

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

УТВЕРЖДАЮ

Заведующий кафедрой

ПиИТ

наименование кафедры, отвечающей за реализацию дисциплины



проф. Махортов С.Д.,
подпись, расшифровка подписи

05.04.2024 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.ДВ.01.02 Теория автоматов и формальных языков

Код и наименование дисциплины в соответствии с учебным планом

1. Код и наименование направления подготовки/специальности:

09.04.04 Программная инженерия

2. Профиль подготовки/специализация: Системное программирование

3. Квалификация выпускника: магистр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: ПиИТ

6. Составители программы: Вахтин А.А., к.ф.-м.н., доц.

7. Рекомендована: НМС ФКН протокол № 5 от 05.04.2024 г.

(наименование рекомендующей структуры, дата, номер протокола,

отметки о продлении вносятся вручную)

8. Учебный год: 2024/2025

Семестр(ы): 1

9. Цели и задачи учебной дисциплины

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

Задачи учебной дисциплины: приобретение навыков построения контекстно-свободных языков программирования, Знакомство с алгоритмами и методами обработки контекстно-свободных формальных языков.

10. Место учебной дисциплины в структуре ООП: учебная дисциплина является курсом по выбору, к началу курса данной дисциплины студенты должны знать и владеть основами программирования.

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

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ПК-1	Способен организовывать работу программистов в группе по созданию системного ПО	ПК-1.1	Выполняет декомпозицию поставленной задачи и распределение подзадач между программистами	знать: Основные принципы построения языков программирования и элементы теории компиляторов уметь: Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа владеть (иметь навык(и)): алгоритмами и методами обработки контекстно-свободных формальных языков.
		ПК-1.2	Определяет процессы интеграции разработанных компонентов системного ПО	знать: Основные принципы построения языков программирования и элементы теории компиляторов уметь: Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа владеть (иметь навык(и)): алгоритмами и методами обработки контекстно-свободных формальных языков.
		ПК-1.3	Определяет задачи для группы стандартов кодирования	знать: Основные принципы построения языков программирования и элементы теории компиляторов уметь:

				<p>Строить контекстно-свободные языки программирования, создавать программы синтаксического и лексического анализа</p> <p>владеть (иметь навык(и)): алгоритмами и методами обработки контекстно-свободных формальных языков.</p>
ПК-15	Способен разрабатывать и исследовать модели объектов профессиональной деятельности, предлагать и адаптировать методики решения научно-исследовательских задач, планировать и проводить исследования	ПК-15.1	Обеспечивает сбор научно-технической (научной) информации, необходимой для постановки и решения задач исследования	<p>Знать: основные методы и технологии теории автоматов и формальных языков</p> <p>Уметь: проводить сбор научно-технической информации, необходимой для постановки и решения задач разработки программ с использованием алгоритмов и методов теории автоматов и формальных языков</p> <p>Владеть: навыками сбора научно-технической информации, необходимой для постановки и решения задач разработки программ с использованием алгоритмов и методов теории автоматов и формальных языков</p>
ПК-16	Способен определять качество проводимых исследований, обрабатывать, интерпретировать и оформлять результаты проведенных исследований и представлять результаты профессиональному сообществу	ПК-16.1	Умеет обрабатывать данные проводимых исследований с использованием современных методов анализа информации и информационных технологий	<p>Знать: основные методы и технологии теории автоматов и формальных языков</p> <p>Уметь: обрабатывать данные проводимых исследований с использованием современных методов анализа информации и информационных технологий</p>

12. Объем дисциплины в зачетных единицах/час. (в соответствии с учебным планом) — 3/108.

Форма промежуточной аттестации экзамен

13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость			
		Всего	По семестрам		
			1 семестр		
Аудиторные занятия		36	36		
в том числе:	лекции	18	18		
	практические	0	0		

	лабораторные	18	18		
Самостоятельная работа		36	36		
в том числе: курсовая работа (проект)		0	0		
Форма промежуточной аттестации (экзамен – __ час.)		36	36		
Итого:		108	108		

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
1.1	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.	
1.2	Алгоритмы преобразования КС 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	Конечный автомат	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.	
1.4	Автоматы с магазинной памятью	Отличие конечных автоматов от автоматов с магазинной памятью. Предиктивные синтаксические анализаторы. Диаграммы переходов предиктивных синтаксических анализаторов. Нерекursивный предиктивный анализ. Восходящий синтаксический анализ. Стековая реализация LR-анализатора. Активные префиксы. Конфликты в процессе анализа.	
1.5	Семантический анализ	Синтаксически управляемые определения. Синтезируемые и наследуемые атрибуты. Синтезируемые атрибуты в синтаксическом анализаторе. Графы зависимости. Направленные ациклические графы выражений. Преобразование типов данных.	
1.6	Генерация промежуточного кода	Языки промежуточных представлений программного кода. Графическое представление. Трехадресный код. Типы трехадресных инструкций. Объявления. Инструкции. Выражения. Технология обратных поправок.	

1.7	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.	
1.8	Компиляторы компиляторов	Компиляторы компиляторов. Языки программирования компиляторов на примере Yacc и Buzon.	
2. Лабораторные занятия			
2.1	Формальные грамматики и языки	Построение формальных КС грамматик по форме Бэкуса-Наура.	
2.2	Алгоритмы преобразования КС грамматик	Реализация алгоритмов эквивалентного преобразования КС грамматик: - Удаление бесполезных символов - Преобразование укорачивающей КС-грамматики - Исключение цепных правил - Устранение прямой левой рекурсии - Левая факторизация - Нормальная форма Хомского.	
2.3	Конечный автомат	Недетерминированные конечные автоматы. Реализация алгоритмов построения НКА по регулярному выражению. Детерминированный конечный автомат. Реализация алгоритма преобразования НКА в ДКА. Алгоритмы построения ДКА по регулярному выражению. Алгоритм минимизации количества состояний в ДКА.	
2.4	Автоматы с магазинной памятью	Алгоритмы предиктивных синтаксических анализаторов. Нерекursивный предиктивный анализ. Восходящий синтаксический анализ. Стековая реализация LR-анализатора.	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1.	Формальные грамматики и языки	2		4	4	10
2.	Алгоритмы преобразования КС грамматик	2		4	6	12
3.	Конечный автомат	2		4	4	10
4.	Автоматы с магазинной памятью	2		6	6	14
5.	Семантический анализ	2		0	4	6
6.	Генерация промежуточного кода	2		0	4	6
7.	Таблицы переменных для вложенных процедур, массивов, записей и объектов	2		0	4	66

8.	Компиляторы компиляторов	4	0	4	8
	Итого:	18	18	36	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.	Вирт Н. Построение компиляторов http://e.lanbook.com/books/element.php?pl1_id=1262
4.	Федосеева Л.И. Адилов 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

(учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных), курсовых работ и др.)

№ 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.

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):

На лабораторных и практических занятиях допускается использовать студентами любой язык программирования и любую среду программирования.

18. Материально-техническое обеспечение дисциплины:

Мультимедийная лекционная аудитория, компьютерный класс, программное обеспечение: среда программирования Visual Studio.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Формальные грамматики и языки	ПК-1 ПК-15 ПК-16	ПК-1.1, ПК-1.2, ПК-1.3 ПК-15.1 ПК-16.1	Практическое задание
2.	Алгоритмы преобразования КС грамматик	ПК-1 ПК-15 ПК-16	ПК-1.1, ПК-1.2, ПК-1.3 ПК-15.1 ПК-16.1	Практическое задание
3.	Конечный автомат	ПК-1 ПК-15 ПК-16	ПК-1.1, ПК-1.2, ПК-1.3 ПК-15.1 ПК-16.1	Практическое задание
4.	Автоматы с магазинной памятью	ПК-1	ПК-1.1, ПК-1.2, ПК-1.3	Практическое задание
5.	Семантический анализ	ПК-1	ПК-1.1, ПК-1.2, ПК-1.3	Практическое задание
6.	Генерация промежуточного кода	ПК-1	ПК-1.1, ПК-1.2, ПК-1.3	Практическое задание
7.	Таблицы переменных для вложенных процедур, массивов, записей и объектов	ПК-1	ПК-1.1, ПК-1.2, ПК-1.3	Практическое задание
8.	Компиляторы компиляторов	ПК-1 ПК-15 ПК-16	ПК-1.1, ПК-1.2, ПК-1.3 ПК-15.1 ПК-16.1	Практическое задание
Промежуточная аттестация форма контроля – экзамен				Практическое задание

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости Контроль успеваемости по дисциплине осуществляется с помощью контроля и оценки следующих заданий лабораторных работ:

20.1.1 Перечень заданий лабораторных работ

Варианты заданий к лабораторной работе № 1.

1. Постройте контекстно-свободную грамматику для формирования цепочки символов алфавита, где Л – локомотив, О – общий вагон, П – плацкартный вагон, К – вагон-купе, С – спальный 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.

Варианты заданий к лабораторной работе № 2

1. Реализуйте программу, выполняющую преобразование контекстно-свободной грамматики в эквивалентную грамматику, не содержащую бесполезных символов. Проверьте правильность работы на контекстно-свободной грамматике $G(V, V_P, (T, N, S))$, где $V_T = \{a, b, c, d, e, f, , , , , \}$, $V_N = \{A, B, C, D, E, F, S\}$, и правила:

$$P \Rightarrow S \quad b \mid C \mid cCB, A \Rightarrow e \mid A, Bb \Rightarrow Bb \mid c \mid B, C \Rightarrow Ca \mid Bfd$$

$$Q \Rightarrow S \quad aC \mid bA, A \Rightarrow cAB, B \Rightarrow aC, C \Rightarrow bA \mid d$$

$$R \Rightarrow S \quad aABC \mid aE, A \Rightarrow SCD \mid c, B \Rightarrow bFD \mid , C \Rightarrow aE, D \Rightarrow aD,$$

$$E \Rightarrow aCE \mid a, F \Rightarrow AB;$$

$$P \Rightarrow S \quad aA, A \Rightarrow aA \mid b \mid C, B \Rightarrow a \mid B, C \Rightarrow bAC$$

2. Реализуйте программу, выполняющую преобразование контекстно-свободной грамматики в эквивалентную неукорачивающую контекстно-свободную грамматику. Проверьте правильность работы на контекстно-свободной грамматике $G(V, V_P, (T, N, S))$, где:

$$T = \{a, b, , \}, V_N = \{A, B, S\}, P = \{S \Rightarrow AB, A \Rightarrow SA \mid BB \mid bB, B \Rightarrow b \mid a \mid A, B \Rightarrow \}; V$$

$$T = \{a, b, c, d, , \}, V_N = \{A, B, S\}, P = \{S \Rightarrow Aa \mid b \mid B, A \Rightarrow c \mid d \mid A \mid a \mid ,$$

$$B \Rightarrow c \mid dd \mid B \mid \};$$

$V_T = \{a, b, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow bA, A \rightarrow bA \mid aB, B \rightarrow bB\}; V$

$V_T = \{a, b, c, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow aAB \mid bA, A \rightarrow aAB, B \rightarrow bB \mid c\}; V$

$V_T = \{a, b, c, d, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow aAB \mid bBS, A \rightarrow cBS, B \rightarrow dB\};$

3. Реализуйте программу, выполняющую преобразование контекстно-свободной грамматики в эквивалентную грамматику, не содержащую цепных правил. Проверьте правильность работы на контекстно-свободной грамматике $G \in VVP(\tau, N, S)$, где:

V

$V_T = \{i, :, (,)\}, V_N = \{A, B, L, P, F, Q, S\}, P = \{S \rightarrow LA \mid LB, L \rightarrow P : ,$

$L \rightarrow Q : , P \rightarrow i, A \rightarrow F, Q \rightarrow i, B \rightarrow F, F \rightarrow Q()i\};$

V

$V_T = \{a, i, \epsilon\}, V_N = \{A, B, C, D, S\}, P = \{S \rightarrow AC, A \rightarrow B \mid AB, Ba \rightarrow i,$

$C \rightarrow D \mid DC, Da \rightarrow i\};$

V

$V_T = \{1, 0\}, V_N = \{A, B, C, S\}, P = \{S \rightarrow 1A \mid B0, A \rightarrow 1A \mid C, B \rightarrow B0 \mid C,$

$C \rightarrow 1C0\};$

$V_T = \{ \epsilon, *, i\}, V_N = \{P, T, S\}, P = \{S \rightarrow T \mid P \mid T, T \rightarrow T * P \mid P, P \rightarrow i\};$

$V_T = \{1, 0, a, b, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow A \mid B, A \rightarrow 1A0 \mid 10, A^a \rightarrow B \mid B00 \mid 100b\};$

4. Реализуйте программу, исключаящую левую рекурсию в контекстно-свободной грамматике. Проверьте правильность работы на контекстно-свободной грамматике $G \in VVP(\tau, N, S)$, где:

$V_T = \{a, b, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow Ba \mid A, Ab \rightarrow Sa \mid AAb \mid a\}, B \rightarrow S, bB \rightarrow BBa \mid b\};$

$V_T = \{a, b, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow AB \mid a, A \rightarrow BS \mid S, Bb \rightarrow SA \mid BB, B \rightarrow a\};$

$V_T = \{a, b, c, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow A, Ab \rightarrow Sa \mid B, B \rightarrow bS \mid c\};$

$V_T = \{a, b, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow AB, A \rightarrow SA \mid BB \mid bB, B \rightarrow ba \mid A, B \rightarrow \epsilon\};$

$V_T = \{a, b, c, d, \epsilon\}, V_N = \{A, S\}, P = \{S \rightarrow SA \mid AA \mid a, A \rightarrow ASa \mid Ad \mid c\};$

Реализуйте программу преобразования контекстно-свободной грамматики в нормальную форму Хомского. Проверьте правильность работы на контекстно-свободной грамматике $G \in VVP(\tau, N, S)$, где:

$V_T = \{1, 0, a, b, \epsilon\}, V_N = \{A, B, S\}, P = \{S \rightarrow AB, A \rightarrow SA \mid BB \mid bB, B \rightarrow b,$

$P \rightarrow aA$;

$V_T \rightarrow a b c d, , , \emptyset, V_N \rightarrow A, B, S, P \rightarrow SS \mid Aa b \mid B, A \rightarrow c d \mid A \mid a, B \rightarrow c d \mid B$;

$V_T \rightarrow 1, 0, V_N \rightarrow A, S, P \rightarrow SS \mid 1A0, A \rightarrow 1A0$;

$V_T \rightarrow a b c d, , , \emptyset, V_N \rightarrow A, B, C, S, P \rightarrow SS \mid aC \mid bA, A \rightarrow cAB, B \rightarrow aC, C \rightarrow bA \mid d$.

Варианты заданий к лабораторной работе № 3

1. Разработать и реализовать программу, которая может выделить из текста программы, написанной на языке Java 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 Java. 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 Java. 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 Java структуры записей и составить на них таблицу символов.
8. Разработать и реализовать программу, которая может выделить из текста на языке Java 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.

Описание технологии проведения

Во время проверки выполненных студентами лабораторных работ проводится устный опрос с целью оценки понимания усвоенного материала.

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 на основе оценки выполнения практических задач.

20.2 Промежуточная аттестация Промежуточная аттестация по дисциплине осуществляется по итогам лабораторных работ. Перечень заданий приведен выше.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования. При оценивании используются качественные шкалы оценок.