# Основные термины и фразы в Си

### Лексемы

Лексема (token) - символ или последовательность символов. Это всё то, из чего состоит программа.

“По завершении работы препроцессора программа разбивается на лексемы: идентификаторы, ключевые слова, константы, строковые литералы, знаки операций и прочие разделители.”.

“Прочие разделители - разделители лексем: горизонтальные таб, вертик. таб, символ заверш. строки, комментарии, пробелы и т.д. Они отбрасываются (**cast aside**) компилятором.”.

“Расширенные наборы символов для азиатских стран: строковая константа - L’...’, строковый литерал L “...“.”.

Литерал (literal) - фиксированное значение, которое не может быть изменено программой. Пример: **ch = ‘r’**.

### Объекты и типы

Переменная (объект) - именованный участок в памяти. “Интерпретировать её в программе можно в зависимости от:

1. Класса памяти - определяет время жизни и контекст объекта.
2. Типа данных - смысл вкладываемый в данные объекта.”.

Классы памяти в Си по K&R: автоматический (auto & register) и статический (static & extern) + khizha: динамический класс памяти.

Расширение целочисленного типа (integer promotion) - итерация приведения операнда к int (или unsigned int) если тип int (unsigned int) позволяет *представить все значения исходного типа операнда*. char, short, int, integer bit field, все либо signed либо нет, а также объект enum типа - все могут применяться в выражениях, где разрешены int. Это возможно из-за IP. По сути, после компиляции IP и происходит для всех вышеперечисленных типов.

Конгруэнтный тип - имеющий тоже двоичное представление.

“Объекты типа char имеют наименее строгие требования к выравниванию. Выравнивание используется когда нужно поместить в адреса очень требовательный тип! Выравнивание регулируется с помощью **typedef** и **union**.”.

“Приведение типов применяется к указателям явным образом и даже приспособлено к неудачному системному интерфейсу”.

“Преобразования указатель в указатель требуют явного приведения типов!”.

“Значение объекта типа void нельзя использовать (x = void f), явно или неявно привести к другому типу. Однако, можно приводить указатели на объекты к void \* без потери информации. void \* можно также употреблять совместно с указателями любого типа в = и <= как угодно. ”.

Неполный тип -

### Выражения

Выражение - композиция операторов (символов типа “+, -”), переменных и литералов. “Выражение может иметь значение (например, выражение присваивания). Такие выражения могут участвовать в других выражениях, так как их значения будут использоваться.”.

Выражения в С++ могут быть двух типов: lvalue и rvalue.

lvalue (в Си устаревшее: left-side value (именующее выражение), в C++: locator value) - объект (переменная), который занимает идентифицированное место в памяти или это выражение, обозначающее объект или ссылающееся на него. Пример: **int a; // a - lvalue, у а есть адресс**.

“Идентификатор - lvalue если он ссылается на объект и если его тип арифметический, struct, union или указатель.”

rvalue - всё, что не является lvalue. rvalue - это временное значение, которое не сохранится за пределами выражения, которое его использует. Пример: **3 // 3 - это просто значение.**

modifiable lvalue - модифицированное lvalue: не массив, не неполный тип, не имеет спецификатора const, не является struct или union с полями const.