**Теоретические вопросы**

* Этапы жизненного цикла программного продукта.
* Состав и схема функционирования классической системы программирования.
* Типы трансляторов, особенности интерпретаторов и компиляторов. Смешанная стратегия трансляции.
* Общая схема работы компилятора.
* Основные понятия теории формальных грамматик и языков.
* Эквивалентные грамматики.
* Классификация формальных грамматик и языков по Хомскому.
* Соотношения между типами грамматик.
* Соотношения между типами языков.
* Разбор цепочек по КС-грамматикам. Задача разбора. Дерево вывода.
* Неоднозначность грамматик и языков.
* Недостижимые и бесплодные символы грамматики. Алгоритмы удаления недостижимых и бесплодных символов. Приведенная грамматика. Алгоритм приведения грамматики.
* Алгоритм удаления правил с пустой правой частью.
* Определение недетерминированного конечного автомата (НКА).
* Диаграмма состояний (ДС) конечного автомата.
* Леволинейные регулярные грамматики и конечные автоматы.
* Определение детерминированного конечного автомата (ДКА).
* Алгоритм построения детерминированного конечного автомата по НКА.
* Задачи лексического анализа.
* Лексический анализ на основе регулярных грамматик.
* Задачи синтаксического анализа.
* Метод рекурсивного спуска (РС-метод): назначение, семантика процедур метода рекурсивного спуска.
* Достаточные условия применимости метода рекурсивного спуска для грамматик без ε-альтернатив и для грамматик с ε-альтернативами.
* Критерий применимости РС-метода.
* Задачи семантического анализа. Грамматики с действиями.
* Свойства языка внутреннего представления программы, примеры таких языков.
* Синтаксически управляемый перевод: идея, принципы организации, примеры. Определение формального перевода.
* ПОЛИЗ выражений.
* ПОЛИЗ операторов языков программирования.
* Генерация ПОЛИЗа выражений и операторов.
* Интерпретация ПОЛИЗа.
* Основные стратегии распределения памяти.
* Машинно-независимая и машинно-зависимая оптимизация. Примеры оптимизирующих преобразований.
* Интегрированная среда разработки программного обеспечения (ИСР).
* Основные функции редактора текстов в рамках ИСР. Примеры его интегрированности с другими компонентами ИСР.
* Отладчики, их возможности. Примеры интегрированности отладчика с другими компонентами ИСР.
* Редактор внешних связей, его назначение и принципы работы. Загрузчик.
* Профилировщики. Назначение. Принцип работы.
* Библиотеки. Основные типы библиотек.
* Критерии проектирования стандартных библиотек.
* Способы подключения библиотек функций.

**Практические навыки**

* Определение типа по Хомскому заданной грамматики.
* Определение типа языка, порождаемого заданной грамматикой.
* Построение грамматики [определенного типа], порождающей заданный язык.
* Определение языка, порождаемого заданной грамматикой.
* Построение дерева вывода (разбора) заданной цепочки по заданной КС-грамматике.
* Построение приведенной грамматики.
* Устранение пустых правых частей заданной КС-грамматики.
* Построение ДС конечного автомата по заданной леволинейной грамматике.
* Построение анализатора на C++ по заданной ДС.
* Восстановление леволинейной регулярной грамматики, порождающей язык, распознаваемый конечным автоматом, заданным в виде ДС.
* Восстановление леволинейной регулярной грамматики по заданному тексту анализатора на C++.
* Построение ДС по праволинейной грамматике и построение праволинейной грамматики по заданной ДС.
* Преобразование леволинейной грамматики в эквивалентную праволинейную с помощью ДС.
* Преобразование НКА в эквивалентный ДКА с помощью соответствующего алгоритма.
* Определение и обоснование применимости метода рекурсивного спуска к заданной КС-грамматике.
* Построение (на C++) анализатора методом рекурсивного спуска для заданной КС-грамматики.
* Восстановление КС-грамматики по заданному анализатору, построенному методом рекурсивного спуска.
* Определение языка, порождаемого грамматикой с действиями.
* Дополнение заданной КС-грамматики действиями, позволяющими учесть дополнительные ограничения на цепочки определяемого ею языка.
* Вставка в заданную (или построенную) грамматику языка *L*1 действий по переводу цепочек этого языка в цепочки языка *L*1 по заданному закону соответствия между цепочками (т. е. реализовать формальный перевод).
* Определение языков *L*1 и *L*1 по заданной грамматике с действиями, реализующей формальный перевод языка *L*1 в язык *L*1.
* Запись в ПОЛИЗе заданного фрагмента программы на языке Х.
* Восстановление текста на языке Х заданного фрагмента программы в ПОЛИЗе.
* Определение, является ли данная запись ПОЛИЗом заданной конструкции.
* Вставка в грамматику, порождающую некоторую конструкцию языка программирования, действий, осуществляющих синтаксически управляемый перевод этой конструкции в ПОЛИЗ при анализе РС-методом.