УДК 004.423.42

# Move-семантика в языке программирования C++

**Трофимов Д. А., студент**
*Белорусский национальный технический университет*
*Минск, Республика Беларусь*
*Научный руководитель: к. т. н., доцент Дробыш А. А.*

Аннотация.
В статье представлен обзор на move-семантику, появившейся в стандарте C++ 2011 года, в языке программирования C++, показаны ее отличия от семантики копирования, дан обзор на lvalue, rvalue и уникальные ссылки их особенности, а также алгоритмы для работы с ними.

В стандарте C++ 2011 года (C++11) появилось такое понятия как move-семантика или семантика перемещения. Вместе с этим появились такие понятия как lvalue, rvalue и уникальные ссылки, а также алгоритмы перемещения для rvalue и уникальных ссылок: std::move, std::forward.

Move-семантика – это название специализированных средств языка программирования C++, которые предназначены для осуществления перемещения данных во время инициализации и конструирования новых объектов, что позволяет сократить издержки на копирование. Для практического осуществления семантики перемещения в синтаксис C++ введены rvalue ссылки, а также конструкторы перемещения и перемещающийся оператор присваивания.

Понятия lvalue и rvalue были введены в C++ в стандарте 2003 года (C++03). Они представляют из себя определение категорию значений (каждое выражение в C++ имеет два свойства тип и категорию значения). На данный момент существует пять основных категорий значений, но при программировании мы встречаемся только с двумя это rvalue и lvalue. Изначально они произошли от слов right/left value, то есть стоящие справа и слева от выражений, но на данный момент они представляют из себя:

– **lvalue –** проще всего думать, как о функции, объекте или переменной, которая имеет свой адрес памяти. Например, имя переменной, функции или члена-данных, независимо от их типа, даже переменная, имеющая тип rvalue-ссылки, образует выражение *lvalue*;

– **rvalue** – это все, что не является lvalue (временные объекты). Например, литералы, временные объекты или анонимные объекты.

Из-за того, что rvalue является все, что не является lvalue можно преобразовывать lvalue в rvalue, пример представлен на рис. 1, но не наоборот.

```
int a = 1;        // a - lvalue
int b = 2;        // b - lvalue
int c = a + b;    // '+' требует rvalue, поэтому a и b конвертируются в rvalue
                  // и rvalue возвращается в качестве результата
```

Рис. 1. Преобразование из lvalue в rvalue

Позже в C++11 было введено понятия rvalue-ссылки – это ссылки, которые инициализируются только значениями r-values. То есть это ссылки, которые позволяют хранить в себе временные объекты. А стандартные ссылки, которые пришли в C++ из языка программирования C стали называть lvalue-ссылками. Rvalue-ссылки в отличии от lvalue создаются с использованием двойного амперсанда, пример представлен на рис. 2.

```
int x = 7;
int &lref = x; // инициализация ссылки l-value переменной x (значение l-value)
int &&rref = 7; // инициализация ссылки r-value литералом 7 (значение r-value)
```

Рис. 2. Пример создание lvalue и rvalue ссылок

Но двойным амперсандом можно создавать не только, но и универсальные ссылки, которые могут ссылать как на rvalue так и на lvalue значения. То есть выглядеть как rvalue, а вести себя как lvalue ссылки. Такие ссылки называются универсальные и возникают они в двух случаях:

– параметры шаблона функции;

– объявление auto.

Семантика перемещения реализовано с помощью rvalue-ссылок. Для того, чтобы понять ее отличия от семантики копирования да-

вайте взглянем на возможную реализацию алгоритмов из стандартной библиотеки C++ STD которые реализуют семантики копирования и перемещения (std::copy, std::move) представленные в документации, представлены на рис. 3 и 4.

```
template<class InputIt, class OutputIt>
OutputIt copy(InputIt first, InputIt last,
              OutputIt d_first)
{
    for (; first != last; (void)++first, (void)++d_first)
        *d_first = *first;

    return d_first;
}
```

Рис. 3. Реализация алгоритма std::copy

```
template<class InputIt, class OutputIt>
OutputIt move(InputIt first, InputIt last, OutputIt d_first)
{
    for (; first != last; ++d_first, ++first)
        *d_first = std::move(*first);

    return d_first;
}
```

Рис. 4. Реализация алгоритма std::move

Обратите внимание, что на рис. 4 вызываемая функция std::move, не является алгоритмом перемещения, а алгоритмом, имеющим такое же имя, но приводящим наше значение lvalue-ссылки к rvalue-ссылке. Получается, что move-семантика работает с временными объектами и ей не надо копировать его содержимое, а нужно просто изменить адрес.

Таким образом семантика перемещения, введенная в C++11, отличается от семантики копирования тем, что работает с rvalue объектами (временными), из-за чего нам не надо создавать копию объекта. Ее стоит использовать, когда у нас запрещено использовать семантику копирования или если надо, чтобы в исходном объекте ничего не оставалось.

## Список использованных источников

1. Meyers, S. Effective Modern C++ / S. Meyers // O'Reilly Media. – 2015. – P. 157–168.

2. O'Dwyer, A. Mastering the C++17 STL / A. O'Dwyer // Packt Publishing. – 2017. – P. 42–49.

3. Value categories [Electronic recourse]. – Mode of access: https://en.cppreference.com/w/cpp/language/value_category. – Date of access: 15.03.2023.

УДК 004.65

## Сравнение PostgreSQL и Oracle Database

**Трофимов Д. А., студент,**
**Вагин Д. И., студент**
*Белорусский национальный технический университет*
*Минск, Республика Беларусь*
*Научный руководитель: преподаватель Михасик Е. И.*

Аннотация.

В статье представлена обзор двух систем управления базами данных (СУБД) PostgreSQL и Oracle Database, их история, характеристики, преимущества и недостатки.

PostgreSQL, или просто postgres, – объектно-реляционная система управления базами данных с открытым исходным кодом. На первом месте в ней стоит расширяемость, техническое совершенство и совместимость. Она конкурирует с основными базами данными: Oracle, MySQL, SQL Server и другими. Используется в самых разных секторах, включая государственные учреждения, открытые и коммерческие продукты. Это кросс-платформенная СУБД, работающая в большинстве современных операционных систем, в т. ч. Windows, macOS и различных дистрибутивах Linux. Совместима со стандартами SQL и обладает всеми свойствами взаимодействия с ACID.

СУБД PostgreSQL начиналась как исследовательский проект в Калифорнийском университете в Беркли от некоммерческой СУБД Postgres.

Стоунбрейкер и его студенты разрабатывали новую СУБД в течение восьми лет, с 1986 по 1994 год. За этот период в синтаксис были введены процедуры, правила, пользовательские типы и многие другие компоненты.

Стоунбрейкер и его студенты разрабатывали новую СУБД в течение восьми лет, с 1986 по 1994 год. За этот период в синтаксис были