TreeNode> GetLeft();	Получение ссылки на левый узел
Std::sharedptr< T > SetRight(std::sharedptr< T >ptr);	Установка ссылки на правый узел
Std::sharedptr< T > GetRight();	Получение ссылки на правый узел
Std::sharedptr< T > GetFigure() const;	Получение фигуры из узла
virtual ~TreeNode();	Деструктор класса

Шабло́ны — средство языка C++, предназначенное для кодирования обобщённых алгоритмов, без привязки к некоторым параметрам (например, типам данных, размерам буферов, значениям по умолчанию).

Шаблоны позволяют создавать параметризованные классы и функции. Параметром может быть любой тип или значение одного из допустимых типов (целое число, enum, указатель на любой объект с глобально доступным именем, ссылка). Например, нам нужен какой-то класс:

Листинг программы:

Вывод:

В данной лабораторной работе использовалось средство языка C++ шаблоны. Шаблоны позволяют программисту описывать класс не учитывая тип данных, что позволяет расширить область работы программы.