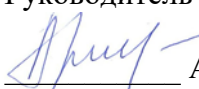




МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

«СОГЛАСОВАНО»
Руководитель ОП
 Артемьева И.Л.

«УТВЕРЖДАЮ»
И.о. директора департамента

Смагин С.В.
«23» марта 2022 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Теория языков программирования и компиляторы
Направление подготовки 09.03.04 Программная инженерия
(Программная инженерия)
Форма подготовки очная

курс 3, 4 семестр 5, 6, 7
лекции 34 час.
практические занятия 34 час.
лабораторные работы 18 час.
в том числе с использованием МАО лек. - / пр. - / лаб. - час.
всего часов аудиторной нагрузки 86 час.
в том числе с использованием МАО 0 час.
самостоятельная работа 166 час.
в том числе на подготовку к экзамену 72 час.
контрольные работы (количество) не предусмотрены
курсовая работа / курсовой проект 7 семестр
зачет 6 семестр
экзамен 5, 7 семестры

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 Программная инженерия, утвержденный приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 920 (с изменениями и дополнениями).

Рабочая программа обсуждена на заседании департамента программной инженерии и искусственного интеллекта, протокол № 3.0 от «23» марта 2022 г.

И.о. директора департамента программной инженерии и искусственного интеллекта к.т.н. Смагин С.В.

Составитель (ли): д.т.н., профессор Артемьева И.Л.; доцент Лемза А.О.

Владивосток

2022

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании кафедры/департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании кафедры/департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании кафедры/департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании кафедры/департамента:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

Цели и задачи освоения дисциплины:

Цель: ознакомление с существующей теорией формальных языков и трансляций; с типами автоматов и преобразователей; с понятием трансляции, схемой и методами построения компиляторов.

Задачи:

- изучение методов задания формального языка с использованием регулярных выражений, порождающих грамматик и распознавателей;
- изучение методов синтаксического анализа заданного формального языка;
- изучение методов продолжения разбора в случае возникновения ошибочной ситуации при разборе цепочки;
- изучение типов контекстных условий языков программирования;
- изучение типов языковых процессоров и методов их разработки.

Для успешного изучения дисциплины «Теория языков программирования и компиляторы» у обучающихся должны быть знания по алгоритмизации, методам составления и тестирования программ.

В результате изучения данной дисциплины у обучающихся формируются следующие профессиональные компетенции:

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
организационно-управленческий	ПК-3 Способность оформления методических материалов и пособий по применению программных систем	ПК-3.1 Демонстрирует знание системы оформления методических материалов по применению программных систем
		ПК-3.2 Использует систему оформления методических материалов по применению программных систем
		ПК-3.3 Оформляет методические материалы и пособия по применению программных систем
научно-исследовательский	ПК-4 Способность к использованию методов и инструментальных средств исследования объектов профессиональной деятельности	ПК-4.1 Демонстрирует знание современных инструментальных средств программного обеспечения
		ПК-4.2 Анализирует и выбирает инструментальные средства программного обеспечения
		ПК-4.3 Использует методы и инструментальные средства исследования программного обеспечения

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
	ПК-5 Способность готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях	ПК-5.1 Демонстрирует знание современных программных продуктов по подготовке презентаций и оформлению научно-технических отчетов
		ПК-5.2 Использует современные программные продукты по подготовке презентаций и оформлению научно-технических отчетов
		ПК-5.3 Выполняет подготовку статей и докладов на научно-технических конференциях
проектный	ПК-6 Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения	ПК-6.1 Демонстрирует знание основ моделирования и формальных методов конструирования программного обеспечения
		ПК-6.2 Использует формальные методы конструирования программного обеспечения
		ПК-6.3 Применяет методы формализации и моделирования программного обеспечения

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-3.1 Демонстрирует знание системы оформления методических материалов по применению программных систем	Знает современные средства по подготовке методических материалов Умеет использовать современные средства по подготовке методических материалов Владеет набором операций, предоставляемых современными средствами по подготовке методических материалов
ПК-3.2 Использует систему оформления методических материалов по применению программных систем	Знает структуры методических материалов по применению программных систем Умеет готовить методические материалы в соответствии со структурой Владеет методами пояснения основных возможностей программных средств с помощью методических материалов
ПК-3.3 Оформляет методические материалы и пособия по применению программных систем	Знает методы оформления методических материалов Умеет оформить методические материалы по применению программных систем Владеет правилами оформления методических материалов

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-4.1 Демонстрирует знание современных инструментальных средств программного обеспечения	<p>Знает современные средства автоматизации разработки программных систем разных классов</p> <p>Умеет использовать средства автоматизации разработки программных систем разных классов.</p> <p>Владеет методами разработки и оценки качества программных систем разных классов</p>
ПК-4.2 Анализирует и выбирает инструментальные средства программного обеспечения	<p>Знает методы сравнения инструментальных средств для разработки программных систем разных классов</p> <p>Умеет сравнивать инструментальные средства в соответствии с требованиями создаваемого программного обеспечения</p> <p>Владеет операциями, предоставляемыми современными инструментальными системами разработки программных систем разных классов</p>
ПК-4.3 Использует методы и инструментальные средства исследования программного обеспечения	<p>Знает методы исследования программного обеспечения</p> <p>Умеет проводить исследование программных средств для разработки программных систем разных классов</p> <p>Владеет методиками оценки соответствия выбранного программного средства разрабатываемому проекту</p>
ПК-5.1 Демонстрирует знание современных программных продуктов по подготовке презентаций и оформлению научно-технических отчетов	<p>Знает современные программные продукты по подготовке презентаций и оформлению научно-технических отчетов</p> <p>Умеет использовать современные программные продукты при подготовке презентаций и оформлении научно-технических отчетов</p> <p>Владеет средствами, предоставляемыми современными программными продуктами для подготовки презентаций и оформлению научно-технических отчетов</p>
ПК-5.2 Использует современные программные продукты по подготовке презентаций и оформлению научно-технических отчетов	<p>Знает набор операций современных программных продуктов по подготовке презентаций и оформлению научно-технических отчетов</p> <p>Умеет пользоваться набором операций современных программных продуктов по подготовке презентаций и оформлению научно-технических отчетов</p> <p>Владеет методами подготовки презентаций и научно-технических отчетов с помощью современных программных продуктов</p>
ПК-5.3 Выполняет подготовку статей и докладов на научно-технических конференциях	<p>Знает методы подготовки статей и докладов</p> <p>Умеет подготовить статью и доклад с помощью современных информационных технологий</p> <p>Владеет методами определения структуры статьи, подбора литературы для обзора и формулировки основного содержания</p>

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-6.1 Демонстрирует знание основ моделирования и формальных методов конструирования программного обеспечения	<p>Знает основные модели информационных технологий, используемых при создании программ на языке высокого уровня, процесс создания программ, стандарты языков программирования,</p> <p>Умеет использовать основные модели информационных технологий при создании программ на языке высокого уровня, общие принципы построения алгоритмов, основные алгоритмические конструкции;</p> <p>Владеет основными моделями информационных технологий, используемых при создании программ на языке высокого уровня</p>
ПК-6.2 Использует формальные методы конструирования программного обеспечения	<p>Знает методы использования информационных технологий при подготовке документации и текстовых редакторов при разработке программ и документации;</p> <p>Умеет использовать информационные технологии при подготовке документации и разработке программ</p> <p>Владеет средствами, предоставляемыми информационными технологиями для подготовки документации и разработке программ</p>
ПК-6.3 Применяет методы формализации и моделирования программного обеспечения	<p>Знает методы формализации и моделирования программного обеспечения;</p> <p>Умеет разрабатывать программы, использующие данные в произвольном формате</p> <p>Владеет навыками разработки формальных моделей программного обеспечения</p>

Трудоёмкость дисциплины и видов учебных занятий по дисциплине

Общая трудоёмкость дисциплины составляет 7 зачётных единиц (252 академических часа).

(1 зачетная единица соответствует 36 академическим часам)

Видами учебных занятий и работы обучающегося по дисциплине являются:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекции
Лаб	Лабораторные работы
Пр	Практические занятия
СР	Самостоятельная работа обучающегося в период теоретического обучения
КР	Курсовая работа
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с

Структура дисциплины:

Форма обучения – очная.

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося					Формы промежуточной аттестации
			Лек	Лаб	Пр	СР	Контроль	
1	Тема 1. Алфавит. Цепочки Тема 2. Способы задания языков Тема 3. Контекстно-свободные языки и грамматики Тема 4. Преобразования КС-грамматик Тема 5. Леворекурсивные и праворекурсивные грамматики Тема 6. Неоднозначные грамматики Тема 7. Распознаватели, их структура и типы Тема 8. Автоматы с магазинной памятью	5	16	0	0	49	27	экзамен
	0		0	16				
3	Тема 9. Понятие трансляции. Обзор процесса трансляции Тема 10. Разработка лексического анализатора языка Тема 11. Разработка синтаксического анализатора языка Тема 12. Проверка контекстных условий Тема 13. Генерация кода Тема 14. Оптимизация кода Тема 15. Системы автоматического построения трансляторов	6	18	0	0	36	0	зачёт
	0		0	18				
4	Практическое занятие №4. Реализация лексического анализатора Практическое занятие №5. Проектирование синтаксического анализатора Практическое занятие №6. Реализация синтаксического анализатора Практическое занятие №7. Реализация семантического анализатора		0	0	18			
5	Лабораторная работа № 1. Проектирование генератора Лабораторная работа №2. Реализация генератора Лабораторная работа №3. Сборка, отладка и	7	0	18	0	27	27	экзамен, курсовая работа

	тестирование «классического» компилятора Лабораторная работа №4. Подготовка документации по разработке компилятора и оформление отчета							
	Итого:		34	18	34	94	72	

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Лекционные занятия (34 часа)

Тема 1. Алфавит. Цепочки (2 часа)

Длина цепочки. Пустая цепочка. Операции над цепочками. Подцепочка, суффикс и префикс цепочки. Позиция в цепочке. Вхождение подцепочки в цепочку. Языки. Конкатенация языков, операция итерации языка. Гомоморфизм, применение гомоморфизма к языку. Обращение гомоморфизма. Операция подстановки языков вместо символов.

Тема 2. Способы задания языков (2 часа)

Регулярные множества и регулярные выражения. Основные алгебраические свойства регулярных выражений. Формальные грамматики. Выводимость цепочки в грамматике. Вывод в грамматике. Полный и неполный выводы. Терминальное порождение символа. Язык, порождаемый грамматикой. Перестройка вывода. Размеченный вывод. Равносильность выводов. Эквивалентность грамматик. Классификация грамматик по Хомскому: праволинейные, леволинейные, правоавтоматные, левоавтоматные, контекстно-свободные грамматики, грамматики непосредственных составляющих, грамматики общего вида. Связь регулярных множеств и праволинейных грамматик. Расщепление грамматик.

Тема 3. Контекстно-свободные языки и грамматики (2 часа)

Граф контекстно-свободной (КС) грамматики. Дерево вывода в КС-грамматике. Крона дерева вывода, сечение дерева вывода, крона сечения. Левый (правый) выводы, левовыводимая (правовыводимая) цепочка. Неоднозначные КС грамматики.

Тема 4. Преобразования КС-грамматик (2 часа)

Бесполезные и недостижимые символы грамматики. Цепные правила. Циклы в грамматике. Удаление бесполезных и недостижимых символов. Удаление цепных правил. Грамматики без λ -правил.

Тема 5. Леворекурсивные и праворекурсивные грамматики (2 часа)

Непосредственная левая рекурсия. Методы устранения левой рекурсии. Нормальная форма Хомского. Нормальная форма Грейбах.

Тема 6. Неоднозначные грамматики (2 часа)

Способы устранения неоднозначности. Контекстные условия языков.

Тема 7. Распознаватели, их структура и типы (2 часа)

Преобразователи, их типы. Конечные автоматы. Конфигурации конечного автомата. Недетерминированные и детерминированные конечные автоматы. Диаграмма конечного автомата. Полностью и неполностью определенный конечный автомат. Достижимые состояния конечного автомата. Язык, допускаемый конечным автоматом. Эквивалентность автоматов. Связь между конечными автоматами и праволинейными грамматиками. Канонический конечный автомат.

Тема 8. Автоматы с магазинной памятью (2 часа)

Конфигурации автомата с магазинной памятью. Язык, допускаемый автоматом с магазинной памятью. Недетерминированный и детерминированный автомат с магазинной памятью. Расширенный автомат с магазинной памятью. Связь между автоматами с магазинной памятью и КС-грамматиками.

Тема 9. Понятие трансляции. Обзор процесса трансляции (2 часа)

Формальный язык и языковой процессор на его основе. Теоретический язык и входной язык языкового процессора. Типы языковых процессоров (компиляторы, интерпретаторы, смешанные формы). Структура языкового процессора. Задачи, решаемые лексическим, синтаксическим и семантическим анализаторами. Задачи, решаемые генератором кода. Задачи блока оптимизации.

Тема 10. Разработка лексического анализатора языка (2 часа)

Реализация лексического анализа в языковом процессоре. Лексические ошибки. Работа с таблицей имен. Способы хранения длинных имен и строк. Токены, шаблоны, лексемы. Атрибуты токенов. Их вычисление при разборе и хранение. Язык спецификации лексических анализаторов.

Тема 11. Разработка синтаксического анализатора языка (6 часов)

Понятие о стратегии синтаксического анализа. Важнейшие классы грамматик и соответствующие им анализаторы. Формализмы для задания грамматик. Нисходящий и восходящий разбор. Обработка синтаксических ошибок. Методы продолжения разбора при ошибках. Способы представления результатов анализа: дерево разбора, синтаксическое дерево, польская запись. Работа с таблицей имен при синтаксическом анализе.

Тема 12. Проверка контекстных условий (2 часа)

Контекстные условия языков. Атрибутные грамматики. Задание контекстных условий как предикатов над атрибутами. Организация вычисления значений атрибутов при семантическом анализе. Организация проверки контекстных условий. Особенности семантического анализа для языков с

полиморфными операциями. Преобразования типов. Область видимости имен в анализируемых программах. Работа с таблицей имен.

Тема 13. Генерация кода (2 часа)

Среды периода выполнения. Стеки управления, деревья активации. Организация памяти для объектной программы. Общая схема генерации и внутреннее представление команд. Генерация выражений и операторов присваивания. Генерация управления вычислениями. Генерация подпрограмм и обращений к ним. Распределение памяти.

Тема 14. Оптимизация кода (2 часа)

Схемы программ. Поточковый анализ программ. Оптимизирующие преобразования. Организация трансляторов. Схемы процесса трансляции и организация трансляторов: однопросмотровая схема, двухпросмотровая схема, многопросмотровые схемы. Влияние типов выходных языков на организацию трансляторов. Распараллеливающие компиляторы.

Тема 15. Системы автоматического построения трансляторов (2 часа)

Атрибутные грамматики. Алгоритмы заполнения атрибутных деревьев, атрибутные СПТ. Примеры.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА И САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Практические занятия (16 часов, 5 семестр)

Практическое занятие № 1. Разработка внешних спецификаций «классического» компилятора (6 часов)

Практическое занятие № 2. Эскизное проектирование компилятора (4 часа)

Практическое занятие № 3. Проектирование лексического анализатора (6 часов)

Практические занятия (18 часов, 6 семестр)

Практическое занятие №4. Реализация лексического анализатора (4 часа)

Практическое занятие №5. Проектирование синтаксического анализатора (4 часа)

Практическое занятие №6. Реализация синтаксического анализатора (6 часов) **Практическое занятие №7.** Реализация семантического анализатора (4 часа)

Лабораторные работы (18 часов, 7 семестр)

Лабораторная работа № 1. Проектирование генератора (4 часа)

Лабораторная работа №2. Реализация генератора (6 часов)

Лабораторная работа №3. Сборка, отладка и тестирование «классического» компилятора (4 часа)

Лабораторная работа №4. Подготовка документации по разработке компилятора и оформление отчета (4 часа)

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Теория языков программирования и компиляторы» включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	1-4 недели семестра	Подготовка к теоретическому опросу по темам 1-2	2 часа	УО-2 (коллоквиум)
	1-4 недели семестра	Подготовка к контрольной работе по темам 1-2	6 часов	ПР-2 (контрольная работа)
	5-10 недели семестра	Подготовка к теоретическому опросу по темам 3-5	2 часа	УО-2 (коллоквиум)
	5-8 недели семестра	Подготовка к контрольной работе по темам 3-4	6 часов	ПР-2 (контрольная работа)
	11-16 недели семестра	Подготовка к теоретическому опросу по темам 6-8	3 часа	УО-2 (коллоквиум)

	9-12 недели семестра	Подготовка к контрольной работе по темам 5-6	6 часов	ПР-2 (контрольная работа)
	13-16 недели семестра	Подготовка к контрольной работе по темам 7-8	6 часов	ПР-2 (контрольная работа)
	17-18 неделя семестра	Подготовка к экзамену	45 часов	Экзамен
2	1-4 недели семестра	Подготовка к теоретическому опросу по темам 9-10	4 часа	УО-2 (коллоквиум)
	1-4 недели семестра	Выполнение домашних заданий по темам 9-10	8 часов	ПР-2 (контрольная работа)
	5-12 недели семестра	Подготовка к теоретическому опросу по темам 11-12	4 часа	УО-2 (коллоквиум)
	5-12 недели семестра	Выполнение домашних заданий по темам 11-12	8 часов	ПР-2 (контрольная работа)
	13-18 недели семестра	Подготовка к теоретическому опросу по темам 13-15	4 часа	УО-2 (коллоквиум)
	13-18 недели семестра	Выполнение домашних заданий по темам 13-15	8 часов	ПР-2 (контрольная работа)
3	1-4 недели семестра	Подготовка к лабораторной работе 1	4 часа	ПР-6 (лабораторная работа)
	5-10 недели семестра	Подготовка к лабораторной работе 2	4 часа	ПР-6 (лабораторная работа)
	11-14 недели семестра	Подготовка к лабораторной работе 3	4 часа	ПР-6 (лабораторная работа)
	15-18 недели семестра	Подготовка к лабораторной работе 4	4 часа	ПР-6 (лабораторная работа)
	1-16 недели семестра	Подготовка текста курсовой работы	11 часов	ПР-9 (проект / курсовой проект)
	17-18 недели семестра	Подготовка к экзамену	27 часов	Экзамен
Итого:			130 часов	

Рекомендации по самостоятельной работе студентов

Самостоятельная работа студентов состоит в работе с литературой, подготовке к проверочным работам и выполнении домашних заданий по темам.

Работа с литературой

В процессе подготовки к тесту студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной учебно-методической литературы. Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами Интернета является наиболее эффективным методом получения знаний, позволяет значительно активизировать процесс овладения информацией, способствует более

глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме.

Подготовка к лабораторным работам и практическим занятиям.

Подготовку к каждой лабораторной работе и практическому занятию каждый студент должен начать с изучения теоретического материала и ознакомления с планом, который отражает содержание предложенной темы. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса. Результат такой работы должен проявиться в способности студента свободно ответить на теоретические вопросы по теме задания, правильном выполнении лабораторной работы.

Работа на лабораторных и практических занятиях ведется в малых группах 3-4 человека в каждой. В ходе выполнения лабораторной работы студенты учатся работе в группе, распределению ресурсов и установлению взаимодействия в коллективной работе. В процессе выполнения лабораторной работы студенты должны создать требуемый документ с помощью предлагаемого программного средства или выполнить кодирование в соответствии с заданием.

Контрольные работы.

По теме лекций студентам выдаются индивидуальные задания в рамках контрольных работ, предназначенные для закрепления и проверки знаний, полученных на лекциях. Задания выполняются самостоятельно. Выполненные задания сдаются преподавателю на проверку и возвращаются им на доработку при совершении ошибок. Задание должно быть выполнено правильно (возможно, за несколько итераций). Структура индивидуального задания определяется темами лекций.

Пример индивидуального задания в рамках контрольной работы по темам 3-4:

1. Для регулярного выражения

$$(0 + 2)(a + b)^*(1 + 2)^*1$$

построить праволинейную грамматику, порождающую тот же язык.

2. С помощью алгоритма 8.1 (Не пуст ли язык?) решить проблему пустоты языка, определяемого грамматикой $G = (\{A, C, S\}, \{a, b, c\}, P, S)$, где $P = \{S \rightarrow SAaA, A \rightarrow Sab \mid aCc, C \rightarrow c \mid \lambda\}$.

3. Для контекстно-свободной грамматики $G = (\{A, C, S\}, \{a, b, c\}, P, S)$, где $P = \{S \rightarrow AaCa, A \rightarrow baa \mid Cc \mid \lambda, C \rightarrow c\}$:

а) построить пример дерева полного вывода;

б) выписать крону построенного дерева;

в) записать левый и правый выводы по построенному дереву.

4. Устранить λ -правила в грамматике задания 3.

5. Устранить цепные правила в грамматике, получившейся в результате выполнения задания 4. Если в такой грамматике цепных правил нет, устранить цепные правила в грамматике $G = (\{A, B, C, S\}, \{a, b, c\}, P, S)$, где $P = \{C \rightarrow aCA \mid aC \mid aA \mid a, A \rightarrow CaB \mid aB, B \rightarrow abc, S \rightarrow C \mid \lambda\}$.

Критерии оценки индивидуальных заданий (проектов)

– 100-86 баллов выставляется, если содержание и составляющие части соответствуют выданному заданию. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет;

– 85-76 баллов выставляется, если при выполнении задания допущено не более одной ошибки. Продемонстрировано владение навыками подготовки документа по теме. Фактических ошибок, связанных с пониманием структуры и содержания задания нет;

– 75-61 балл выставляется, если при выполнении задания допущено не более двух ошибок. Продемонстрировано знание и владение навыками подготовки документа по теме. Допущено не более 2 ошибок, связанных с пониманием структуры и содержания задания;

– 60-0 баллов выставляется, если структура и содержание задания не соответствуют требуемым.

Шкала оценивания

Менее 60 баллов	не зачтено	неудовлетворительно
От 61 до 75 баллов	зачтено	удовлетворительно
От 76 до 85 баллов	зачтено	хорошо
От 86 до 100 баллов	зачтено	отлично

Командные проекты выполняются в рамках курсовой работы в 7 семестре.

Тематика проектов связана с разработкой учебной версии языкового процессора, выполняющего разбор программы на исходном языке и формирование программы на объектном языке.

Содержание курсовой работы

1. Разработка грамматики исходного языка – подмножества одного из существующих алгоритмических.
2. Разработка грамматики объектного языка.
3. Разработка контекстных условий исходного языка.
4. Проектирование лексического анализатора исходного языка, определение ошибочных ситуаций.
5. Выбор метода синтаксического анализа и проектирование синтаксического анализатора исходного языка.
6. Проектирование семантического анализатора исходного языка.
7. Определение соответствия конструкций исходного и объектного языков, проектирование генератора программы на объектном языке.
8. Проектирование всех требуемых структур данных.
9. Реализация программной системы, состоящей из лексического, синтаксического и семантического анализаторов, и генератора объектной программы.
10. Разработка тестов для проверки работоспособности программной системы.
11. Написание отчета по курсовой работе.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы/темы дисциплины	Код и наименование индикатора достижения	Результаты обучения	Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Тема 1. Алфавит. Цепочки.	ПК-5.1	Знает понятие языка и его свойства, теоретические способы задания языков; знает методы поиска информации по указанным вопросам.	УО-2 (коллоквиум), ПР-2 (контрольная работа)	Экзамен, вопросы 1-2
2	Тема 2. Способы задания языков	ПК-5.1	Знает понятие языка и его свойства, теоретические способы задания языков; знает методы поиска информации по указанным	УО-2, ПР-2	Экзамен, вопросы 3-6

			вопросам.		
3	Тема 3, 4. Контекстно-свободные языки и грамматики. Преобразования КС – грамматик	ПК-5.2	Знает теоретические основы информационных процессов преобразования грамматик; знает методы поиска информации по указанным вопросам.	УО-2, ПР-2	Экзамен, вопросы 7-17
4	Тема 5. Леворекурсивные и праворекурсивные грамматики.	ПК-5.2	Знает теоретические основы информационных процессов преобразования грамматик; знает методы поиска информации по указанным вопросам.	УО-2, ПР-2	Экзамен, вопросы 18-21
5	Тема 6, 7, 8. Неоднозначные грамматики. Распознаватели, их структура и типы. Автоматы с магазинной памятью	ПК-5.3	Знает теоретические основы информационных процессов преобразования распознавателей; знает методы поиска информации по указанным вопросам.	УО-2, ПР-2	Экзамен, вопросы 22-29
6	Тема 9. Понятие трансляции. Обзор процесса трансляции.	ПК-4.1	Знает методы и средства создания трансляторов.	УО-2, ПР-2	Зачёт, вопрос 1
7	Тема 10. Разработка лексического анализатора языка	ПК-4.2	Знает о возможностях существующих программных средств и сред проектирования программного обеспечения.	УО-2, ПР-2	Зачёт, вопросы 2-8

8	Тема 11. Разработка синтаксического анализатора языка	ПК-4.3	Знает о возможностях существующих программных средств и сред разработки программного обеспечения.	УО-2, ПР-2	Зачёт, вопросы 9-14
9	Тема 12, 13. Проверка контекстных условий. Генерация кода	ПК-6.1	Знает об основных классах схем программ и типов языков.	УО-2, ПР-2	Зачёт, вопросы 15-21
10	Тема 14, 15. Оптимизация кода. Системы автоматического построения трансляторов	ПК-6.2 ПК-6.3	Знает о соответствиях основных классов схем программ и типов языков. Знает о методах формализации и моделирования программного обеспечения.	УО-2, ПР-2	Зачёт, вопросы 22-26
11	Практическое занятие № 1. Разработка внешних спецификаций «классического» компилятора	ПК-4.1 ПК-5.1	Умеет выбирать, проектировать, реализовывать, оценивать качество и проводить анализ эффективности программного обеспечения для решения задач трансляции. Владеет способами подготовки документов по сравнению разных языков и языковых процессоров.	УО-2, ПР-2	ПР-9 (проект/курсовый проект)
12	Практическое занятие № 2. Эскизное проектирование компилятора	ПК-5.1	Умеет искать требуемую литературу по разработке компиляторов. Владеет способами подготовки	УО-2, ПР-2	ПР-9

			документов по сравнению разных языков и языковых процессоров.		
13	Практическое занятие № 3. Проектирование лексического анализатора	ПК-5.2 ПК-5.3 ПК-6.1	Умеет искать литературу по разработке грамматики требуемого языка; искать литературу по разработке распознавателей структурных элементов требуемого языка. Владеет методами создания проекта лексического анализатора.	УО-2, ПР-2	ПР-9
14	Практическое занятие № 4. Реализация лексического анализатора	ПК-5.2 ПК-5.3 ПК-6.1 ПК-6.3	Умеет искать литературу по разработке грамматики требуемого языка; искать литературу по разработке распознавателей структурных элементов требуемого языка; создавать алгоритмы при разработке лексического анализатора. Владеет методами моделирования программного обеспечения.	УО-2, ПР-2	ПР-9
15	Практическое занятие № 5. Проектирование синтаксического анализатора	ПК-4.1 ПК-5.2 ПК-5.3 ПК-6.2	Умеет искать литературу по разработке грамматики требуемого языка; искать	УО-2, ПР-2	ПР-9

			<p>литературу по разработке распознавателей структурных элементов требуемого языка. Владеет методами сравнения алгоритмов синтаксического анализа, методами выбора схемы анализа при выполнении проекта, методами обоснования; методами создания проекта синтаксического анализатора.</p>		
16	<p>Практическое занятие № 6. Реализация синтаксического анализатора</p>	<p>ПК-4.1 ПК-5.2 ПК-5.3 ПК-6.2</p>	<p>Умеет искать литературу по разработке грамматики требуемого языка; искать литературу по разработке распознавателей структурных элементов требуемого языка; создавать алгоритмы при разработке синтаксического анализатора. Владеет методами сравнения алгоритмов синтаксического анализа, методами выбора схемы анализа при выполнении</p>	<p>УО-2, ПР-2</p>	<p>ПР-9</p>

			проекта, методами обоснования.		
17	Практическое занятие № 7. Реализация семантического анализатора.	ПК-5.2 ПК-6.3	Умеет использовать методы формализации программного обеспечения. Владеет современными программными продуктами по оформлению научно-технических отчётов.	УО-2, ПР-2	ПР-9
18	Лабораторная работа № 1. Проектирование генератора.	ПК-5.3 ПК-6.3	Умеет искать литературу по разработке распознавателей структурных элементов требуемого языка. Владеет методами моделирования программного обеспечения.	ПР-6 (лабораторная работа)	ПР-9
19	Лабораторная работа № 2. Реализация генератора	ПК-6.3	Умеет использовать методы формализации программного обеспечения. Владеет методами моделирования программного обеспечения.	ПР-6	ПР-9
20	Лабораторная работа № 3. Сборка, отладка и тестирование «классического» компилятора	ПК-4.1 ПК-4.2	Умеет выбирать, проектировать, реализовывать, оценивать качество и проводить анализ эффективности программного	ПР-6	ПР-9

			<p>обеспечения для решения задач трансляции; выбирать средства, подходящие для выполнения поставленных задач. Владеет методами анализа и оценивания.</p>		
21	<p>Лабораторная работа № 4. Подготовка документации по разработке компилятора и оформление отчета</p>	<p>ПК-3.1 ПК-3.2 ПК-3.3 ПК-5.3</p>	<p>Знает методы поиска информации для подготовки методических материалов по существующим методам генерации и оптимизации кода; методы поиска информации для подготовки методических материалов по существующим методам разработки компиляторов; методы поиска информации для подготовки методических материалов по существующим методам разработки систем автоматического построения трансляторов. Умеет оформлять методические материалы по применению компиляторов; оформлять</p>	<p>ПР-6</p>	<p>ПР-9</p>

			<p>результаты поиска и разработок в виде методических пособий; оформлять результаты поиска и разработок в виде отчётов и докладов. Владеет навыками оформления методических материалов по применению компиляторов; современными средствами подготовки документов и презентаций; навыками оформления методических пособий; навыками подготовки статей и докладов для участия в научно- технических конференциях.</p>		
--	--	--	---	--	--

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также качественные критерии оценивания, которые описывают уровень сформированности компетенций, представлены в разделе VIII.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Серебряков, В. А. Теория и реализация языков программирования: учебное пособие / В.А. Серебряков [и др.]. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2021. – 372 с.

<http://www.iprbookshop.ru/102068.html>

2. Котляров, В. П. Основы тестирования программного обеспечения: учебное пособие / В. П. Котляров, Т. В. Коликова. – М. : Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2016. – 285 с.

<http://lib.dvfu.ru:8080/lib/item?id=chamo:797490&theme=FEFU>

3. Артемьева И. Л. Теория вычислительных процессов и структур. Часть 1. Языки и способы их задания. Учебно-методическое пособие. – Владивосток : Издательство Дальневосточного федерального университета, 2011. – 60 с.

4. Артемьева И.Л. Теория вычислительных процессов и структур. Часть 2. Методы синтаксического анализа программ. Учебно-методическое пособие. – Владивосток: Издательство Дальневосточного федерального университета, 2011. – 56 с.

5. Гагарина, Л. Г. Введение в теорию алгоритмических языков и компиляторов: учеб. пособие / Л. Г. Кокорева, Е. В. Кокорева. – М.: ИД ФОРУМ, 2011. – 176 с.

<http://znanium.com/catalog/product/265617>

6. Вирт, Н. Построение компиляторов / Н. Вирт // Пер. с англ. Е. В. Борисов, Л. Н. Чернышов. – М. : ДМК Пресс, 2013. – 192 с.

7. Ахо, А. Теория синтаксического анализа, перевода и компиляции. Синтаксический анализ / А. Ахо, Дж. Ульман. – М. : Книга по требованию, 2012. – 613 с.

8. Федосеева Л.И., Адилов Р.М., Шмоткин М.Н. Основы теории автоматов и формальных языков: учебное пособие . – Пенза : Издательство ПензГТУ, 2013. – 136 с. <https://e.lanbook.com/book/62703>

Дополнительная литература

(печатные и электронные издания)

1. Ахо, А. Компиляторы. Принципы, технологии и инструментарий / А. Ахо, Р. Сети, Дж. Ульман. – М. : Издательский дом «Вильямс», 2001. – 767 с.

<https://lib.dvfu.ru:8443/lib/item?id=chamo:334968&theme=FEFU>

2. Шульга, Т. Э. Теория автоматов и формальных языков : учебное пособие / Т. Э. Шульга. – Саратов : Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2015. – 104 с.

<http://www.iprbookshop.ru/76519.html>

3. Малявко, А. А. Формальные языки и компиляторы: учебник / А. А. Малявко. – Новосибирск : Новосибирский государственный технический университет, 2014. – 431 с. <http://www.iprbookshop.ru/47725.html>

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. <https://e.lanbook.com/book/66125> Залогова, Л. А. Разработка Паскаль-компилятора / Л. А. Залогова. – М. : Лаборатория знаний, 2014. – 186 с.

2. <https://e.lanbook.com/book/39992> Вирт, Н. Разработка операционной системы и компилятора. Проект Оберон / Н. Вирт, Ю. Гуткнехт. – М. : ДМК Пресс, 2012. – 560 с.

3. <http://www.coders-library.ru/> Библиотека программиста: все, что нужно для комфортной работы.

Перечень информационных технологий и программного обеспечения

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используется следующее программное обеспечение:

1. Microsoft Office (PowerPoint, Word и Visio).
2. Open Office.
3. Microsoft Visual Studio.

При осуществлении образовательного процесса студентами и профессорско-преподавательским составом используются следующие информационно-справочные системы:

1. Научная электронная библиотека eLIBRARY.
2. Электронно-библиотечная система издательства «Лань».
3. Электронная библиотека «Консультант студента».
4. Электронно-библиотечная система IPRbooks.
5. Электронно-библиотечная система Znanium.
6. Доступ к электронному заказу книг в библиотеке ДВФУ, доступ к нормативным документам ДВФУ, расписанию, рассылке писем.

Лекции проводятся с использованием проектора и мультимедийного

комплекса для проведения лекций внутренней системы портала ДВФУ. Практические занятия и лабораторные работы проводятся в специализированном компьютерном классе.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина изучается в следующих организационных формах: лекции, лабораторное занятие; курсовая работа, самостоятельное изучение теоретического материала; самостоятельное выполнение индивидуального задания; индивидуальные консультации.

Работа на лекции

Слушание и запись лекций – сложный вид вузовской аудиторной работы. Внимательное слушание и конспектирование лекций предполагает интенсивную умственную деятельность студента. Краткие записи лекций, их конспектирование помогает усвоить учебный материал. Конспект является полезным тогда, когда записано самое существенное, основное и сделано это самим студентом. Запись лекций рекомендуется вести по возможности собственными формулировками. Желательно запись осуществлять на одной странице, а следующую оставлять для проработки учебного материала самостоятельно в домашних условиях. Конспект лекции лучше подразделять на пункты, параграфы, соблюдая красную строку. Этому в большой степени будут способствовать пункты плана лекции, предложенные преподавателям. Принципиальные места, определения, формулы и другое следует сопровождать замечаниями «важно», «особо важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек. Лучше если они будут собственными, чтобы не приходилось просить их у однокурсников и тем самым не отвлекать их во время лекции. Целесообразно разработать собственную «маркографию» (значки, символы), сокращения слов. Не лишним будет и изучение основ стенографии. Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть знаниями.

К прослушиванию лекции следует готовиться, для этого необходимо знать программу курса и рекомендованную литературу. Тогда в процессе лекции легче отделить главное от второстепенного, легче сориентироваться: что записать, что самостоятельно проработать, что является трудным для понимания, а что легко усвоить.

Лабораторные работы и практические занятия

В результате выполнения лабораторных работ и практических занятий студент должен изучить основные возможности современных программных средств информационных технологий, позволяющих разрабатывать и создавать программные средства. Каждый этап работы должен быть задокументирован и отражен в соответствующей главе документации.

Результатом лабораторной работы и практического занятия является созданный документ, который демонстрируется преподавателю в конце работы, либо работающее программное средство. Студент должен уметь отвечать на вопросы преподавателя, поясняя процесс создания кода для возможности тестирования по «белому ящику».

Самостоятельная работа студента

Основной формой самостоятельной работы студента является изучение теоретического материала, его дополнение рекомендованной литературой, выполнение индивидуальных заданий, а также активная работа на лабораторных занятиях.

Целью выполнения индивидуальных заданий является

- проверка знаний теоретического лекционного материала
- закрепление практических навыков, полученных в процессе выполнения лабораторной работы.

Индивидуальное задание включает в себя описание той работы, которую необходимо проделать студенту. Примеры индивидуальных заданий приведены в приложении 2.

Контроль за выполнением самостоятельной работы студента производится в виде контроля каждого этапа работы (см. приложение 1). Студент должен планировать график самостоятельной работы по дисциплине и придерживаться его.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень материально-технического и программного обеспечения дисциплины приведен в таблице.

Материально-техническое и программное обеспечение дисциплины

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
690922, Приморский край, г. Владивосток,	Помещение укомплектовано специализированной	1С Предприятие (8.2), 7-Zip, ABBYY Lingvo 12, Alice 3, Anaconda 3, Autodesk, CodeBlocks, CorelDRAW X7, Dia, Directum 4.8,

<p>остров Русский, полуостров Саперный, поселок Аякс, 10, корпус D, ауд. D 733,733а. Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации</p>	<p>учебной мебелью (посадочных мест – 13) Оборудование: ЖК-панель 47", Full HD, LG M4716 CCBA – 1 шт. Доска аудиторная, Моноблок Lenovo C360G-i34164G500UDK с лицензионными программами Microsoft Office 2013 (13 шт.) и аудиовизуальными средствами проектор Panasonic DLPProjectorPT-D2110XE</p>	<p>DosBox-0.74, Farmanager, Firebird 2.5, FlameRobin, Foxit Reader, Free Pascal, Geany, Ghostscript, Git, Greenfoot, gsview, Inscape 0.91, Java, Java development Kit, Kaspersky, Lazarus, LibreOffice 4.4, MatLab R2017b, Maxima 5.37.2, Microsoft Expression, Microsoft Office 2013, Microsoft Silverlight Microsoft System Center, Microsoft Visual Studio 2012, MikTeX 2.9, MySQL, NetBeans, Notepad++, Oracle VM VirtualBox, PascalABC.NET, PostgreSQL 9.4, PTC Mathcad, Putty, PyQt GPL v5.4.1 for Python 3.4, Python2.7(3.4, 3.6), QGIS Brighton, RStudio, SAM CoDeC Pack, SharePoint, Strawberry Perl, Tecnomatix, TeXnicCenter, TortoiseSVN, Unity 2017.3.1f1, Veusz, Vim 8.1, Visual Paradigm CE, Windows Kits, Windows Phone SDK 8.1, Xilinx Design Tools, Acrobat Reader DC, Adobe Bridge CS3, Adobe Device Central CS3, Adobe Extend Script Toolkit 2, Adobe Photoshop CS3, DVD-студия Windows, Google Chrome, Internet Explorer, ITMOproctor, Mozilla Firefox, Windows Media Center, WinSCP</p>
---	--	---

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья в ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащенные туалетными комнатами, табличками информационно-навигационной поддержки.

VIII. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Текущая аттестация студентов. Текущая аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

Текущая аттестация проводится в форме проверки индивидуальных домашних заданий, теста для проверки теоретических знаний, а также в форме защиты курсового проекта, выполняемого в рамках самостоятельной работы параллельно с лабораторными работами и осуществляется ведущим преподавателем.

Объектами оценивания выступают:

- степень усвоения теоретических знаний - оценивается в форме контрольной работы или тестирования;
- уровень овладения практическими умениями и навыками – оценивается в форме защиты индивидуального задания (проекта).

Промежуточная аттестация студентов. Промежуточная аттестация студентов проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По дисциплине предусмотрены экзамены, курсовой проект и зачет, которые проводятся в устной форме.

Критерии выставления оценки студенту на зачете/экзамене

Баллы (рейтингов ой оценки)	Оценка зачета (стандартная)	Требования к сформированным компетенциям
86-100	«зачтено»/ «отлично»	Оценка «зачтено» / «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
76-85	«зачтено»/ «хорошо»	Оценка «зачтено» / «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
61-75	«зачтено»/ «удовлетворительно»	Оценка «зачтено» / «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.
0-60	«не зачтено»/ «неудовлетворительно»	Оценка «не зачтено» / «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

ВОПРОСЫ К ЭКЗАМЕНУ

1. Алфавит. Цепочка. Длина цепочки. Пустая цепочка.

2. Операции над цепочками. Подцепочка, суффикс и префикс цепочки. Позиция в цепочке. Вхождение подцепочки в цепочку.
3. Языки. Конкатенация языков, операция итерации языка.
4. Гомоморфизм, применение гомоморфизма к языку. Обращение гомоморфизма.
5. Операция подстановки языков вместо символов.
6. Регулярные множества и регулярные выражения. Основные алгебраические свойства регулярных выражений.
7. Формальные грамматики. Выводимость цепочки в грамматике.
8. Вывод в грамматике. Полный и неполный выводы.
9. Терминальное порождение символа. Язык, порождаемый грамматикой.
10. Перестройка вывода. Размеченный вывод. Равносильность выводов.
11. Эквивалентность грамматик. Классификация грамматик по Хомскому: праволинейные, левوليнейные, правоавтоматные, левоавтоматные, контекстно-свободные грамматики, грамматики непосредственных составляющих, грамматики общего вида.
12. Связь регулярных множеств и праволинейных грамматик.
13. Контекстно-свободные языки и грамматики. Граф контекстно-свободной (КС) грамматики. Дерево вывода в КС-грамматике. Крона дерева вывода, сечение дерева вывода, крона сечения. Левый (правый) выводы, левовыводимая (правовыводимая) цепочка. Неоднозначные КС грамматики.
14. Удаление бесполезных символов в грамматике
15. Удаление недостижимых символов грамматики.
16. Цепные правила. Циклы в грамматике. Удаление цепных правил.
17. Грамматики без λ -правил.
18. Леворекурсивные и праворекурсивные грамматики. Непосредственная левая рекурсия. Методы устранения левой рекурсии.
19. Нормальная форма Хомского.
20. Нормальная форма Грейбах
21. Расщепление грамматик.
22. Неоднозначные грамматики. Контекстные условия языков.
23. Распознаватели, их структура и типы.
24. Конечные автоматы. Конфигурации конечного автомата. Недетерминированные и детерминированные конечные автоматы. Диаграмма конечного автомата. Полностью и неполностью определенный конечный автомат.
25. Достижимые состояния конечного автомата. Язык, допускаемый конечным автоматом.

26. Эквивалентность автоматов. Канонический конечный автомат.
27. Связь между конечными автоматами и праволинейными грамматиками.
28. Автоматы с магазинной памятью. Конфигурации автомата с магазинной памятью. Недетерминированный и детерминированный автомат с магазинной памятью. Расширенный автомат с магазинной памятью.
29. Язык, допускаемый автоматом с магазинной памятью. Связь между автоматами с магазинной памятью и КС-грамматиками.

ВОПРОСЫ К ЗАЧЁТУ

1. Формальный язык и языковой процессор на его основе. Теоретический язык и входной язык языкового процессора.
2. Типы языковых процессоров (компиляторы, интерпретаторы, смешанные формы).
3. Структура языкового процессора.
4. Реализация лексического анализа в языковом процессоре.
5. Лексические ошибки.
6. Работа с таблицей имен. Способы хранения длинных имен и строк.
7. Токены, шаблоны, лексемы. Атрибуты токенов. Их вычисление при разборе и хранение.
8. Язык спецификации лексических анализаторов.
9. Понятие о стратегии синтаксического анализа. Нисходящий и восходящий разбор.
10. Важнейшие классы грамматик и соответствующие им анализаторы.
11. Формализмы для задания грамматик.
12. Обработка синтаксических ошибок. Методы продолжения разбора при ошибках.
13. Способы представления результатов анализа: дерево разбора, синтаксическое дерево, польская запись.
14. Работа с таблицей имен при синтаксическом анализе.
15. Контекстные условия языков. Их классы. Атрибутные грамматики.
16. Задание контекстных условий как предикатов над атрибутами.
17. Организация вычисления значений атрибутов при семантическом анализе. Организация проверки контекстных условий.
18. Особенности семантического анализа для языков с полиморфными операциями.
19. Преобразования типов.
20. Область видимости имен в анализируемых программах.

21. Среды периода выполнения. Стеки управления, деревья активации.
22. Организация памяти для объектной программы.
23. Общая схема генерации и внутреннее представление команд.
24. Генерация выражений и операторов присваивания.
25. Генерация управления вычислениями.
26. Генерация подпрограмм и обращений к ним.

Образец экзаменационного билета

Структура экзаменационного билета по курсу «Теория языков программирования и компиляторы»:

1. Теоретический вопрос (1-15 из списка вопросов к экзамену).
2. Теоретический вопрос (16-29 из списка вопросов к экзамену).

Экзаменационный билет № ___

1. Языки. Конкатенация языков, операция итерации языка.
2. Грамматики без λ -правил.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕЙ АТТЕСТАЦИИ

Примеры тестовых заданий

1. Алфавит – это непустое множество символов, которое
 - 1) бесконечное
 - 2) конечное
 - 3) счетное
2. Всякая конечная последовательность символов алфавита A называется
 - 1) предложением
 - 2) цепочкой
 - 3) строкой
3. Не содержащая ни одного символа цепочка является
 - 1) укороченной
 - 2) не укороченной
 - 3) пустой
4. Порядок символов в цепочке
 - 1) важен
 - 2) не важен.
5. Длина цепочки X равна
 - 1) количеству символов в цепочке

- 2) количеству символов в цепочке минус один
6. Если x и y цепочки, то xy является цепочкой, которая называется
- 1) конкатенацией
 - 2) произведением
7. Если $z=xy$ – цепочка, то x
- 1) голова цепочки z
 - 2) начало цепочки z
 - 3) фраза цепочки z
8. Если $z=xy$ – цепочка, то y
- 1) фраза цепочки z
 - 2) хвост цепочки z
 - 3) конец цепочки z
9. Пусть $z=xy$ – цепочка, то x - правильная голова, если y
- 1) пустая цепочка
 - 2) не пустая цепочка
10. Пусть $z=xy$ – цепочка, то y - правильный хвост, если x
- 1) пустая цепочка
 - 2) не пустая цепочка.
11. Продукцией или правилом подстановки называется упорядоченная пара, которая обычно записывается так $U::=X$, где U
- 1) символ
 - 2) цепочка символов
 - 3) не пустая цепочка символов
12. Продукцией или правилом подстановки называется упорядоченная пара, которая обычно записывается так $U::=X$, где X
- 1) символ
 - 2) цепочка символов
 - 3) не пустая конечная цепочка символов
13. Грамматикой $G[z]$ называется, непустое конечное множество правил, где z – начальный символ грамматики, который должен встречаться в левой части правил
- 1) один раз
 - 2) более одного раза
 - 3) один или более раз
14. Все символы, которые встречаются в левых и правых частях правил грамматики, образуют V
- 1) множество терминальных символов
 - 2) множество нетерминальных символов
 - 3) алфавит символов

15. В заданной грамматике G символы, которые встречаются в левой части правил называются

- 1) терминальными
- 2) нетерминальными
- 3) начальными

16. В заданной грамматике G символы, которые не входят во множество V_N нетерминальных символов называются

- 1) входными
- 2) терминальными
- 3) конечными

17. Мы говорим в грамматике G цепочка V порождает цепочку w и обозначаем

- 1) $V \Rightarrow w$
- 2) $V \Rightarrow +w$
- 3) $V \Rightarrow -w$

18. Пока в цепочке есть хотя бы один нетерминальный символ из нее можно вывести

- 1) терминальную цепочку
- 2) новую цепочку

19. В грамматике $G[z]$ цепочка x , выводимая из начального символа z , называется

- 1) предложением языка
- 2) сентенциальной формой
- 3) терминальной цепочкой

20. Если две различные грамматики порождают один и тот же язык, то эти грамматики называются

- 1) совпадающими
- 2) эквивалентными
- 3) различающими

Критерии оценки выполнения тестов

Процент правильных ответов	Оценка
От 95% до 100%	отлично
От 76% до 95%	хорошо
От 61% до 75%	удовлетворительно
Менее 61 %	неудовлетворительно

Примеры индивидуальных заданий

Задание 1. Выпишите все префиксы, суффиксы указанной цепочки. Выпишите все подцепочки указанной цепочки, имеющие длину 3. Указать все вхождения символа a в цепочку w .

Вариант 1. $w = abcaabca$

Вариант 2. $w = acdbacdb$

Вариант 3. $w = cabccabc$

Вариант 4. $w = cccaccsa$

Вариант 5. $w = bbcabbca$

Вариант 6. $w = aaacaacs$

Вариант 7. $w = aaccsaacs$

Вариант 8. $w = cbabcbab$

Вариант 9. $w = caaacaas$

Вариант 10. $w = cbaccbac$

Вариант 11. $w = cacbcacb$

Вариант 12. $w = bbaabbaa$

Вариант 13. $w = dbacdbac$

Вариант 14. $w = babcbabc$

Вариант 15. $w = accdaccd$

Вариант 16. $w = cbcacbca$

Вариант 17. $w = acbcacbc$

Вариант 18. $w = cabccabc$

Вариант 19. $w = abccabcc$

Вариант 20. $w = adcdadcd$

Задание 2. Привести примеры языков в алфавите Σ

Вариант 1. $\Sigma = \{a, d, b, c, m\}$

Вариант 2. $\Sigma = \{a, d, b, f, k\}$

Вариант 3. $\Sigma = \{a, d, c, s, f\}$

Вариант 4. $\Sigma = \{a, b, c, n, m\}$

Вариант 5. $\Sigma = \{d, b, c, m, k\}$

Вариант 6. $\Sigma = \{b, c, s, k, l\}$

Вариант 7. $\Sigma = \{b, c, n, m, k\}$

Вариант 8. $\Sigma = \{c, k, f, n, l\}$

Вариант 9. $\Sigma = \{f, h, k, g, l\}$

Вариант 10. $\Sigma = \{a, b, c, f, i\}$

Вариант 11. $\Sigma = \{a, b, c, j, i\}$

Вариант 12. $\Sigma = \{a, b, c, k, l\}$

Вариант 13. $\Sigma = \{a, d, c, j, i\}$

Вариант 14. $\Sigma = \{a, d, c, k, i\}$

Вариант 15. $\Sigma = \{a, d, b, j, i\}$

Вариант 16. $\Sigma = \{a, d, b, k, f\}$

Вариант 17. $\Sigma = \{a, b, c, k, m\}$

Вариант 18. $\Sigma = \{a, c, k, j, i\}$

Вариант 19. $\Sigma = \{a, c, n, m, f\}$

Вариант 20. $\Sigma = \{a, c, h, g, k\}$

Задание 3. Устранить левую рекурсию у грамматики с заданной схемой.
Представить результирующую грамматику.

1. $S \rightarrow Aab; A \rightarrow Sabc; A \rightarrow abc$
2. $S \rightarrow Aa; A \rightarrow SAb; A \rightarrow aa$
3. $S \rightarrow Aaa; A \rightarrow SaAb; A \rightarrow cAc$
4. $S \rightarrow Aa; A \rightarrow b; A \rightarrow Cc; C \rightarrow Sc$
5. $S \rightarrow Aa; A \rightarrow b; A \rightarrow Cc; C \rightarrow c$
6. $S \rightarrow Aa; A \rightarrow Bb; B \rightarrow Saa; B \rightarrow c$
7. $S \rightarrow AabA; A \rightarrow Sabc; A \rightarrow c$
8. $S \rightarrow AaA; A \rightarrow SbA; A \rightarrow aaa$
9. $S \rightarrow aA; A \rightarrow Bb; B \rightarrow Acd; B \rightarrow c$
10. $S \rightarrow aA; A \rightarrow Bb; B \rightarrow Cc; C \rightarrow Ac; C \rightarrow c$
11. $S \rightarrow aA; A \rightarrow Bab; B \rightarrow Cac; C \rightarrow Ac; C \rightarrow c$
12. $S \rightarrow AaA; A \rightarrow Bb; A \rightarrow Sb; B \rightarrow Sa; B \rightarrow c$
13. $S \rightarrow Aab; A \rightarrow Bab; B \rightarrow Sabc; B \rightarrow c$
14. $S \rightarrow Aa; A \rightarrow Bcaa; B \rightarrow Sbssa; B \rightarrow aa$
15. $S \rightarrow Aa; A \rightarrow Bd; A \rightarrow Sfs; B \rightarrow Sba; B \rightarrow ccc$
16. $S \rightarrow Aa; A \rightarrow Ba; B \rightarrow Sb; B \rightarrow BaCc; B \rightarrow dc; C \rightarrow c$
17. $S \rightarrow Aav; A \rightarrow Bas; B \rightarrow Bb; B \rightarrow Sc; B \rightarrow c$
18. $S \rightarrow Aa; A \rightarrow BabB; B \rightarrow Sb; B \rightarrow aad$
19. $S \rightarrow abcA; A \rightarrow BabB; B \rightarrow Aabc; B \rightarrow c$
20. $S \rightarrow aA; A \rightarrow Bab; B \rightarrow Ab; A \rightarrow aa$
21. $S \rightarrow AaA; A \rightarrow Ba; B \rightarrow Sb; S \rightarrow cd$
22. $S \rightarrow AaA; A \rightarrow Bba; A \rightarrow Sb; B \rightarrow Sc; B \rightarrow c; A \rightarrow aaa$
23. $S \rightarrow aA; A \rightarrow BdB; B \rightarrow Ab; A \rightarrow Ac; B \rightarrow c$
- 24.** $S \rightarrow AaA; A \rightarrow BaA; B \rightarrow Ab; B \rightarrow ass$

Задание 4. Для КС-грамматики в нормальной форме Хомского и цепочки, принадлежащей языку, порождаемому данной грамматикой, построить таблицу разбора и левый разбор с использованием метода Кока-Янгера-Касами.

Задание 5. Для КС-грамматики и цепочки, принадлежащей языку, порождаемому данной грамматикой, построить таблицу разбора и правый разбор методом Эрли.

Задание 6. Дана КС–грамматика.

1. Проверить, что она принадлежит к классу LL(1) грамматик.
2. Построить для грамматики таблицу разбора.
3. Для цепочки, принадлежащей языку, порождаемому данной грамматикой, построить историю разбора цепочки.
4. Для цепочки, не принадлежащей языку, порождаемому грамматикой, построить историю разбора. Для этого модифицировать таблицу разбора, указав диагностические сообщения об ошибке в требуемых клетках таблицы.

Задание 7. Для КС–грамматика класса LL(1) определить таблицу разбора, управляющую анализатором, продолжающим разбор при обнаружении ошибок. Для этого определить синхронизирующие множества и действия при ошибках. Построить историю разбора цепочки с ошибками.

Задание 8. Для КС–грамматика класса LL(1) построить анализатор методом рекурсивного спуска. Анализатор должен продолжать разбор в случае наличия ошибок

Задание 9. Дана КС–грамматика.

1. Проверить, что она принадлежит к классу грамматик простого предшествования.
2. Построить для грамматики отношения простого предшествования.
3. Для цепочки, принадлежащей языку, порождаемому данной грамматикой, построить историю разбора цепочки.
4. Для цепочки, не принадлежащей языку, порождаемому грамматикой, построить историю разбора. Определить диагностические сообщения об ошибках.