Бгуир, Кафедра Интеллектуальных Информационных Технологий

Отчёт
по лабораторной работе №3

Выполнил: Якимович Илья Викторович

Группа: 121703

Проверил: Никифоров Сергей Александрович

Минск 2022

Цель: Получить навыки проведения объектно-ориентированного анализа предметной области.

Задание:

В этой лабораторной работе студенту необходимо реализовать шаблон STL-контейнера в соответствие с выбранным вариантом. Реализованныйный шаблон контейнера должен соответствовать следующим требованиям:

· иметь как минимум один шаблонный аргумент, который задает тип элементов, для хранения которых будет использоваться специализация контейнера

· предлагать typedef'ы value_type, reference, const_reference, pointer и др. (полный перечень определяется вариантом задания и разработчиком)

· конструктор по умолчанию (создает пустой контейнер)

· конструктор копирования

· деструктор

· проверка на пустой контейнер (метод empty)

· очистка контейнера (метод clear)

· перегруженный оператор присваивания =

· перегруженные операторы сравнения: ==, !=, >, <, >=, <=

· методы для доступа к элементам контейнера (перечень зависит от варианта задания)

· методы для добавления элементов в контейнер (перечень зависит от варианта задания)

· методы для удаления элементов из контейнера (перечень зависит от варианта задания)

· предлагать классы итераторов для перебора элементов и методы для их создания (тип итераторов и перечень методов зависит от варианта задания)

· перегруженный оператор вывода <<, который использует итераторы контейнера и обобщенный алгоритм std::for_each для вывода элементов в поток

· при возникновении ошибочной ситуации должно выбрасываться исключение

Описание:

Написан класс Graph реализующий структуру графа.

Функция isEmpty() проверка пустой граф или нетт

Функция verticesSize() количество вершин

Функция edgesSize() количество ребер

Функция verticesBegin() итератор на старт вершин

Функция edgesBegin() итератор на старт ребер

Функция adjBegin() итератор на старт

Функция verticesEnd() итератор на конец вершин

Функция edgesEnd() итератор на конец ребер

Функция adjEnd() итератор на конец

Функция atVertices(VertexKey id) получить значение вершины по id

Функция clearGraph() очистить граф

Функция insertVertex(VertexKey id, VertexValue data) вставить вершину

Функция insertEdge(VertexKey v1, VertexKey v2, EdgeValue data) вставить ребро

Функция deleteVertex(VertexKey id) удалить вершину

Функция deleteEdge(VertexKey v1, VertexKey v2) удалить ребро

Функция makeEdgeId(VertexKey v1, VertexKey v2) создать уникальное значение ребра

Вывод:

В ходе работы были изучены некоторые основы ООП.