Стандартная библиотека шаблонов. Итераторы и алгоритмы.

№ урока: 4 **Курс:** C++ Advanced

Средства обучения: Qt Creator

Обзор, цель и назначение урока

Научить студентов понимать и применять на практике итераторы и алгоритмы из стандартной библиотеки шаблонов STL, разобрать такие типы итераторов как ввода, вывода, однонаправленные, двунаправленные и произвольного доступа. Рассмотреть базовые алгоритмы.

Изучив материал данного занятия, учащийся сможет

- Понимать, что такое итератор, уметь работать с различными стандартными итераторами.
- Уметь объяснить разницу между итератором ввода и вывода.
- Понимать, где применяются итераторы произвольного доступа.
- Знать основную сигнатуру и способы применения стандартных алгоритмов.
- Применять нужный алгоритм на практике, исходя из поставленной задачи.

Содержание урока

1. Типы итераторов.
2. Типы алгоритмов.
3. Использование алгоритмов.

Резюме

- Итератор – это обобщение для указателя, умный указатель относительно контейнеров – обеспечивает стандартный интерфейс для доступа к элементам стандартных контейнеров либо своих собственных, которые поддерживают итераторы, без учета того, как устроен и разработан контейнер изнутри и какие данные хранятся внутри данного контейнера.
- Базовые типы итераторов: ввода, вывода, однонаправленные, двунаправленные и произвольного доступа.
- std::advance() - смещение, std::distance() – расстояние.
- (back_\front_\)inserter() – вставка элементов в контейнер по заданному итератору.
- next()\prev() - ++\--
- std::move_iterator – проводит смещение элементов, а не копирование, в случае работы с двумя контейнерами (подробнее о move-семантике смотрите в 7 уроке).
- Немодифицирующие операции над последовательностями - for_each, count_if, find_if, etc.
- Модифицирующие операции над последовательностями – copy, fill, transform, remove, etc.
- Операции разделения - *partition* - stable_partition, partition_copy, etc.
- Операции сортировки (на отсортированных диапазонах) – is_sorted, sort, stable_sort, etc.
- Операции двоичного поиска (на отсортированных диапазонах) – lower_bound, upper_bound, etc.
- Операции над множествами (на отсортированных диапазонах) – merge, set_intersection, etc.
- Операции над кучей – *heap* - make_heap, sort_heap, push_heap, pop_heap, is_heap.
- Операции минимума/максимума – min, max, min_element, max_element, etc.
- Операции сравнения – equal, lexicographical_compare.
- Операции перестановки - *permutation* - (is_\next_\prev_)permutation.

CyberBionic systematics

CyberBionic Systematics ® 2019
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com

- Числовые операции – iota, accumulate, partial_sum, etc.

Закрепление материала

- Какие операторы перегружены в каждом из 5 основных типов итераторов?
- Какая асимптотика выполнения сортировки в STL? Что лежит в основе данной сортировки?
- Как реализован алгоритм copy()?
- Что такое лексикографический порядок?
- Чем отличаются алгоритмы copy() & copy_if(), find() & find_if()?
- Что делают алгоритмы for_each() & transform()?

Дополнительное задание

Задание

Изучите все алгоритмы, которые указаны на сайте https://ru.cppreference.com/w/cpp/algorithm.

Самостоятельная деятельность учащегося

Задание 1
Выучите основные понятия, рассмотренные на уроке.

Задание 2
Реализуйте игру судоку на базе STL контейнеров, итераторов и алгоритмов. Постарайтесь максимально избежать написания собственных «велосипедов». Игра может быть как интерактивной с пользователем, так и подавать на вход заданное расположение цифр и на выходе давать решение, если это возможно.

Задание 3
Зайдите на сайт MSDN.
Используя поисковые механизмы MSDN, найдите самостоятельно описание темы по каждому примеру, который был рассмотрен на уроке, так, как это представлено ниже, в разделе «Рекомендуемые ресурсы», описания данного урока. Сохраните ссылки и дайте им короткое описание.

Рекомендуемые ресурсы

https://ru.cppreference.com/w/cpp/algorithm

https://www.codeproject.com/Articles/854127/Top-Beautiful-Cplusplus-std-Algorithms-Examples

http://cs.stmarys.ca/~porter/csc/ref/stl/iterators.html

https://github.com/caiorss/C-Cpp-Notes/blob/master/STL%20Iterators%20and%20Algorithms.org

https://ru.cppreference.com/w/cpp/iterator

CyberBionic systematics

CyberBionic Systematics ® 2019
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com