Московский авиационный институт

(национально исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра прикладной математики и информатики

Лаборатория №4 по курсу «Объектно-ориентируемое программирование»

Студент: Рождественских Аднрей Сергеевич

Преподаватель: А. В. Поповкин

Группа: М80-206Б

Вариант: 15

Дата:

Оценка:

Подпись:

Москва, 2018.

1 Цель работы

• Знакомство с шаблонами классов.

• Построение шаблонов динамических структур данных.

2 Задача

Необходимо спроектировать и запрограммировать на языке С++ класс-контейнер первого уровня, содержащий все три фигуры, согласно варианту задания.

Классы должны удовлетворять следующим правилам:

• Требования к классам фигуры аналогичны требованиям из лабораторной работы 1.

• Шаблон класса-контейнера должен соджержать объекты используя std:sharedptr.

• Шаблон класса-контейнера должен иметь метод по добавлению фигуры в контейнер.

• Шаблон класса-контейнера должен иметь методы по получению фигуры из контейнера (опеределяется структурой контейнера).

• Шаблон класса-контейнера должен иметь метод по удалению фигуры из контейнера (опеределяется 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).

3 Описание

Динамические структуры данных используются в тех случаях, когда мы заранее не знаем, сколько памяти необходимо выделить для нашей программы – это выясняется только в процессе работы. В общем случае эта структура представляет собой отдельные элементы, связанные между собой с помощью ссылок. Каждый элемент состоит из двух областей памяти: поля данных и ссылок. Ссылки – это адреса других узлов того же типа, с которыми данный элемент логически связан. При добавлении нового элемента в такую структуру выделяется новый блок памяти и устанавливаются связи этого элемента с уже существующими.

Параметры в функцию могут передаваться одним из следующих способов: по значению и по ссылке. При передаче аргументов по значению компилятор создает временную копию объекта, который должен быть передан, и размещает его в области стековой памяти, предназначенной для хранения локальных объектов. Вызываемая функция оперирует именно с этой копией, не оказывая влияния на оригинал объекта. Прототипы функций, принимающих аргументы по значению, предусматривают в качестве параметров указание типа объекта, а не его адреса. Если же необходимо, чтобы функция модифицировала оригинал объекта, используется передача параметров по ссылке. При этом в функцию передается не сам объект, а только его адрес. Таким образом, все модификации в теле функции переданных ей по ссылке аргументов воздействуют на объект. Использование передачи адреса объекта весьма эффективный способ работы с большим числом данных. Кроме того, так как передается адрес, а не сам объект, существенно экономится стековая память.

4 Изменения

Изменились файлы TbinaryTree.cpp и TbinaryTree.h TbinaryTreeItem.h TbinaryTreeItem.cpp

5 Выводы

В данной работе был спроектирован шаблон класс массива, который позволяет пользователю хранить произвольные типы данных. Было изучено использование шаблонов и принципы их работы.