**Контрольные вопросы:**

**1. Назовите последовательность обработки исходного текста транслятором.**

Лексический анализ (сканер), синтаксический анализ (разбор), семантический анализ.

**2, 3. Лексический анализ: определение, назначение, применение, место в структуре транслятора, входная и выходная информация:**

Первая стадия работы компилятора называется лексическим анализом, а программа, её реализующая, – лексическим анализатором (сканером). На вход лексического анализатора подаётся последовательность символов входного языка. Лексический анализатор выделяет в этой последовательности простейшие конструкции языка, которые называют лексическими единицами. Примеры лексических единиц: идентификаторы, числа, символы операций, служебные слова и т.д. Лексический анализатор преобразует исходный текст, заменяя лексические единицы их внутренним представлением – лексемами, для создания промежуточного представления исходной программы. Каждой лексеме сопоставляется ее тип и запись в таблице идентификаторов, в которой хранится дополнительная информация. Таблица лексем (ТЛ) и таблица идентификаторов (ТИ) являются входом для следующей фазы компилятора – синтаксического анализа (разбора, парсера)

**4. Назначение и принципы работы лексических анализаторов.**

Назначение лексического анализатора:

− лексический анализ уменьшает длину программы, устраняя из ее исходного представления несущественные пробелы и комментарии;

− замена в программе идентификаторов, констант, разделителей, ключевых слов лексемами делает представление программы более удобным для дальнейшей обработки;

− упрощает разработку (для выделения и анализа лексем применяются эффективные и простые алгоритмы разбора);

− избавляет синтаксический анализатор от работы с текстом исходный программы.

**5. Какие функции выполняет лексический анализатор?**

− удаление лишних пробелов и комментариев;

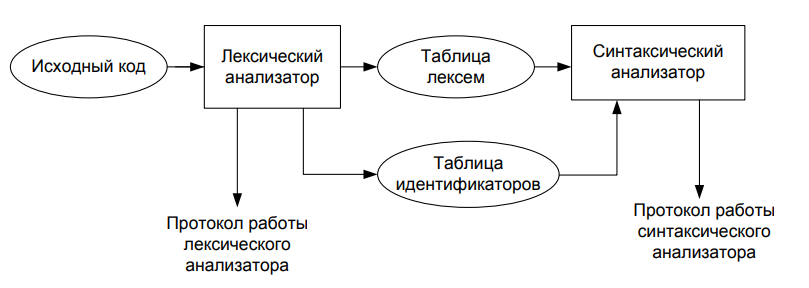
− распознавание идентификаторов, ключевых слов, констант, сепараторов и знаков операций.

**6. Что является результатом успешного лексического разбора исходного текста?**

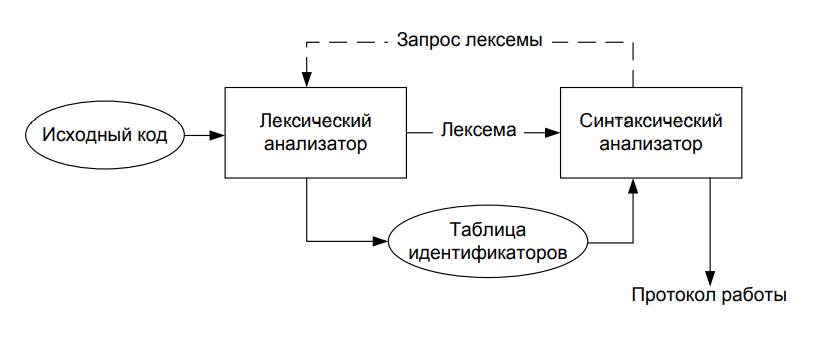
Таблица лексем и таблица идентификаторов.

**7. Лексический анализатор: способы взаимодействия с синтаксическим анализатором.**

Последовательный:

****

Параллельный:

****

**8. Таблица идентификаторов и область видимости конкретного идентификатора:**

Таблица идентификаторов – таблица, в которой в единичном экземпляре хранятся все идентификаторы и информация о том, где был их первый вход, чтобы знать, куда ссылаться при выполнении операций с ними.

Определение границ лексем — это выделение тех строк в общем потоке входных символов, для которых надо выполнять распознавание. Для простейших входных языков границы лексем распознаются по заданным терминальным символам (пробелы, скобки, знаки операций, символы комментариев, а также разделители (запятые, точки с запятой и др.) [они тоже являются лексемами, их важно не пропустить при разборе]).

**9. Ошибки, определяемые лексическим анализатором.**

Если в процессе работы лексический анализатор не смог правильно определить тип лексемы, считается, что программа содержит ошибку. Информация об ошибке с указанием строки и позиции выдается пользователю (в консоль и сохраняется в протоколе). На уровне лексического анализатора определяются только некоторые ошибки, поскольку лексический анализатор рассматривает исходный текст программы в ограниченном контексте.

**10. Какие грамматики применяются для описания лексики языка программирования?**

Для описания лексики языка программирования обычно применяются регулярные грамматики.

Регулярный язык – все правила этого языка имеют вид:

Н порождает Т (для обоих видов регулярных грамматик)  
и  
Н порождает НТ (левосторонняя регулярная грамматика) или

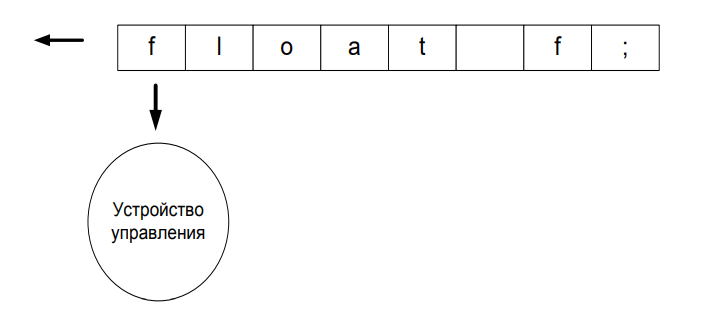
Н порождает ТН (правосторонняя)  
где Т – терминал (символ a-z, 0-9), Н – нетерминал (сущность языка: формула, выражение, команда (понятная только для компьютера)), НТ — цепочка символов, где в начале идут только нетерминалы, а после них – только терминалы.

Язык программирования (на уровне лексического анализа) представляет собой регулярный язык, заданный регулярными выражениеми (язык типа 3 иерархии Хомского).

**11.Простейший распознаватель регулярного множества:**

Простейший распознаватель состоит из входной ленты (входная цепочка символов), и управляющего устройства с конечной памятью. Класс алгоритмов, соответствующих приведенной схеме, может быть записан в форме конечного автомата (KA). Регулярные выражения описывают регулярные множества. Для распознавания регулярных множеств служат конечные автоматы.

Лексический анализатор можно создать на базе регулярной грамматики, и построить эквивалентный ДКА, что снизить сложность разбора до O(N), где N — длина строки.



**12.Алгоритм разбора цепочки символов (алгоритм сканера):**

• сканер проверяет входной поток символов программы на исходном языке на допустимость, удаляет лишние пробелы и комментарии и добавляет сепаратор для вычисления номера строки для каждой лексемы;

• для выделенной части входного потока выполняется функция распознавания лексемы;

• при успешном распознавании информация о выделенной лексеме заносится в таблицу лексем и таблицу идентификаторов, и алгоритм возвращается к первому этапу;

• формирует протокол работы;

• при неуспешном распознавании выдается сообщение об ошибке, а дальнейшие действия зависят от реализации сканера - либо его выполнение прекращается, либо делается попытка распознать следующую лексему (идет возврат к первому этапу алгоритма).