

	<p><i>1.3. Элементы теории трансляции. Общие принципы организации синтаксического разбора. Семантический анализ.</i> Обобщенная структура транслятора. Назначение и необходимость фазы лексического анализа. Методы лексического анализа. Грамматики и распознаватели для лексического анализа. Лексический анализ на основе конечных автоматов. Преобразователи с магазинной памятью. Схемы синтаксически управляемого перехода. Описание областей видимости и блочной структуры. Линеаризованные представления. Атрибутная схема для алгоритма сопоставления образцов. Назначение синтаксического разбора. Классификация методов синтаксического разбора. Последовательность разбора. Использование просмотра вперед. Использование возвратов. Применение автоматов с магазинной памятью для нисходящего разбора слева направо. Общая связь между грамматиками и автоматами с магазинной памятью. Программная реализация нисходящего автомата с магазинной памятью.</p> <p>Назначение семантического анализа. Семантическая модель программы. Построение элементов таблицы имен. Синтаксически управляемый перевод. Использование синтаксически управляемого перевода при построении таблиц имен. Использование синтаксически управляемого перевода при работе с таблицей имен. Синтаксически управляемый перевод при генерации промежуточного Представления. Организация промежуточного представления программы.</p> <p><i>1.4. Генерация кода. Системы автоматизации построения трансляторов.</i> Генерация кода, этапы, основные понятия. Динамическая организация памяти. Назначение адресов, трансляция переменных и выражений. Трансляция объектно-ориентированных свойств языков программирования. Генерация оптимального кода методами синтаксического анализа. Выбор дерева вывода наименьшей стоимости.</p> <p>Структуры систем, основные термины и определения и части программного кода реализации систем автоматизации построения трансляторов. Система СУПЕР, система Yacc, LEX. Разделы типов, констант, файлов, библиотек. Атрибутная схема.</p>
2	<p>лабораторные работы 4 шт. по 2 часа:</p> <p><i>2.1. Алгоритмы построения конечных автоматов.</i> Построение Машины Тьюринга, Марковских моделей. Составление отчета.</p> <p><i>2.2. Построение лексического анализатора.</i> Формальные грамматики, задачи. Построение лексического анализатора. Полный список выбранных ключевых слов и стандартных функций. Внутренние таблицы сканера. Отладочные примеры работы сканера с выходными таблицами и дескрипторным текстом.</p> <p><i>2.3. Построение распознавателя. Использование таблицы имен.</i> Синтаксический анализ, группирование лексем исходной программы в грамматические фразы, используемые компилятором для синтеза вывода. Реализация дерева разбора, синтаксическое дерево. Алгоритм устранения недостижимых символов. Алгоритм устранения бесполезных символов. Алгоритм преобразования грамматики в грамматику без пустых продукций. Алгоритм устранения цепных продукций. Устранение в КС-грамматике левой рекурсии. Детерминированный синтаксический анализ сверху вниз.</p> <p><i>2.4. Семантический анализ и генерация промежуточного представления.</i> Внутренние формы представления программы. Постфиксная запись. Представление генерируемого кода в форме четверок.</p>



	Организация генератора кода. Методы управления памятью. Фазы управления памятью.
3	практические занятия не предусмотрены в структуре дисциплины
4	<p>В процессе курсового проектирования, студенты разрабатывают программный комплекс на языке высокого уровня по свободному выбору. Целью работы является освоение методологий проектирования и реализации трансляторов программных языков на основе формальных грамматик и конечных автоматов.</p> <p>Разработка программного комплекса направлена на приобретение студентом практических навыков анализа задачи, ее декомпозиции, работы с различными структурами данных, типовыми алгоритмами, подпрограммами, стандартными модулями, сетевыми компонентами, средствами операционной системы.</p> <p>Студент должен разобраться с поставленной задачей, отладить ее на персональном компьютере, продемонстрировать преподавателю работу отлаженных программ в соответствии с вариантом задания и выполнить отчет по работе в соответствии с требованиями.</p> <p>При выполнении расчетного задания студент должен обеспечить выполнение следующих требований:</p> <ol style="list-style-type: none">1. Все решаемые в системе задачи должны быть реализованы в виде отдельных модулей.2. Каждая подпрограмма/модуль должен начинаться с комментария, объясняющего его назначение, назначение переменных, передаваемых в него и из него.3. В работе должен быть предусмотрен «дружественный» пользовательский интерфейс, организовано меню решаемых задач и вызов подпрограмм по выбору пункта меню.4. Ввод данных должен сопровождаться проверкой корректности.5. Реализация разработанной системы должна начинаться с названия программы, информации об ее назначении, группы, фамилии и имени автора разработки. <p>В ходе выполнения работы для каждого задания необходимо выполнить по соответствующему номеру варианта задания. Номер варианта курсового проекта N выбирается в соответствии с порядковым номером фамилии студента в журнале группы. Номер N используется при конкретизации заданий.</p>
5	расчетно-графическая работа не предусмотрена в структуре дисциплины
6	<p>Самостоятельная работа студентов:</p> <ol style="list-style-type: none">6.1. 2 контрольных опроса после 2-й и 4-й лекций;6.2. Закрепление материала по тематике лекционных занятий: закрепление изучения материалов лекций 1.1-1.4 – основы программирования на языке высокого уровня; классификация программного обеспечения и формальных грамматик; проектирование транслятора языка высокого уровня.6.3. Подготовка к экзамену по дисциплине (оценочные материалы приведены в разделе 6 настоящей РПД).