

Rvalue-ссылки, семантика перемещения и прямая передача.

№ урока: 7 **Курс:** C++ Advanced

Средства обучения: Qt Creator

Обзор, цель и назначение урока

Научить студентов понимать и применять на практике базовые механизмы семантики перемещения, изучить понятия rvalue & lvalue ссылок, универсальных ссылок, прямой передачи и применение их на практике.

Изучив материал данного занятия, учащийся сможет

- Понимать, что такое семантика перемещения, для чего необходима.
- Уметь объяснить разницу между lvalue & rvalue, rvalue reference & lvalue reference.
- Понимать, что такое прямая передача (perfect forwarding).
- Знать особенности работы `std::move()` в различных практических применениях.
- Знать ключевые правила применимости RVO совместно с семантикой перемещения.

Содержание урока

1. Типы ссылок C++
2. Семантика перемещения
3. Использование прямой передачи

Резюме

gl-value ("generalized" lvalue - "обобщённое" l-value) - это выражение, вычисление которого определяет идентичность (тождество) объекта, битового поля или функции;

pr-value ("pure" rvalue - "чистое" r-value) - это выражение, вычисление которого исчисляет значение операнда оператора (такое pr-value не имеют объекта-результата), либо инициализирует объект или битовое поле (о таком pr-value говорят, что оно имеет объект-результат).

x-value ("eXpiring" value - "истекающее" значение) - это gl-value, обозначает объект или битовое поле, ресурсы которого могут быть вновь использованы;

l-value (названо так исторически) - это gl-value, не является x-value;

r-value (названно так исторически) - это pr-value или x-value.

Lvalue – можно взять адрес переменной.

Rvalue – невозможно взять адрес, служит для временной переменной.

Universal reference (forwarding reference)– T&&, auto&& - в рамках шаблонной функции или auto.

Rvalue reference – MyClass&& - имеет явное определение класса, без вывода типа.

Lvalue reference – MyClass& - обычная ссылка.

При этом сама по себе rvalue reference может быть lvalue (и является им, если она именована).

Свертывание ссы-лок (reference collapsing):

`&& && -> &&`

`&& & -> &`

`& && -> &`

`& & -> &`

Можно (неточно, но образно) сказать, что `rvalue` – это то, брать адрес от чего нельзя, `lvalue` – от чего можно, а `xvalue` – от чего бесполезно.

Временные объекты не могут иметь имя. (Важно!)

`std::move()` – `static_cast` к `Rvalue` reference.

`std::forward()` – тоже самое, только условное приведение, необходимо для `rvalue` и имеет смысл в шаблонных методах.

`std::remove_reference` – обычно используется в шаблонных классах, где нужно убрать любую ссылку.

`std::decay` – `std::remove_reference` + удаляет `cv` квалификаторы.

Также есть аналогичные функторы.

`noexcept` – тесно связан с использованием `std::move()` и перемещаемыми классами.

Передача параметров в том виде, в котором они были получены, без лишних копирований.

`t1, t2` – являются `lvalue` как параметры функции

```
template <typename T1, typename T2> void outer(T1&& t1, T2&& t2)
{
    inner(std::forward<T1>(t1), std::forward<T2>(t2));
}
```

Закрепление материала

- Чем отличается `rvalue` reference от `universal (forwarding) reference`?
- Что такое `std::forward`?
- Для чего необходим `std::move`?
- В каких случаях лучше всего использовать `RVO` & `std::move`?
- Как связаны `noexcept` & `std::move`?

Дополнительное задание

Задание

Изучите особенности `emplace_back` функции, ее поведение с `std::move` и семантикой перемещения.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание 2

Рассмотрите разницу выполнения по времени одного из своих достаточно больших проектов, примените семантику перемещения в тех местах, где это имеет смысл, сделайте замеры по

времени «до - после». Оцените полученные результаты, попробуйте проделать запуск программы без изменений с использованием компилятора C++98 vs C++11.

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/language/value_category

http://thbecker.net/articles/rvalue_references/section_01.html

https://www.ibm.com/support/knowledgecenter/en/SSGH3R_16.1.0/com.ibm.xlcpp161.aix.doc/language_ref/lvalue.html

<https://docs.microsoft.com/en-us/cpp/cpp/move-constructors-and-move-assignment-operators-cpp?view=vs-2017>