


МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
Программирования и информационных технологий
наименование кафедры, отвечающей за реализацию дисциплины


_____ проф. Махортов С.Д.,
подпись, расшифровка подписи
03.05.2023 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.15 Лингвистические основы информатики
Код и наименование дисциплины в соответствии с учебным планом

1. Код и наименование направления подготовки/специальности:

02.03.01 Математика и компьютерные науки

2. Профиль подготовки/специализация:

Распределенные системы и искусственный интеллект

3. Квалификация (степень) выпускника: бакалавр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: ПиИТ

6. Составители программы: Вахтин А.А., к.ф-м.н, доц.
(ФИО, ученая степень, ученое звание)

7. Рекомендована: НМС ФКН протокол № 7 от 03.05.2023

(наименование рекомендующей структуры, дата, номер протокола,

отметки о продлении вносятся вручную)

8. Учебный год: 2026/2027

Семестр(ы): 7

9. Цели и задачи учебной дисциплины:

Целями освоения учебной дисциплины являются: изучение студентами видов и парадигм языков программирования. Основных принципов построения языков программирования и элементов теории компиляторов.

Задачи учебной дисциплины: приобретение навыков построения контекстно-свободных языков программирования, программ синтаксического и лексического анализа, знакомство с алгоритмами и методами обработки контекстно-свободных формальных языков.

10. Место учебной дисциплины в структуре ООП: Дисциплина относится к вариативной части учебного плана. К началу курса студенты должны владеть основами программирования.

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):

Код и название компетенции	Код и название индикатора компетенции	Знания, умения, навыки
ПК-3. Способен создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе, с учетом возможностей современных информационных технологий и программирования и компьютерной техники.	ПК-3.1. Знает основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения, администрирования и развития (эволюции)	знать: Виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов. уметь: Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа. владеть: Алгоритмами и методами обработки контекстно-свободных формальных языков.
	ПК-3.2. Умеет использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта	знать: Виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов. уметь: Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа. владеть: Алгоритмами и методами обработки контекстно-свободных формальных языков.
	ПК-3.3. Имеет практический опыт применения указанных выше методов и технологий	знать: виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов. уметь: строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.

		<p>владеть: алгоритмами и методами обработки контекстно-свободных формальных языков.</p>
<p>ПК-4. Способен использовать современные методы разработки и реализации алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования.</p>	<p>ПК-4.1. Знает современные методы разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования</p>	<p>знать: Виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов.</p> <p>уметь: Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.</p> <p>владеть: Алгоритмами и методами обработки контекстно-свободных формальных языков.</p>
	<p>ПК-4.2. Умеет разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования</p>	<p>знать: Виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов.</p> <p>уметь: Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.</p> <p>владеть: Алгоритмами и методами обработки контекстно-свободных формальных языков.</p>
	<p>ПК-4.3. Имеет практический опыт разработки и реализации алгоритмов на базе языков и пакетов прикладных программ моделирования</p>	<p>знать: Виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов.</p> <p>уметь: Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.</p> <p>владеть: Алгоритмами и методами обработки контекстно-свободных формальных языков.</p>

12. Объем дисциплины в зачетных единицах/час — 2/72.

Форма промежуточной аттестации зачет с оценкой.

13. Виды учебной работы

Вид учебной работы	Трудоемкость			
	Всего	По семестрам		
		8 семестр		
Аудиторные занятия	24	24		
в том числе: лекции	12	12		
практические	—	—		
лабораторные	12	12		
Самостоятельная работа	48	48		
Форма промежуточной аттестации (зачет – 0 час. / экзамен – __ час.)	—	—		
Итого:	72	72		

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Лекции		
1.1	Роль языков программирования. Виды, аспекты, парадигмы.	Виды языков. Роль искусственных языков в информационных системах. Свойства хорошего языка. Синтаксис и семантика. Парадигмы языка.
1.2	Формальные грамматики и языки.	Алфавит и цепочка. Свойства цепочек. Формальные грамматики и языки. Классификация формальных грамматик по Н. Хомскому. Соотношения между типами грамматик. Разбор цепочек по КС-грамматикам. Дерево вывода. Неоднозначность грамматик. Эквивалентные преобразования грамматик. Удаление бесполезных символов. Преобразование укорачивающей КС-грамматики. Исключение цепных правил. Устранение прямой левой рекурсии. Левая факторизация. Нормальная форма Хомского.
1.3	Введение в компиляцию. Структура компилятора.	Компиляторы. Фазы компилятора. Окружение компилятора. Проблемы создания компиляторов.
1.4	Алгоритмы и методы лексического анализа.	Роль лексического анализатора в компиляторе. Структура и виды лексических анализаторов. Лексемы. Типы лексем. Лексические ошибки. Регулярные выражения и определения. Диаграммы переходов. Недетерминированные конечные автоматы. Построение НКА по регулярному выражению. Детерминированный конечный автомат. Преобразование НКА в ДКА. Построение ДКА по регулярному выражению. Минимизация количества состояний в ДКА.
1.5	Алгоритмы и методы синтаксического анализа.	Роль синтаксического анализатора в компиляторе. Структура и виды синтаксических анализаторов. Связь синтаксического анализа с разработкой грамматики языка. Нисходящий синтаксический анализ методом рекурсивного спуска. Предиктивные синтаксические анализаторы. Диаграммы переходов предиктивных синтаксических анализаторов. Нерекурсивный предиктивный анализ. Восходящий синтаксический анализ. Стековая реализация LR-анализатора. Активные префиксы. Конфликты в процессе анализа.
1.6	Семантический анализ. Проверка типов.	Синтаксически управляемые определения. Синтезируемые и наследуемые атрибуты. Синтезируемые атрибуты в синтаксическом анализаторе. Графы зависимости. Направленные ациклические графы выражений. Преобразование типов данных.

		Роль типизации в программировании. Системы типов. Спецификация проверки типов. Эквивалентность выражений типа. Преобразование типов. Перегрузка функций и операторов. Полиморфные функции.
1.7	Генерация промежуточного кода.	Языки промежуточных представлений программного кода. Графическое представление. Трехадресный код. Типы трехадресных инструкций. Объявления. Инструкции. Выражения. Технология обратных поправок.
1.8	Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Таблицы переменных для вложенных процедур. Реализация критерия области видимости глобальных переменных. Представление массивов и записей в таблице переменных. Таблицы переменных для объектов. Реализация основных критериев ООП: наследование, инкапсуляция, полиморфизм. Таблица виртуальных методов.
2. Лабораторные работы		
2.1	Формальные грамматики и языки.	Алфавит и цепочка. Свойства цепочек. Формальные грамматики и языки. Классификация формальных грамматик по Н. Хомскому. Соотношения между типами грамматик. Разбор цепочек по КС-грамматикам. Дерево вывода. Неоднозначность грамматик. Эквивалентные преобразования грамматик. Удаление бесполезных символов. Преобразование укорачивающей КС-грамматики. Исключение цепных правил. Устранение прямой левой рекурсии. Левая факторизация. Нормальная форма Хомского.
2.2	Алгоритмы и методы лексического анализа.	Роль лексического анализатора в компиляторе. Структура и виды лексических анализаторов. Лексемы. Типы лексем. Лексические ошибки. Регулярные выражения и определения. Диаграммы переходов. Недетерминированные конечные автоматы. Построение НКА по регулярному выражению. Детерминированный конечный автомат. Преобразование НКА в ДКА. Построение ДКА по регулярному выражению. Минимизация количества состояний в ДКА.
2.3	Алгоритмы и методы синтаксического анализа.	Роль синтаксического анализатора в компиляторе. Структура и виды синтаксических анализаторов. Связь синтаксического анализа с разработкой грамматики языка. Нисходящий синтаксический анализ методом рекурсивного спуска. Предиктивные синтаксические анализаторы. Диаграммы переходов предиктивных синтаксических анализаторов. Нерекурсивный предиктивный анализ. Восходящий синтаксический анализ. Стековая реализация LR-анализатора. Активные префиксы. Конфликты в процессе анализа.
2.4	Семантический анализ. Проверка типов.	Синтаксически управляемые определения. Синтезируемые и наследуемые атрибуты. Синтезируемые атрибуты в синтаксическом анализаторе. Графы зависимости. Направленные ациклические графы выражений. Преобразование типов данных. Роль типизации в программировании. Системы типов. Спецификация проверки типов. Эквивалентность выражений типа. Преобразование типов. Перегрузка функций и операторов. Полиморфные функции.
2.5	Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Таблицы переменных для вложенных процедур. Реализация критерия области видимости глобальных переменных. Представление массивов и записей в таблице переменных. Таблицы переменных для объектов. Реализация основных критериев ООП: наследование, инкапсуляция, полиморфизм. Таблица виртуальных методов.

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Роль языков программирования. Виды, аспекты, парадигмы.	1	—	0	6	7
2	Формальные грамматики и языки.	2	—	2	6	10
3	Введение в компиляцию. Структура компилятора.	2	—	0	6	8
4	Алгоритмы и методы лексического анализа.	2	—	4	6	12
5	Алгоритмы и методы синтаксического анализа.	2	—	2	6	10
6	Семантический анализ. Проверка типов.	1	—	2	6	9
7	Генерация промежуточного кода.	1	—	0	6	7
8	Таблицы переменных для вложенных процедур, массивов, записей и объектов.	1	—	2	6	9
	Итого:	12	—	12	48	72

14. Методические указания для обучающихся по освоению дисциплины

Большая часть часов выделена на самостоятельную работу и лабораторные работы, форма промежуточной аттестации – зачет с оценкой, рекомендуется студентам читать рекомендованную литературу, во время проверки выполнения лабораторных работ, преподавателю рекомендуется проводить теоретический опрос с целью определения степени усвоения материала, чтобы в дальнейшем вывести оценку к промежуточной аттестации

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1	Опалева Э. А. Языки программирования и методы трансляции / Э. А. Опалева, В. П. Самойленко. – СПб.: БХВ-Петербург, 2005. – 480 с.
2	Серебряков В. А. Основы конструирования компиляторов / В. А. Серебряков, М. П. Галочкин. – М.: Эдиториал УРСС, 2001. – 224 с.
3	Ахо А. В. Компиляторы: принципы, технологии и инструменты / А. В. Ахо, Р. Сети, Д. Д. Ульман.: Пер. с англ. – М.: Вильямс, 2001. – 768 с.
4	Маккиман У. Генератор компиляторов / У. Маккиман, Дж. Хорнинг, Д. Уортман; Пер. с англ. Круговой С. М.; Под ред. И с предисл. Савинкова В. М. – М.: Статистика, 1980. – 527 с.

б) дополнительная литература:

№ п/п	Источник
1	Гладкий А. В. Формальные грамматики и языки / А. В. Гладкий. – М.: Наука, 1973. – 368 с.
2	Мозговой М.В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. Практический подход М.В. Мозговой. – СПб.: Наука и Техника, 2006. – 320 с.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)*:

№ п/п	Ресурс
1.	https://lib.vsu.ru/?p=4&t=8
2.	JFLAP Version 7.0. Tutorial http://www.jflap.org/tutorial/
3.	Вурт Н. Построение компиляторов http://e.lanbook.com/books/element.php?pl1_id=1262
4.	Федосеева Л.И. Адипов Р.М. Шмокин М.Н. Основы теории конечных автоматов и формальных языков http://e.lanbook.com/books/element.php?pl1_id=62703

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	<i>JFLAP Version 7.0. Tutorial</i> http://www.jflap.org/tutorial/
2	Опалева Э. А. <i>Языки программирования и методы трансляции</i> / Э. А. Опалева, В. П. Самойленко. – СПб.: БХВ-Петербург, 2005. – 480 с.
3	Серебряков В. А. <i>Основы конструирования компиляторов</i> / В. А. Серебряков, М. П. Галочкин. – М.: Эдиториал УРСС, 2001. – 224 с.
4	Вирт Н. <i>Построение компиляторов</i> http://e.lanbook.com/books/element.php?pl1_id=1262
5	Федосеева Л.И. Адилов Р.М. Шмокин М.Н. <i>Основы теории конечных автоматов и формальных языков</i> http://e.lanbook.com/books/element.php?pl1_id=62703

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)

Любой язык программирования, для реализации алгоритмов. Эмулятор конечных автоматов JFLAP.

18. Материально-техническое обеспечение дисциплины:

Доска с проектором, доска с фломастером, для лабораторных и практических занятий компьютерный класс с установленным языком программирования и виртуальной машиной JAVA.

19. Фонд оценочных средств:

19.1. Перечень компетенций с указанием этапов формирования и планируемых результатов обучения

Код и содержание компетенции (или ее части)	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенции посредством формирования знаний, умений, навыков)	Этапы формирования компетенции (разделы (темы) дисциплины или модуля и их наименование)	ФОС* (средства оценивания)
ПК-3.1. Знает основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения, администрирования и развития (эволюции)	Знать: виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов.	1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Лабораторные работы, Эссе
	Уметь: Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.	1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка	Лабораторные работы, Эссе

		<p>типов.</p> <p>7. Генерация промежуточного кода</p> <p>8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.</p>	
	<p>Владеть: алгоритмами и методами обработки контекстно-свободных формальных языков.</p>	<p>1. Роль языков программирования. Виды, аспекты, парадигмы.</p> <p>2. Формальные грамматики и языки</p> <p>3. Введение в компиляцию. Структура компилятора.</p> <p>4. Алгоритмы и методы лексического анализа.</p> <p>5. Алгоритмы и методы синтаксического анализа.</p> <p>6. Семантический анализ. Проверка типов.</p> <p>7. Генерация промежуточного кода</p> <p>8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.</p>	<p>Лабораторные работы, Эссе</p>
<p>ПК-3.2. Умеет использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта</p>	<p>Знать: виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов.</p>	<p>1. Роль языков программирования. Виды, аспекты, парадигмы.</p> <p>2. Формальные грамматики и языки</p> <p>3. Введение в компиляцию. Структура компилятора.</p> <p>4. Алгоритмы и методы лексического анализа.</p> <p>5. Алгоритмы и методы синтаксического анализа.</p> <p>6. Семантический анализ. Проверка типов.</p> <p>7. Генерация промежуточного кода</p> <p>8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.</p>	<p>Лабораторные работы, Эссе</p>
	<p>Уметь: строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.</p>	<p>1. Роль языков программирования. Виды, аспекты, парадигмы.</p> <p>2. Формальные грамматики и языки</p> <p>3. Введение в компиляцию. Структура компилятора.</p> <p>4. Алгоритмы и методы лексического анализа.</p> <p>5. Алгоритмы и методы синтаксического анализа.</p> <p>6. Семантический анализ. Проверка типов.</p> <p>7. Генерация промежуточного кода</p> <p>8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.</p>	<p>Лабораторные работы, Эссе</p>
	<p>Владеть: алгоритмами и методами обработки контекстно-свободных формальных языков.</p>	<p>1. Роль языков программирования. Виды, аспекты, парадигмы.</p> <p>2. Формальные грамматики и языки</p> <p>3. Введение в компиляцию. Структура компилятора.</p> <p>4. Алгоритмы и методы лексического анализа.</p> <p>5. Алгоритмы и методы синтаксического анализа.</p> <p>6. Семантический анализ. Проверка типов.</p> <p>7. Генерация промежуточного кода</p> <p>8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.</p>	<p>Лабораторные работы, Эссе</p>
	<p>Знать: виды и парадигмы языков программирования. Основные принципы построения языков</p>	<p>1. Роль языков программирования. Виды, аспекты, парадигмы.</p> <p>2. Формальные грамматики и языки</p> <p>3. Введение в компиляцию. Структура компилятора.</p> <p>4. Алгоритмы и методы лексического</p>	<p>Лабораторные работы, Эссе</p>
<p>ПК-3.3. Имеет практический опыт применения указанных выше методов и</p>	<p>Знать: виды и парадигмы языков программирования. Основные принципы построения языков</p>	<p>1. Роль языков программирования. Виды, аспекты, парадигмы.</p> <p>2. Формальные грамматики и языки</p> <p>3. Введение в компиляцию. Структура компилятора.</p> <p>4. Алгоритмы и методы лексического</p>	<p>Лабораторные работы, Эссе</p>

технологий	программирования и элементы теории компиляторов.	анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.	
	Уметь: строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.	1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Лабораторные работы, Эссе
	Владеть: алгоритмами и методами обработки контекстно-свободных формальных языков.	1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Лабораторные работы, Эссе
ПК-4.1. Знает современные методы разработки и реализации алгоритмов математических моделей на базе языков и пакетов прикладных программ моделирования	Знать: виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов.	1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Лабораторные работы, Эссе
	Уметь: строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.	1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Лабораторные работы, Эссе
	Владеть: алгоритмами и методами обработки	1. Роль языков программирования. Виды, аспекты, парадигмы.	Лабораторные работы,

	контекстно-свободных формальных языков.	<ol style="list-style-type: none"> 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов. 	Эссе
ПК-4.2. Умеет разрабатывать и реализовывать алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования	Знать: виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов.	<ol style="list-style-type: none"> 1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов. 	Лабораторные работы, Эссе
	Уметь: строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.	<ol style="list-style-type: none"> 1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов. 	Лабораторные работы, Эссе
	Владеть: алгоритмами и методами обработки контекстно-свободных формальных языков.	<ol style="list-style-type: none"> 1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов. 	Лабораторные работы, Эссе
ПК-4.3. Имеет практический опыт разработки и реализации алгоритмов на базе языков и пакетов прикладных программ моделирования	Знать: виды и парадигмы языков программирования. Основные принципы построения языков программирования и элементы теории компиляторов.	<ol style="list-style-type: none"> 1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для 	Лабораторные работы, Эссе

		вложенных процедур, массивов, записей и объектов.	
	Уметь: строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа.	1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Лабораторные работы, Эссе
	Владеть: алгоритмами и методами обработки контекстно-свободных формальных языков.	1. Роль языков программирования. Виды, аспекты, парадигмы. 2. Формальные грамматики и языки 3. Введение в компиляцию. Структура компилятора. 4. Алгоритмы и методы лексического анализа. 5. Алгоритмы и методы синтаксического анализа. 6. Семантический анализ. Проверка типов. 7. Генерация промежуточного кода 8. Таблицы переменных для вложенных процедур, массивов, записей и объектов.	Лабораторные работы, Эссе
Промежуточная аттестация			КИМ

19.2 Описание критериев и шкалы оценивания компетенций (результатов обучения) при промежуточной аттестации

Владение теоретическими основами дисциплины, способность иллюстрировать ответ примерами, фактами, данными научных исследований, применять теоретические знания для решения практических задач.

Для оценивания результатов обучения на зачете с оценкой используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<i>Обучающийся в полной мере владеет теоретическими основами дисциплины, способен иллюстрировать ответ примерами, фактами, данными научных исследований, применять теоретические знания для решения практических задач.</i>	<i>Повышенный уровень</i>	<i>Отлично</i>
<i>Ответ на контрольно-измерительный материал не соответствует одному (двум) из перечисленных показателей, но обучающийся дает правильные ответы на дополнительные вопросы. Недостаточно продемонстрировано, или содержатся отдельные пробелы....., ИЛИ Обучающийся владеет понятийным аппаратом данной области науки (теоретическими основами дисциплины), способен, допускает ошибки при.....</i>	<i>Базовый уровень</i>	<i>Хорошо</i>
<i>Ответ на контрольно-измерительный материал не соответствует любым двум(трем) из перечисленных показателей, обучающийся дает неполные ответы на</i>	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>

<p><i>дополнительные вопросы. Демонстрирует частичные знания..., или не умеет..., или имеет не полное представление..., допускает существенные ошибки... ИЛИ Обучающийся владеет частично теоретическими основами дисциплины, фрагментарно способен, не умеет применять.... Не умеет</i></p>		
<p><i>Ответ на контрольно-измерительный материал не соответствует любым трем(четырем) из перечисленных показателей. Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки.....</i></p>	–	<i>Неудовлетворительно</i>

19.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

19.3.1 Перечень вопросов к эссе (зачету):

№ Текст вопроса

1. Алфавит, цепочка, формальные грамматики и языки. Основные свойства и определения.
2. Классификация грамматик по Н. Хомскому. Иерархия классов.
3. Разбор цепочек по КС-грамматикам. Неоднозначность КС-грамматик.
4. Алгоритм удаления бесполезных символов в КС-грамматиках.
5. Алгоритм преобразования укорачивающей КС-грамматики в НКС-грамматику.
6. Алгоритм исключения цепных правил в КС-грамматиках.
7. Алгоритм устранения прямой левой рекурсии в КС-грамматиках.
8. Алгоритм левой факторизации КС-грамматики.
9. Алгоритм преобразования КС-грамматики в нормальную форму Хомского
10. Лексический анализ. Задачи лексического анализатора. Виды лексических анализаторов. Лексемы (определение, виды).
11. Регулярные выражения. Правила построений РВ. Синтаксическое дерево регулярного выражения. Диаграмма переходов (определение, пример).
12. Недетерминированный конечный автомат (определение). Алгоритм построения НКА по регулярному выражению.
13. Детерминированный конечный автомат. Алгоритм построения ДКА по НКА.
14. Алгоритм построения ДКА по регулярному выражению.
15. Алгоритм минимизации количества состояний в ДКА.
16. Синтаксический анализ. Роль синтаксического анализатора. Методы синтаксического анализа и их характеристики.
17. Нисходящий синтаксический анализ методом рекурсивного спуска.
18. Алгоритм нерекурсивного предиктивного синтаксического анализа.
19. Алгоритм построения таблиц для предиктивного синтаксического анализа.
20. Восстановление после ошибок в нерекурсивном предиктивном анализаторе.
21. Восходящий синтаксический анализ. Алгоритм LR-анализа.
22. Восстановление после ошибок в восходящем синтаксическом LR-анализаторе.

23. Атрибуты. Виды, определения.
24. Графы зависимости. Порядок выполнения.
25. Вычисление S-атрибутивных определений. Их роль в синтаксическом анализаторе.
26. L-атрибутивные определения. Схемы трансляции.
27. Генератор промежуточного кода.
28. Виды представлений промежуточного кода. Дерево разбора.
29. Виды представлений промежуточного кода. Направленный ациклический граф
30. Виды представлений промежуточного кода. Трехадресный код
31. Виды представлений промежуточного кода. Префиксная запись кода.
32. Виды представлений промежуточного кода. Постфиксная запись кода.
33. Трехадресный код. Виды трехадресных инструкций.
34. Трехадресный код. Методы реализации трехадресного кода (четверки, тройки).
35. Таблицы символов для вложенных процедур.
36. Таблицы символов для записей.
37. Таблицы символов для объектов.
38. Оптимизация кода. Свойства оптимизирующего компилятора.
39. Методы повышения производительности программного кода.
40. Преобразования кода сохраняющие функции.
41. Общее подвыражение. Методы устранения общих подвыражений в коде.
42. Размножение копий. Назначение, методы, цель
43. Дублирование констант. Назначение, методы, цель.
44. Беспольный код. Методы выявления бесполезного кода.
45. Методы оптимизации циклов.
46. Выражения инвариантные относительно цикла. Цель их выявления.
47. Индуктивные переменные в цикле. Цель их выявления в методах оптимизации.
48. Граф потока. Его назначение.
49. Назначение доминаторов в графе потока.
50. естественные циклы в графе потока.
51. внутренние циклы в графе потока.

19.3.2 Перечень лабораторных заданий

Комплект заданий № 1

1. Постройте контекстно-свободную грамматику для формирования цепочки символов алфавита $\Sigma = \{L, O, P, K, S\}$, где L – локомотив, O – общий вагон, P – плацкартный вагон, K – вагон-купе, S – спальня вагон люкс. Вагоны разных категорий должны следовать друг за другом. Локомотивов может быть несколько, и они могут располагаться вначале, в середине или в конце поезда. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.
2. Постройте контекстно-свободную грамматику для формирования римских цифр. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.

3. Постройте контекстно-свободную грамматику для формирования арифметических выражений. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.
4. Постройте контекстно-свободную грамматику для формирования логических выражений. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.
5. Постройте контекстно-свободную грамматику для формирования условного оператора. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.
6. Постройте контекстно-свободную грамматику для формирования цикла while. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.
7. Постройте контекстно-свободную грамматику для формирования цикла for. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.
8. Постройте контекстно-свободную грамматику для формирования цикла repeat. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.
9. Постройте контекстно-свободную грамматику для формирования описания переменных. Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.
10. Постройте контекстно-свободную грамматику для формирования структуры записи (record, structure). Реализуйте конечный автомат распознавания цепочек данной грамматики в JFlap.

Комплект заданий № 2

1. Реализуйте программу, выполняющую преобразование контекстно-свободной грамматики в эквивалентную грамматику, не содержащую бесполезных символов. Проверьте правильность работы на контекстно-свободной грамматике $G = (V_T, V_N, P, S)$, где $V_T = \{a, b, c, d, e, f\}$, $V_N = \{A, B, C, D, E, F, S\}$, и правила:
 - a) $P = \{S \rightarrow b | C | cCB, A \rightarrow e | Ab, B \rightarrow Bb | cB, C \rightarrow Ca | Bf | d\}$;
 - b) $P = \{S \rightarrow aC | bA, A \rightarrow cAB, B \rightarrow aC, C \rightarrow bA | d\}$;
 - c) $P = \{S \rightarrow aABC | aE, A \rightarrow SCD | c, B \rightarrow bFD |, C \rightarrow aE, D \rightarrow aD, E \rightarrow aCE | a, F \rightarrow AB\}$;
 - d) $P = \{S \rightarrow aA, A \rightarrow aA | b | cC, B \rightarrow a | cB, C \rightarrow bAC\}$.
2. Реализуйте программу, выполняющую преобразование контекстно-свободной грамматики в эквивалентную неукорачивающую контекстно-свободную грамматику. Проверьте правильность работы на контекстно-свободной грамматике $G = (V_T, V_N, P, S)$, где:
 - a) $V_T = \{a, b\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow AB, A \rightarrow SA | BB | bB, B \rightarrow b | aA, B \rightarrow \varepsilon\}$;
 - b) $V_T = \{a, b, c, d\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow Aa | bB, A \rightarrow cAdA | a | \varepsilon, B \rightarrow cBdd | \varepsilon\}$;
 - c) $V_T = \{a, b\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow bA, A \rightarrow bA | aB | \varepsilon, B \rightarrow bB | \varepsilon\}$;
 - d) $V_T = \{a, b, c\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow aAB | bA | \varepsilon, A \rightarrow aAB | \varepsilon, B \rightarrow bB | c\}$;
 - e) $V_T = \{a, b, c, d\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow aAB | bBS | \varepsilon, A \rightarrow cBS | \varepsilon, B \rightarrow dB | \varepsilon\}$;
3. Реализуйте программу, выполняющую преобразование контекстно-свободной грамматики в эквивалентную грамматику, не содержащую цепных правил. Проверьте правильность работы на контекстно-свободной грамматике $G = (V_T, V_N, P, S)$, где:
 - a) $V_T = \{i, :, =, (,)\}$, $V_N = \{A, B, L, P, F, Q, S\}$, $P = \{S \rightarrow LA | LB, L \rightarrow P :=, L \rightarrow Q :=, P \rightarrow i, A \rightarrow F, Q \rightarrow i, B \rightarrow F, F \rightarrow Q(i)\}$;
 - b) $V_T = \{a, i\}$, $V_N = \{A, B, C, D, S\}$, $P = \{S \rightarrow AC, A \rightarrow B | AaB, B \rightarrow i, C \rightarrow D | DaC, D \rightarrow i\}$;

- c) $V_T = \{1, 0\}$, $V_N = \{A, B, C, S\}$, $P = \{S \rightarrow 1A \mid B0, A \rightarrow 1A \mid C, B \rightarrow B0 \mid C, C \rightarrow 1C0 \mid \varepsilon\}$;
- d) $V_T = \{+, *, i\}$, $V_N = \{P, T, S\}$, $P = \{S \rightarrow T + P \mid T, T \rightarrow T * P \mid P, P \rightarrow i\}$;
- e) $V_T = \{1, 0, a, b\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow A \mid B, A \rightarrow 1A0 \mid 1a0, B \rightarrow 1B00 \mid 1b00\}$.
4. Реализуйте программу, исключаящую левую рекурсию в контекстно-свободной грамматике. Проверьте правильность работы на контекстно-свободной грамматике $G = (V_T, V_N, P, S)$, где:
- a) $V_T = \{a, b\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow Ba \mid Ab, A \rightarrow Sa \mid AAb \mid a, B \rightarrow Sb, B \rightarrow BBa \mid b\}$;
- b) $V_T = \{a, b\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow AB \mid a, A \rightarrow BS \mid Sb, B \rightarrow SA \mid BB, B \rightarrow a\}$;
- c) $V_T = \{a, b, c\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow Ab, A \rightarrow Sa \mid cB, B \rightarrow bS \mid c\}$;
- d) $V_T = \{a, b\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow AB, A \rightarrow SA \mid BB \mid bB, B \rightarrow b \mid aA, B \rightarrow \varepsilon\}$;
- e) $V_T = \{a, b, c, d\}$, $V_N = \{A, S\}$, $P = \{S \rightarrow SaA \mid AA \mid b, A \rightarrow ASa \mid Ad \mid c\}$.
5. Реализуйте программу преобразования контекстно-свободной грамматики в нормальную форму Хомского. Проверьте правильность работы на контекстно-свободной грамматике $G = (V_T, V_N, P, S)$, где:
- a) $V_T = \{1, 0, a, b\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow AB, A \rightarrow SA \mid BB \mid bB, B \rightarrow b, B \rightarrow aA \mid \varepsilon\}$;
- b) $V_T = \{a, b, c, d\}$, $V_N = \{A, B, S\}$, $P = \{S \rightarrow Aa \mid bB, A \rightarrow cAdA \mid a \mid \varepsilon, B \rightarrow cBdd \mid \varepsilon\}$;
- c) $V_T = \{1, 0\}$, $V_N = \{A, S\}$, $P = \{S \rightarrow SS \mid 1A0, A \rightarrow 1A0 \mid \varepsilon\}$;
- d) $V_T = \{a, b, c, d\}$, $V_N = \{A, B, C, S\}$, $P = \{S \rightarrow aC \mid bA, A \rightarrow cAB, B \rightarrow aC, C \rightarrow bA \mid d\}$.

Комплект заданий № 3

1. Разработать и реализовать программу, которая может выделить из текста программы, написанной на языке Pascal служебные слова, идентификаторы, константы, и операнды. Программа должна составлять таблицу лексем, где для каждой лексемы хранится информация о типе лексемы и число повторений в тексте.
2. Разработать и реализовать программу, которая может выделить из текста программы, написанной на языке CLIPS операторы, факты, условия и действия. Программа должна составлять таблицу лексем, где для каждой лексемы хранится информация о типе лексемы, и к какому оператору она принадлежит.
3. Разработать и реализовать программу, которая может из записи числа римскими цифрами выделить единицы, десятки, сотни и т. п. (реализовать считывание текста с лева направо). Программа должна составлять соответствующую таблицу.
4. Разработать и реализовать программу, которая может по заданному выражению, состоящему из функций $\text{Min}(x_1, x_2, \dots, x_n)$ и $\text{Max}(x_1, x_2, \dots, x_n)$, выделить лексемы Min и Max и их аргументы. Программа должна составлять таблицу лексем, где аргументу будет соответствовать уровень вложенности в функции.
5. Разработать и реализовать программу, которая может выделять аргументы и операнды арифметических выражений языка программирования Pascal. Программа должна составлять таблицу лексем, где каждой лексеме соответствует уровень вложенности. Если операнды находятся не в скобках, то их уровень вложенности 0, если в первых скобках, то уровень – 1, и т. п.

6. Разработать и реализовать программу, которая может выделять аргументы и операнды логических выражений языка программирования Pascal. Программа должна составлять таблицу лексем, где каждой лексеме соответствует уровень вложенности. Если операнды находятся не в скобках, то их уровень вложенности 0, если в первых скобках, то уровень – 1, и т. п.
7. Разработать и реализовать программу, которая может выделить из текста на языке Pascal структуры записей и составить на них таблицу символов.
8. Разработать и реализовать программу, которая может выделить из текста на языке Pascal процедуры и функции, их идентификаторы (переменные, константы, типы данных) и составлять для них таблицы символов. Предусмотреть вложенность процедур и функций.
9. Разработать и реализовать программу, которая может выделить из текста на языке C++ структуры записей и составить на них таблицу символов.
10. Разработать и реализовать программу, которая может выделить из текста на языке C++ процедуры и функции, их идентификаторы (переменные, константы, типы данных) и составлять для них таблицы символов.

19.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины осуществляется в ходе текущей и промежуточной аттестаций.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в форме лабораторных работ. Критерии оценивания приведены выше.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Контрольно-измерительные материалы промежуточной аттестации включают в себя теоретические вопросы, позволяющие оценить уровень полученных знаний.

При оценивании используются количественные шкалы оценок. Критерии оценивания приведены выше.