

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Агафонов Александр Викторович

Должность: Директор филиала

Дата подписания: 18.06.2023 09:54:58

Университет: Московский политех

2539477a8ecf706dc9cff164bc411eb6d3c4ab06

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ЧЕБОКСАРСКИЙ ИНСТИТУТ (ФИЛИАЛ) МОСКОВСКОГО ПОЛИТЕХНИЧЕСКОГО УНИВЕРСИТЕТА

Кафедра Информационных технологий и систем управления



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Теория вычислительных процессов и языков программирования»

(наименование дисциплины)

Направление подготовки	09.03.02 «Информационные системы и технологии» (код и наименование направления подготовки)
Направленность (профиль) подготовки	«Информационные технологии в медиаиндустрии и дизайне» (наименование профиля подготовки)
Квалификация выпускника	бакалавр
Форма обучения	очная, заочная
Год начала обучения	2026

Чебоксары, 2026

Рабочая программа дисциплины разработана в соответствии с:

- федеральным государственным образовательным стандартом высшего образования - бакалавриат по направлению подготовки 09.03.02 – Информационные системы и технологии, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 926 от 19 сентября 2017 г. зарегистрированный в Минюсте 12 октября 2017 года, рег. номер 48535 (далее – ФГОС ВО);

- учебным планом (очной, заочной форм обучения) по направлению подготовки 09.03.02 «Информационные системы и технологии».

Рабочая программа дисциплины включает в себя оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине (п.6 Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины).

Автор Пикина Наталия Евгеньевна, кандидат педагогических наук, доцент кафедры информационных технологий и систем управления

(указать ФИО, ученую степень, ученое звание или должность)

Программа одобрена на заседании кафедры Информационных технологий и систем управления (протокол № 9 от 22.05.2026 г.).

1. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения образовательной программы (Цели освоения дисциплины)

1.1. *Целями* освоения дисциплины «Теория вычислительных процессов и языков программирования» являются:

освоение теоретических основ формальных языков и трансляций, современных подходов распознавания и транслирования языков, концепций автоматного программирования, изучение основных положений, связанных с понятием и взаимодействием вычислительных процессов, а также некоторых структурных решений построения вычислительных машин и систем;

Для достижения целей дисциплины необходимо решить следующую *основную задачу* – сформировать у обучающихся теоретические знания и практические навыки, необходимые для:

изучения классической теории вычислительных процессов, а также приобретения практических навыков по разработке программ.

изучения структурных решений ЭВМ, методов и алгоритмов управления процессами, и ресурсами, принципов организации вычислительных процессов в ОС;

систематического рассмотрения основных понятий языков программирования, синтаксиса, семантики, формальных способов описания языков программирования, типов данных, способов и механизмов управления данными, методов и основных этапов трансляции, конструкции распределенного и параллельного программирования.

1.2. Области профессиональной деятельности и (или) сферы профессиональной деятельности, в которых выпускники, освоившие программу, могут осуществлять профессиональную деятельность:

06 Связь, информационные и коммуникационные технологии (в сфере исследования, разработки, внедрения и сопровождения информационных технологий и систем).

1.3. К основным задачам изучения дисциплины относится подготовка обучающихся к выполнению трудовых функций в соответствии с профессиональными стандартами:

Наименование профессиональных стандартов (ПС)	Код, наименование и уровень квалификации ОТФ, на которые ориентирована дисциплина	Код и наименование трудовых функций, на которые ориентирована дисциплина
06.015 Профессиональный стандарт «Специалист по информационным системам»,	С Выполнение работ и управление работами по созданию (модификации) и	С/16.6 Проектирование и дизайн ИС С/14.6 Разработка архитектуры ИС С/15.6

Наименование профессиональных стандартов (ПС)	Код, наименование и уровень квалификации ОТФ, на которые ориентирована дисциплина	Код и наименование трудовых функций, на которые ориентирована дисциплина
утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2014 г. № 896н (зарегистрирован Министерством юстиции Российской Федерации 24 декабря 2014 г., регистрационный № 35361) (с изменениями на 12.12.2016, регистрационный номер 153)	сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	Разработка прототипов ИС С/16.6 Проектирование и дизайн ИС С/18.6 Организационное и технологическое обеспечение кодирования на языках программирования
06.025 Профессиональный стандарт «Специалист по дизайну графических пользовательских интерфейсов», утв. приказом Министерством труда и социальной защиты РФ 29 сентября 2020 № 671н	D Эвристическая оценка графического пользовательского интерфейса	D/01.6 Формальная оценка графического пользовательского интерфейса D/02.6 Анализ данных о действиях пользователей при работе с интерфейсом

1.4. Компетенции обучающегося, формируемые в результате освоения дисциплины

Наименование категории (группы) компетенций	Код и наименование компетенций	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения
Выполнение работ и управление работами по созданию (модификации) и сопровождению ИС, автоматизирующих задачи организационного управления и бизнес-процессы	ПК-6 Способен организовывать и обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования	<i>на уровне знаний:</i> знать основные результаты теории формальных языков; <i>на уровне умений:</i> уметь создавать распознаватели, интерпретаторы и трансляторы информационных потоков; уметь находить и

			<p>устранять проблем взаимодействия вычислительных процессов; <i>на уровне навыков:</i> навыками программного построения распознавателей, интерпретаторов и трансляторов информационных потоков</p>
		<p>ПК-6.2. Уметь: распределять работы и выделять ресурсы</p>	<p><i>на уровне знаний:</i> знать формальные модели основных вычислительных процессов, методы анализа вычислительных процессов; <i>на уровне умений:</i> уметь моделировать работу алгоритмов взаимодействия процессов и ресурсов в работе вычислительной системы. <i>на уровне навыков:</i> навыками использования стандартных инструментальных средств построения распознавателей и трансляторов в системном программировании</p>

		ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	<p><i>на уровне знаний:</i> теоретические основы методов проектирования и способы описания языков программирования, стандарты, используемые для языков программирования</p> <p><i>на уровне умений:</i> уметь разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик</p> <p><i>на уровне навыков:</i> навыками практического анализа проблем взаимодействия вычислительных процессов.</p>
--	--	--	--

2. Место дисциплины в структуре ОПОП

Дисциплина Б1.Д(М).В.10 «Теория вычислительных процессов и языков программирования» реализуется в рамках части, формируемой участниками образовательных отношений Блока 1 «Элективные дисциплины (модули)» программы бакалавриата.

Дисциплина «Теория вычислительных процессов и языков программирования» преподается обучающимся по очной форме обучения в 6-м и 7-м семестрах, по заочной форме обучения в 7-м и 8-м семестрах.

Дисциплина «Теория вычислительных процессов и языков программирования» является промежуточным этапом формирования компетенций ПК-6 в процессе освоения ОПОП.

Дисциплина «Теория вычислительных процессов и языков программирования» основывается на знаниях, умениях и навыках, приобретенных в ходе изучения дисциплин Разработка игровых приложений, Математические методы обработки изображений, Математические основы теории систем, учебная практика: технологическая (проектно-технологическая) практика и является предшествующей для изучения дисциплин производственная практика: технологическая (проектно-технологическая) практика, Цифровая обработка аудио и видеoinформации, производственная практика: преддипломная практика, Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная

итоговая аттестация: выполнение и защита выпускной квалификационной работы..

Формой промежуточной аттестации знаний обучаемых по очной форме обучения является зачет в 6-м семестре и экзамен в 7-м семестре, по заочной форме обучения зачет в 7-м семестре и экзамен в 8-м семестре.

3. Объем дисциплины

очная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 6 в часах	Семестр 7 в часах
Общая трудоёмкость дисциплины	6 з.е. - 216 ак.час	108 ак.час	108 ак.час
Контактная работа - Аудиторные занятия	103	54	49
<i>Лекции</i>	34	18	16
<i>Лабораторные занятия</i>	34	18	16
<i>Семинары, практические занятия</i>	34	18	16
<i>Консультация</i>	1	-	1
Самостоятельная работа	77	54	23
Курсовая работа (курсовой проект)	-	-	-
Вид промежуточной аттестации	Экзамен – 36 часов	Зачет	Экзамен – 36 часов

заочная форма обучения:

Вид учебной работы по дисциплине	Всего в з.е. и часах	Семестр 7 в часах	Семестр 8 в часах
Общая трудоёмкость дисциплины	6 з.е. - 216 ак.час	108 ак.час	108 ак.час
Контактная работа - Аудиторные занятия	25	12	13
<i>Лекции</i>	8	4	4
<i>Лабораторные занятия</i>	8	4	4
<i>Семинары, практические занятия</i>	8	4	4
<i>Консультация</i>	1	-	1
Самостоятельная работа	178	92	86
Курсовая работа (курсовой проект)	-	-	-
Вид промежуточной аттестации	Экзамен – 13 часов	Зачет – 4 часа	Экзамен – 9 часов

4. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

4.1. Учебно-тематический план

Очная форма обучения

Тема (раздел)	Количество часов	Код
---------------	------------------	-----

	контактная работа			самостоятельная работа	индикатора достижений компетенции
	лекции	лабораторные занятия	семинары и практические занятия		
Тема 1. История языков высокого уровня.	2	2	2		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 2. Представление о назначении и структуре языка Бэкуса-Наура.	2	2	2		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 3. Формальные языки и грамматики	2	2	2		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 4. Классификация по Хомскому.	4	4	4		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 5. Преобразования грамматик	4	4	4		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 6. Дерево разбора (дерево вывода)	4	4	4		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 7. Ограниченность контекстно-свободной грамматики	4	4	4		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 8. Конечные автоматы	4	4	4		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 9. Минимизация конечных автоматов	4	4	4		ПК-6.1, ПК-6.2, ПК-6.3,
Тема 10. Построение лексического анализатора	4	4	4		ПК-6.1, ПК-6.2, ПК-6.3,
Консультация	1			-	ПК-6.1, ПК-6.2, ПК-6.3,
Курсовая работа (курсовой проект)				-	ПК-6.1, ПК-6.2, ПК-6.3,
Контроль (экзамен)	36				ПК-6.1, ПК-6.2, ПК-6.3,
ИТОГО	103			77	

Заочная форма обучения

Тема (раздел)	Распределение часов			Самостоятельная работа	Формируемые компетенции (код)
	Лекции	Лабораторные занятия	Практические занятия		
Тема 1. История языков высокого уровня.	2	-	-	16	ПК-6.1, ПК-6.2,

					ПК-6.3,
Тема 2. Представление о назначении и структуре языка Бэкуса-Наура.	2	-	-	18	ПК-6.1, ПК-6.2, ПК-6.3,
Тема 3. Формальные языки и грамматики	-	2	2	18	ПК-6.1, ПК-6.2, ПК-6.3,
Тема 4. Классификация по Хомскому.	2	-	-	18	ПК-6.1, ПК-6.2, ПК-6.3,
Тема 5. Преобразования грамматик	-	2	-	18	ПК-6.1, ПК-6.2, ПК-6.3,
Тема 6. Дерево разбора (дерево вывода)	-	2	2	18	ПК-6.1, ПК-6.2, ПК-6.3,
Тема 7. Ограниченность контекстно-свободной грамматики	2	-	-	18	ПК-6.1, ПК-6.2, ПК-6.3,
Тема 8. Конечные автоматы	-	-	2	18	ПК-6.1, ПК-6.2, ПК-6.3,
Тема 9. Минимизация конечных автоматов	-	2	-	18	ПК-6.1, ПК-6.2, ПК-6.3,
Тема 10. Построение лексического анализатора	-	-	2	18	ПК-6.1, ПК-6.2, ПК-6.3,
Консультация	1			-	ПК-6.1, ПК-6.2, ПК-6.3,
Курсовая работа (курсовой проект)				-	ПК-6.1, ПК-6.2, ПК-6.3,
Контроль (экзамен)				13	ПК-6.1, ПК-6.2, ПК-6.3,
ИТОГО	25			178	

4.2. Содержание дисциплины

Тема 1. История языков высокого уровня.

Эволюция языков программирования: от машинного к языкам высокого уровня.

Основные этапы развития и классификация языков.

Роль языков высокого уровня в развитии вычислительной техники и программного обеспечения.

Влияние исторического контекста на формирование языков.

Тема 2. Представление о назначении и структуре языка Бэкуса-Наура.

Введение в форму записи синтаксиса языка — нотация Бэкуса-Наура (BNF).

Назначение BNF для описания формальных грамматик.

Основные элементы BNF: терминалы, нетерминалы, правила продукции.

Примеры описания синтаксиса простых языков с помощью BNF.

Тема 3. Формальные языки и грамматики

Понятие формального языка. Алфавиты, слова и множества слов.

Типы грамматик и их роль в формальном описании языков.

Понятие продукции, правил и языка, задаваемого грамматикой.

Связь между грамматиками и автоматами.

Тема 4. Классификация по Хомскому.

Четырёхуровневая иерархия грамматик: тип 0, тип 1, тип 2, тип 3.

Описание каждого типа грамматик и соответствующих классов языков.

Связь грамматик с автоматами: Тьюринг-машины, линейно-ограниченные автоматы, магазинные автоматы, конечные автоматы.

Практическое значение классификации для теории компиляторов.

Тема 5. Преобразования грамматик

Необходимость преобразования грамматик для упрощения и анализа.

Удаление леворекурсии и левого фактора.

Приведение грамматики к нормальной форме Хомского и нормальной форме Грейбера–Нэйр.

Примеры преобразований и их применение в парсерах.

Тема 6. Дерево разбора (дерево вывода)

Понятие дерева разбора как представления синтаксической структуры.

Виды деревьев разбора: левая и правая рекурсия.

Построение дерева разбора для входной строки.

Значение дерева разбора для компиляции и интерпретации программ.

Тема 7. Ограниченность контекстно-свободной грамматики

Пределы выразительной способности контекстно-свободных грамматик.

Языки, не описываемые контекстно-свободными грамматиками.

Примеры сложных языков и их характеристики.

Значение ограничений при разработке языков программирования.

Тема 8. Конечные автоматы

Определение и виды конечных автоматов: детерминированные и недетерминированные.

Формальное описание автоматов: состояния, алфавит, переходы, начальное и конечные состояния.

Использование конечных автоматов для распознавания регулярных языков.

Примеры построения и работы автоматов.

Тема 9. Минимизация конечных автоматов

Цели минимизации: уменьшение числа состояний без изменения распознаваемого языка.

Алгоритмы минимизации детерминированных конечных автоматов.

Примеры процесса минимизации.

Применение минимизации в оптимизации лексических анализаторов.

Тема 10. Построение лексического анализатора

Роль лексического анализа в процессе компиляции.

Принципы построения лексических анализаторов на основе конечных автоматов.

Использование регулярных выражений для описания лексем.

Автоматическая генерация лексических анализаторов (lex, flex).

Обработка ошибок и исключительных ситуаций на этапе лексического анализа.

5. Учебно-методическое обеспечение самостоятельной работы студентов.

Самостоятельная работа проводится с целью: выявления оптимальных конструктивных решений и параметров, определения наиболее эффективных режимов эксплуатации, стратегии текущего технического обслуживания и ремонтов; углубления и расширения теоретических знаний студентов; формирования умений использовать нормативную, справочную документацию, учебную и специальную литературу; развития познавательных способностей и активности обучающихся: самостоятельности, ответственности, организованности; формирования профессиональных компетенций; развитию исследовательских умений студентов.

Формы и виды самостоятельной работы студентов: чтение основной и дополнительной литературы – самостоятельное изучение материала по рекомендуемым литературным источникам; работа с библиотечным каталогом, самостоятельный подбор необходимой литературы; работа со справочником; поиск необходимой информации в сети Интернет; конспектирование источников; подготовка к различным формам текущей и промежуточной аттестации; выполнение домашних контрольных работ; самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, задачи, тесты).

Технология организации самостоятельной работы обучающихся включает использование информационных и материально-технических ресурсов образовательного учреждения: библиотеку с читальным залом, компьютерные классы с возможностью работы в Интернет; аудитории (классы) для консультационной деятельности.

Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель проводит консультирование по выполнению задания, который включает цель задания, его содержания, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. Во время выполнения обучающимися внеаудиторной самостоятельной работы и при необходимости преподаватель может проводить индивидуальные и групповые консультации.

Контроль самостоятельной работы студентов предусматривает: соотнесение содержания контроля с целями обучения; объективность контроля; валидность контроля (соответствие предъявляемых заданий тому, что

предполагается проверить); дифференциацию контрольно-измерительных материалов.

Формы контроля самостоятельной работы: просмотр и проверка выполнения самостоятельной работы преподавателем; организация самопроверки, взаимопроверки выполненного задания в группе; обсуждение результатов выполненной работы на занятии; проведение устного опроса.

Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Тема 1. История языков высокого уровня.	<ol style="list-style-type: none"> 1. Развитие языков программирования: от машинного к высокоуровневым языкам. 2. Появление первого компилятора: история и значение. 3. Основные этапы развития языков программирования: FORTRAN, COBOL, C, C++ и другие. 4. Влияние теории вычислений на создание языков высокого уровня. 5. Сравнение процедурных, объектно-ориентированных и функциональных языков. 6. Современные языки программирования: Python, Java, JavaScript и их особенности. 7. Влияние стандартов на развитие языков программирования. 8. Паттерны и парадигмы программирования: как они развивались. 9. Будущее языков программирования: тенденции и новые направления. 10. Влияние открытого программного обеспечения на развитие языков. 	<p>Изучение литературы по истории языков программирования. Проведение сравнительного анализа нескольких языков программирования. Просмотр и конспектирование видеоматериалов о первом компиляторе.</p>
Тема 2. Представление о назначении и структуре языка Бэкуса-Наура.	<ol style="list-style-type: none"> 1. Определение языка Бэкуса-Наура (BNF) и его назначение. 2. Синтаксическая структура языка BNF: грамматика и правила. 3. Примеры использования BNF для описания языков программирования. 4. Сравнение BNF с другими формальными языками. 5. Применение BNF в компиляторах и интерпретаторах. 6. Ограничения BNF и альтернативные подходы: ABNF и EBNF. 7. Роль BNF в стандартизации языков программирования. 8. Примеры грамматик, написанных на BNF. 9. Влияние BNF на разработку новых языков программирования. 10. Будущее языка Бэкуса-Наура: новые подходы и технологии. 	<p>Исследование примеров грамматик, написанных на BNF. Проведение анализа применения BNF в современных языках программирования. Просмотр материалов о языке Бэкуса-Наура.</p>
Тема 3. Формальные языки	<ol style="list-style-type: none"> 1. Определение формальных языков и их роль в теории вычислений. 	<p>Выполнение практических задач на</p>

и грамматики	<ol style="list-style-type: none"> 2. Классификация формальных языков: регулярные, контекстно-свободные, контекстно-зависимые. 3. Примеры формальных языков и их грамматик. 4. Роль грамматик в синтаксическом анализе программ. 5. Применение формальных языков в компиляторах и интерпретаторах. 6. Связь формальных языков с автоматами. 7. Конструкции грамматик: правила, нетерминалы и терминалы. 8. Применение формальных языков в области искусственного интеллекта. 9. Будущее формальных языков: новые подходы и технологии. 10. Разработка и анализ формальных языков для специфических задач. 	создание простых грамматик. Исследование примеров применения формальных языков в программировании. Просмотр материалов о синтаксическом анализе.
Тема 4. Классификация по Хомскому.	<ol style="list-style-type: none"> 1. Определение классификации формальных языков по Хомскому. 2. Уровни классификации: регулярные, контекстно-свободные, контекстно-зависимые и рекурсивно перечислимые языки. 3. Примеры языков для каждой категории классификации. 4. Связь между классами языков и соответствующими автоматами. 5. Применение классификации Хомского в компиляторах и теории вычислений. 6. Ограничения различных классов языков. 7. Влияние классификации Хомского на разработку языков программирования. 8. Примеры задач на распознавание языков из разных классов. 9. Будущее теории формальных языков: новые подходы и исследования. 10. Сравнение классификации Хомского с другими классификациями формальных языков. 	Исследование примеров языков для каждой категории классификации. Выполнение практических задач на распознавание языков. Просмотр материалов о классовой классификации языков.
Тема 5. Преобразования грамматик	<ol style="list-style-type: none"> 1. Определение преобразований грамматик и их роль в теории формальных языков. 2. Основные виды преобразований: упрощение, нормализация, эквивалентность. 3. Преобразование контекстно-свободных грамматик в нормальную форму Хомского. 4. Преобразование регулярных грамматик в конечные автоматы и наоборот. 5. Применение преобразований грамматик в компиляторах. 6. Грамматические преобразования и их влияние на синтаксический анализ. 7. Примеры преобразований грамматик в реальных языках программирования. 8. Автоматизированные инструменты для преобразования грамматик. 9. Будущее преобразований грамматик: новые подходы и 	Выполнение практических задач на преобразование грамматик. Исследование примеров использования преобразований в компиляторах. Просмотр и обсуждение материалов о преобразованиях грамматик.

	технологии. 10. Сравнение различных методов преобразования грамматик.	
Тема 6. Дерево разбора (дерево вывода)	<ol style="list-style-type: none"> 1. Определение дерева разбора и его назначение. 2. Структура дерева разбора: узлы, листья и их значения. 3. Процесс построения дерева разбора для выражений и предложений. 4. Связь между деревом разбора и грамматикой. 5. Примеры построения деревьев разбора для различных языков. 6. Применение деревьев разбора в синтаксическом анализе. 7. Преимущества и недостатки использования деревьев разбора. 8. Алгоритмы для работы с деревьями разбора. 9. Будущее деревьев разбора: новые подходы и технологии. 10. Сравнение деревьев разбора с другими представлениями синтаксиса. 	Выполнение практических задач на построение деревьев разбора. Исследование примеров применения деревьев разбора в языках программирования. Просмотр материалов о синтаксическом анализе с использованием деревьев разбора.
Тема 7. Ограниченность контекстно-свободной грамматики	<ol style="list-style-type: none"> 1. Определение контекстно-свободной грамматики и её назначение. 2. Ограничения контекстно-свободных грамматик: примеры нерешаемых проблем. 3. Сравнение контекстно-свободных грамматик с другими классами грамматик. 4. Примеры языков, которые не могут быть описаны контекстно-свободными грамматиками. 5. Применение ограничений контекстно-свободных грамматик в практике. 6. Теорема Чомского о ограниченности контекстно-свободных грамматик. 7. Будущее исследований в области контекстно-свободных грамматик: новые подходы и технологии. 8. Ограничения контекстно-свободных грамматик в компиляторах. 9. Примеры задач на определение контекстно-свободных языков. 10. Сравнение различных классов грамматик и их возможностей. 	Выполнение практических задач на определение контекстно-свободных языков. Исследование примеров языков, которые не могут быть описаны контекстно-свободными грамматиками. Просмотр материалов о теореме Чомского.
Тема 8. Конечные автоматы	<ol style="list-style-type: none"> 1. Определение конечных автоматов и их назначение. 2. Типы конечных автоматов: детерминированные и недетерминированные. 3. Применение конечных автоматов в теории формальных языков. 4. Примеры построения конечных автоматов для различных языков. 5. Связь между конечными автоматами и регулярными выражениями. 6. Преобразование регулярных выражений в конечные автоматы. 7. Реализация конечных автоматов в программировании. 8. Ограничения конечных автоматов и их применение. 	Выполнение практических задач на построение конечных автоматов. Исследование примеров применения конечных автоматов в языках программирования. Просмотр материалов о конечных автоматах.

	<p>9. Будущее исследований в области конечных автоматов: новые подходы и технологии.</p> <p>10. Сравнение конечных автоматов с другими моделями вычислений.</p>	
<p>Тема 9. Минимизация конечных автоматов</p>	<p>1. Определение минимизации конечных автоматов и её значение.</p> <p>2. Алгоритмы минимизации конечных автоматов: алгоритм Мура и алгоритм Хопкрофта.</p> <p>3. Примеры применения минимизации конечных автоматов в практике.</p> <p>4. Понятие эквивалентности автоматов и её роль в минимизации.</p> <p>5. Связь минимизации конечных автоматов с оптимизацией.</p> <p>6. Практическое применение минимизированных автоматов в программировании.</p> <p>7. Ограничения и сложности, связанные с минимизацией автоматов.</p> <p>8. Будущее минимизации конечных автоматов: новые подходы и технологии.</p> <p>9. Сравнение различных методов минимизации конечных автоматов.</p> <p>10. Применение минимизации в компиляторах и анализаторах.</p>	<p>Выполнение практических задач на минимизацию конечных автоматов. Исследование примеров использования минимизации в программировании. Просмотр материалов об алгоритмах минимизации.</p>
<p>Тема 10. Построение лексического анализатора</p>	<p>1. Определение лексического анализатора и его роль в компиляторах.</p> <p>2. Процесс лексического анализа: этапы и методы.</p> <p>3. Использование регулярных выражений для описания лексических элементов.</p> <p>4. Создание лексического анализатора: инструменты и фреймворки.</p> <p>5. Примеры реализации лексических анализаторов на различных языках.</p> <p>6. Связь лексических анализаторов с синтаксическим анализом.</p> <p>7. Проблемы, возникающие при построении лексических анализаторов.</p> <p>8. Будущее лексического анализа: новые подходы и технологии.</p> <p>9. Сравнение различных методов и инструментов для построения лексических анализаторов.</p> <p>10. Практическое применение лексических анализаторов в программировании.</p>	<p>Выполнение практических задач на создание лексического анализатора. Исследование примеров реализации лексических анализаторов. Просмотр материалов о лексическом анализе и его значении.</p>

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему самостоятельной работы, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер
«Хорошо»	Обучающийся в целом раскрывает тему

	самостоятельной работы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера
«Удовлетворительно»	Обучающийся в целом раскрывает тему самостоятельной работы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой самостоятельной работы

6. Оценочные материалы (фонд оценочных средств) для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

6.1. Паспорт фонда оценочных средств

№	Контролируемые разделы (темы) дисциплины	Код и наименование компетенции	Индикатор достижения компетенции	Наименование оценочного средства
1.	Тема 1. История языков высокого уровня.	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	Опрос, тест, доклад, зачет, экзамен
2.	Тема 2. Представление о назначении и структуре языка Бэкуса-Наура.	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь:	Опрос, тест, доклад, зачет, экзамен

			распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	
3.	Тема 3. Формальные языки и грамматики	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	Опрос, тест, доклад, зачет, экзамен
4.	Тема 4. Классификация по Хомскому.	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	Опрос, тест, доклад, зачет, экзамен
5.	Тема 5. Преобразования грамматик	ПК-6 Способен организовывать и	ПК-6.1. Знать: инструменты и	Опрос, тест, доклад,

		технологически обеспечивать кодирование на языках программирования	методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	зачет, экзамен
6.	Тема 6. Дерево разбора (дерево вывода)	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	Опрос, тест, доклад, зачет, экзамен
7.	Тема 7. Ограниченность контекстно-свободной грамматики	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением	Опрос, тест, доклад, зачет, экзамен

			соответствия разработанного кода и процесса кодирования на языках программирования	
8.	Тема 8. Конечные автоматы	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	Опрос, тест, доклад, зачет, экзамен
9.	Тема 9. Минимизация конечных автоматов	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	Опрос, тест, доклад, зачет, экзамен
10.	Тема 10. Построение лексического анализатора	ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	ПК-6.1. Знать: инструменты и методы верификации структуры программного кода, регламенты кодирования на языках программирования ПК-6.2. Уметь: распределять работы	Опрос, тест, доклад, зачет, экзамен

			и выделять ресурсы ПК-6.3. Владеть: обеспечением соответствия разработанного кода и процесса кодирования на языках программирования	
--	--	--	---	--

Этапы формирования компетенций в процессе освоения ОПОП прямо связаны с местом дисциплин в образовательной программе. Каждый этап формирования компетенции, характеризуется определенными знаниями, умениями и навыками и (или) опытом профессиональной деятельности, которые оцениваются в процессе текущего контроля успеваемости, промежуточной аттестации по дисциплине (практике) и в процессе итоговой аттестации.

Дисциплина «Теория вычислительных процессов и языков программирования» является промежуточным этапом комплекса дисциплин, в ходе изучения которых у студентов формируются компетенция ПК-6.

Формирования компетенции ПК-6 начинается в ходе изучения дисциплин Разработка игровых приложений, Математические методы обработки изображений, Математические основы теории систем, учебная практика: технологическая (проектно-технологическая) практика.

Завершается работа по формированию у студентов указанных компетенций в ходе изучения дисциплин производственная практика: технологическая (проектно-технологическая) практика, Цифровая обработка аудио и видеoinформации, производственная практика: преддипломная практика.

Итоговая оценка сформированности компетенций ПК-6 определяется в период Государственная итоговая аттестация: подготовка к сдаче и сдача государственного экзамена, Государственная итоговая аттестация: выполнение и защита выпускной квалификационной работы.

В процессе изучения дисциплины, компетенции также формируются поэтапно.

Основными этапами формирования ПК-6 при изучении дисциплины Б1.Д(М).В.10 «Теория вычислительных процессов и языков программирования» является последовательное изучение содержательно связанных между собой тем учебных занятий. Изучение каждой темы предполагает овладение студентами необходимыми дескрипторами (составляющими) компетенций. Для оценки уровня сформированности компетенций в процессе изучения дисциплины предусмотрено проведение текущего контроля успеваемости по темам (разделам) дисциплины и промежуточной аттестации по дисциплине – экзамен.

6.2. Контрольные задания и материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

6.2.1. Контрольные вопросы по темам (разделам) для опроса на занятиях

Тема (раздел)	Вопросы
Тема 1. История языков высокого уровня.	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Этапы развития языков программирования высокого уровня. 2. Преимущества и недостатки первых языков высокого уровня. 3. Основные языки программирования первого поколения (Fortran, Lisp, Algol). 4. Роль компиляции и интерпретации в развитии языков программирования. 5. Как возникали и развивались идеологии структурного программирования и объектно-ориентированного подхода. 6. Развитие языков с автоматическим управлением памятью (например, язык Java). 7. Переход от процедурного к объектно-ориентированному программированию. 8. Основные идеи, лежащие в основе функциональных языков программирования. 9. Современные тенденции в развитии языков программирования высокого уровня. 10. Влияние истории языков программирования на современные разработки.
Тема 2. Представление о назначении и структуре языка Бэкуса-Наура.	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Что такое синтаксическая структура языка Бэкуса-Наура? 2. Основные компоненты грамматики Бэкуса-Наура. 3. Принципы записи правил грамматики в форме Бэкуса-Наура. 4. Роль языка Бэкуса-Наура в теории компиляторов. 5. Примеры использования грамматик Бэкуса-Наура для описания синтаксиса языков. 6. Отличие грамматик Бэкуса-Наура от других методов описания синтаксиса. 7. Применение грамматик Бэкуса-Наура в современных компиляторах. 8. Преимущества и недостатки использования Бэкуса-Наура в теории языков. 9. Влияние грамматик Бэкуса-Наура на дальнейшее развитие формальных грамматик. 10. Реализация компилятора с использованием грамматик Бэкуса-Наура.
Тема 3. Формальные языки и грамматики	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Определение формальных языков и их роль в теории вычислений. 2. Различие между регулярными, контекстно-свободными и контекстно-зависимыми языками. 3. Применение формальных языков в теории алгоритмов. 4. Преимущества и недостатки использования формальных грамматик для описания синтаксиса. 5. Пример контекстно-свободной грамматики. 6. Роль формальных языков в определении синтаксической и семантической правильности программ. 7. Влияние формальных грамматик на создание языков

	<p>программирования.</p> <p>8. Роль формальных языков в анализе и синтезе программ.</p> <p>9. Модели вычислений, основанные на формальных языках.</p> <p>10. Применение формальных грамматик в обработке естественных языков.</p>
Тема 4. Классификация по Хомскому.	<p>ПК-6</p> <p>1. Классы языков по Хомскому: регулярные, контекстно-свободные, контекстно-зависимые, рекурсивно перечислимые.</p> <p>2. Принципы классификации формальных языков по Хомскому.</p> <p>3. Различие между регулярными и контекстно-свободными языками.</p> <p>4. Примеры грамматик для каждого из классов по Хомскому.</p> <p>5. Связь между автоматами и классификацией формальных языков по Хомскому.</p> <p>6. Преимущества использования классификации по Хомскому для анализа языков программирования.</p> <p>7. Трудности, связанные с преобразованием грамматик в другие классы по Хомскому.</p> <p>8. Как классификация по Хомскому помогает в разработке компиляторов.</p> <p>9. Различие между регулярными выражениями и контекстно-свободными грамматиками.</p> <p>10. Применение классификации по Хомскому в разработке лексических и синтаксических анализаторов.</p>
Тема 5. Преобразования грамматик	<p>ПК-6</p> <p>1. Что такое преобразования грамматик и зачем они нужны?</p> <p>2. Преобразование контекстно-свободной грамматики в нормальную форму Хомского.</p> <p>3. Преобразование грамматик в регулярные выражения.</p> <p>4. Преобразования грамматик в автоматные модели.</p> <p>5. Как преобразования грамматик влияют на эффективный анализ и компиляцию.</p> <p>6. Основные методы преобразования грамматик: нормализация, минимизация.</p> <p>7. Практическое применение преобразований грамматик в разработке компиляторов.</p> <p>8. Преобразования для устранения леворекурсивности.</p> <p>9. Автоматические инструменты для преобразования грамматик.</p> <p>10. Преимущества и недостатки различных методов преобразования грамматик.</p>
Тема 6. Дерево разбора (дерево вывода)	<p>ПК-6</p> <p>1. Что такое дерево разбора (дерево вывода) и его роль в синтаксическом анализе.</p> <p>2. Структура дерева разбора и его элементы.</p> <p>3. Преимущества и недостатки использования дерева разбора для представления синтаксиса программ.</p> <p>4. Как происходит построение дерева разбора на основе грамматик.</p> <p>5. Различие между деревом разбора и деревом синтаксического анализа.</p> <p>6. Алгоритмы построения дерева разбора.</p> <p>7. Как дерево разбора используется в компиляторах и интерпретаторах.</p>

	<p>8. Связь между деревом разбора и машинными кодами.</p> <p>9. Применение дерева разбора для оптимизации кода.</p> <p>10. Использование дерева разбора в обработке и трансформации программ.</p>
Тема 7. Ограниченность контекстно-свободной грамматики	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Что такое контекстно-свободная грамматика и её ограничения. 2. Примеры синтаксических конструкций, которые нельзя описать с помощью контекстно-свободных грамматик. 3. Преимущества и ограничения контекстно-свободных грамматик в теории языков программирования. 4. Как контекстно-свободная грамматика используется в компиляторах. 5. Проблемы, возникающие при обработке контекстно-свободных грамматик в реальных языках программирования. 6. Преимущества и недостатки контекстно-свободных грамматик по сравнению с другими типами грамматик. 7. Контекстно-свободные грамматики и их роль в синтаксическом анализе. 8. Проблемы с выражением синтаксической зависимости между компонентами языка с помощью контекстно-свободной грамматики. 9. Ограничения контекстно-свободных грамматик в теории обработки естественных языков. 10. Методы расширения контекстно-свободных грамматик для более сложных конструкций.
Тема 8. Конечные автоматы	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Определение конечного автомата и его компоненты. 2. Различие между детерминированным и недетерминированным конечным автоматом. 3. Примеры конечных автоматов для распознавания регулярных языков. 4. Применение конечных автоматов в лексическом анализе. 5. Как конечные автоматы помогают в разработке алгоритмов для поиска и распознавания паттернов. 6. Построение конечных автоматов для распознавания простых языков. 7. Роль конечных автоматов в реализации регулярных выражений. 8. Преимущества использования конечных автоматов в компиляторах. 9. Основные ограничения конечных автоматов в теории вычислений. 10. Использование конечных автоматов для синтаксического анализа.
Тема 9. Минимизация конечных автоматов	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Зачем и как минимизируют конечные автоматы. 2. Алгоритмы минимизации конечных автоматов. 3. Преимущества минимизации для повышения эффективности распознавания языков. 4. Различие между минимизированным и неминимизированным конечным автоматом. 5. Применение минимизации конечных автоматов в компиляторах и интерпретаторах.

	<p>6. Как минимизация автоматов помогает в оптимизации работы регулярных выражений.</p> <p>7. Алгоритм минимизации для детерминированных конечных автоматов.</p> <p>8. Алгоритм минимизации для недетерминированных конечных автоматов.</p> <p>9. Сложности минимизации конечных автоматов для сложных языков.</p> <p>10. Применение минимизации конечных автоматов в задачах обработки естественных языков.</p>
Тема 10. Построение лексического анализатора	<p>ПК-6</p> <p>1. Задачи лексического анализатора и его роль в компиляторе.</p> <p>2. Принципы работы лексического анализатора.</p> <p>3. Как строится лексический анализатор на основе конечных автоматов.</p> <p>4. Использование регулярных выражений для построения лексического анализатора.</p> <p>5. Преимущества и недостатки различных методов лексического анализа.</p> <p>6. Алгоритм работы лексического анализатора: от сканирования до классификации лексем.</p> <p>7. Применение лексического анализатора в интерпретаторах и компиляторах.</p> <p>8. Как лексический анализатор обрабатывает ошибочные лексемы.</p> <p>9. Оптимизация лексического анализатора для повышения скорости работы.</p> <p>10. Различие между лексическим и синтаксическим анализом.</p>

Шкала оценивания ответов на вопросы

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает ответ на каждый теоретический вопрос, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает теоретические вопросы, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает теоретические вопросы и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не знает ответов на поставленные теоретические вопросы.

6.2.2. Темы для докладов

Тема (раздел)	Вопросы
---------------	---------

<p>Тема 1. История языков высокого уровня.</p>	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Эволюция языков программирования: от машинных кодов к языкам высокого уровня. 2. Развитие языков программирования: от первого языка высокого уровня до современных языков. 3. Роль языка Фортран в развитии языков программирования. 4. Влияние языка С на современные языки программирования. 5. Языки программирования с акцентом на параллельные вычисления: от Fortran до OpenMP. 6. Язык Паскаль: влияние на обучение и стандарты программирования. 7. Как появились объектно-ориентированные языки: от Simula до Python. 8. Влияние современных языков программирования на разработку программного обеспечения.
<p>Тема 2. Представление о назначении и структуре языка Бэкуса-Наура.</p>	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Введение в язык Бэкуса-Наура: что это такое и как используется в теории языков. 2. Структура грамматики Бэкуса-Наура: терминальные и нетерминальные символы, правила продукции. 3. Как использовать форму Бэкуса-Наура для описания синтаксиса языков программирования. 4. Примеры грамматик, описанных с помощью Бэкуса-Наура. 5. Особенности представления рекурсивных грамматик в форме Бэкуса-Наура. 6. Использование грамматик Бэкуса-Наура в процессе разработки компиляторов. 7. Преимущества и недостатки использования Бэкуса-Наура для описания синтаксиса языков. 8. Применение формы Бэкуса-Наура в современных языках программирования.
<p>Тема 3. Формальные языки и грамматики</p>	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Введение в формальные языки и их классификация. 2. Формальные грамматики и их роль в теории языков. 3. Принципы построения грамматик для языков программирования. 4. Грамматики регулярных языков: синтаксис и примеры. 5. Контекстно-свободные грамматики: теория и практика. 6. Системы формальных грамматик для описания синтаксиса и семантики языков. 7. Применение формальных грамматик в анализе и синтаксическом анализе программ. 8. Теория формальных языков и их связь с компиляторами и интерпретаторами.
<p>Тема 4. Классификация по Хомскому.</p>	<p>ПК-6</p> <ol style="list-style-type: none"> 1. Введение в классификацию языков по Хомскому: основные категории. 2. Принципы классификации грамматик по Хомскому. 3. Классы Хомского: регулярные, контекстно-свободные, контекстно-зависимые и тип-0 грамматики. 4. Примеры языков для каждого класса классификации Хомского. 5. Свойства грамматик разных классов: выражаемость, сложности вычислений, применимость.

	<p>6. Алгоритмы для анализа грамматик по Хомскому.</p> <p>7. Пределы вычислений в контексте классификации Хомского.</p> <p>8. Практическое применение теории Хомского в создании компиляторов.</p>
<p>Тема 5. Преобразования грамматик</p>	<p>ПК-6</p> <p>1. Введение в преобразования грамматик: зачем это нужно и как применяется.</p> <p>2. Преобразования регулярных грамматик в конечные автоматы.</p> <p>3. Преобразование контекстно-свободных грамматик в нормальную форму Хомского.</p> <p>4. Как преобразования грамматик помогают в разработке компиляторов.</p> <p>5. Алгоритмы преобразования грамматик: примеры и применение.</p> <p>6. Преобразование грамматик для упрощения их анализа и обработки.</p> <p>7. Стандарты нормализации грамматик: нормальная форма и её применение.</p> <p>8. Преобразования и оптимизация грамматик в практике синтаксического анализа.</p>
<p>Тема 6. Дерево разбора (дерево вывода)</p>	<p>ПК-6</p> <p>1. Введение в концепцию дерева разбора и его роль в синтаксическом анализе.</p> <p>2. Структура и компоненты дерева разбора.</p> <p>3. Как строится дерево разбора для программы на языке высокого уровня.</p> <p>4. Применение дерева разбора в компиляторах и интерпретаторах.</p> <p>5. Рекурсивный и итеративный способы построения дерева разбора.</p> <p>6. Свойства дерева разбора и их анализ.</p> <p>7. Примеры построения дерева разбора для различных выражений.</p> <p>8. Алгоритмы синтаксического анализа с использованием деревьев разбора.</p>
<p>Тема 7. Ограниченность контекстно-свободной грамматики</p>	<p>ПК-6</p> <p>1. Введение в концепцию дерева разбора и его роль в синтаксическом анализе.</p> <p>2. Структура и компоненты дерева разбора.</p> <p>3. Как строится дерево разбора для программы на языке высокого уровня.</p> <p>4. Применение дерева разбора в компиляторах и интерпретаторах.</p> <p>5. Рекурсивный и итеративный способы построения дерева разбора.</p> <p>6. Свойства дерева разбора и их анализ.</p> <p>7. Примеры построения дерева разбора для различных выражений.</p> <p>8. Алгоритмы синтаксического анализа с использованием деревьев разбора.</p>
<p>Тема 8. Конечные автоматы</p>	<p>ПК-6</p> <p>1. Введение в конечные автоматы: что это такое и как они используются.</p> <p>2. Различие между детерминированными и недетерминированными конечными автоматами.</p> <p>3. Принципы работы конечных автоматов: состояние, переходы, входные символы.</p> <p>4. Применение конечных автоматов для распознавания регулярных языков.</p>

	<p>5. Как строятся и применяются конечные автоматы в анализе языков программирования.</p> <p>6. Реализация конечных автоматов в программировании.</p> <p>7. Примеры использования конечных автоматов для обработки текстов и строк.</p> <p>8. Минимизация конечных автоматов: задачи и решения.</p>
Тема 9. Минимизация конечных автоматов	<p>ПК-6</p> <p>1. Зачем нужна минимизация конечных автоматов и как это влияет на производительность.</p> <p>2. Принципы минимизации детерминированных конечных автоматов.</p> <p>3. Алгоритм минимизации конечных автоматов: описание и примеры.</p> <p>4. Преимущества минимизированных автоматов в контексте вычислительных систем.</p> <p>5. Как минимизация конечных автоматов используется в реализации компиляторов.</p> <p>6. Применение минимизации конечных автоматов в различных областях: теория языков, обработка данных.</p> <p>7. Алгоритмы минимизации автоматов: как выбрать подходящий в зависимости от задачи.</p> <p>8. Практическое применение минимизированных конечных автоматов в реальных системах.</p>
Тема 10. Построение лексического анализатора	<p>ПК-6</p> <p>1. Введение в лексический анализ и его роль в компиляторах.</p> <p>2. Принципы работы лексического анализатора: токены, регулярные выражения.</p> <p>3. Как лексический анализатор преобразует исходный код в последовательность токенов.</p> <p>4. Алгоритмы лексического анализа: поиск, распознавание, фильтрация.</p> <p>5. Реализация лексического анализатора на практике: примеры и библиотеки.</p> <p>6. Проблемы лексического анализа: сканирование и обработка больших объемов текста.</p> <p>7. Как лексический анализатор взаимодействует с синтаксическим анализатором.</p> <p>8. Оптимизация лексических анализаторов: методы и примеры.</p>

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	Обучающийся глубоко и содержательно раскрывает тему доклада, не допустив ошибок. Ответ носит развернутый и исчерпывающий характер.
«Хорошо»	Обучающийся в целом раскрывает тему доклада, однако ответ хотя бы на один из них не носит развернутого и исчерпывающего характера.
«Удовлетворительно»	Обучающийся в целом раскрывает тему доклада и допускает ряд неточностей, фрагментарно раскрывает содержание теоретических вопросов или их раскрывает содержательно, но допуская значительные неточности.
«Неудовлетворительно»	Обучающийся не владеет выбранной темой

6.2.3. Оценочные средства остаточных знаний (тест)

ПК-6.

1. Что такое вычислительный процесс?

- 1) Процесс преобразования данных в информацию
- 2) Процесс выполнения инструкций программного обеспечения на вычислительном устройстве
- 3) Процесс компиляции исходного кода
- 4) Процесс хранения данных в базе данных

2. Какой из языков программирования является процедурным?

- 1) Java
- 2) Python
- 3) Pascal
- 4) Lisp

3. Что такое компилятор?

- 1) Программа для выполнения исходного кода на машине
- 2) Программа для преобразования исходного кода в машинный код
- 3) Программа для взаимодействия с пользователем
- 4) Программа для хранения данных

4. Что такое интерпретатор?

- 1) Программа для преобразования исходного кода в машинный код
- 2) Программа для выполнения исходного кода построчно
- 3) Программа для оптимизации кода
- 4) Программа для управления памятью

5. Какой из этих языков программирования является объектно-ориентированным?

- 1) C
- 2) Python
- 3) Fortran
- 4) COBOL

6. Что такое грамматика в контексте теории языков программирования?

- 1) Правила, определяющие, как можно составлять строки на языке программирования
- 2) Совокупность функций, доступных в языке программирования
- 3) Набор инструкций для выполнения программ
- 4) Описание структур данных в языке программирования

7. Что такое алфавит в теории формальных языков?

- 1) Множество символов, используемых для создания строк
- 2) Совокупность всех возможных программ

- 3) Коллекция всех грамматик
- 4) Перечень инструкций языка программирования

8. Какая из следующих теорем относится к теории вычислений?

- 1) Теорема о неполноте Гёделя
- 2) Теорема о решении уравнений с использованием числовых методов
- 3) Теорема о максимальном потоке в графах
- 4) Теорема о сложности алгоритмов

9. Что такое рекурсивный язык?

- 1) Язык, который не имеет строгих синтаксических правил
- 2) Язык, где правила вывода могут использовать себя в качестве подзадач
- 3) Язык, который используется для создания графиков
- 4) Язык, не поддерживающий функции

10. Что такое тьюрингова машина?

- 1) Теоретическое устройство для вычислений, которое может моделировать любую вычислительную задачу
- 2) Машина для компиляции программ
- 3) Машина для выполнения циклических задач
- 4) Машина для создания графиков

11. Что такое конечный автомат?

- 1) Теоретическая модель вычислений, имеющая ограниченное количество состояний
- 2) Программа для выполнения вычислений в реальном времени
- 3) Модель вычислений, использующая бесконечную память
- 4) Модель, которая не может обрабатывать циклы

12. Какие из следующих языков являются языками высокого уровня?

- 1) Assembly
- 2) Fortran
- 3) COBOL
- 4) Java

13. Что такое контекстно-свободная грамматика?

- 1) Грамматика, где каждое правило состоит из строки, которая может быть заменена в произвольном месте текста
- 2) Грамматика, где каждое правило состоит из строки, которая может быть заменена только в начале строки
- 3) Грамматика, где правила не зависят от контекста
- 4) Грамматика, в которой правила не могут быть использованы для создания новых строк

14. Что такое регулярный язык?

- 1) Язык, который может быть описан с помощью регулярных выражений
- 2) Язык, который использует только фиксированное количество символов
- 3) Язык, который не поддерживает переменные
- 4) Язык, который имеет ограниченные возможности для вычислений

15. Какое из следующих утверждений верно для вычислительных процессов?

- 1) Каждый процесс имеет конечное количество состояний
- 2) Каждый процесс может работать бесконечно долго
- 3) Все процессы заканчиваются одинаково
- 4) Процесс всегда завершится после одного шага

16. Что такое трансляция в контексте языков программирования?

- 1) Преобразование одного языка в другой
- 2) Сохранение промежуточных данных
- 3) Процесс выполнения программы
- 4) Процесс компиляции исходного кода в машинный

17. Что такое предикат в логике программирования?

- 1) Множество данных, которые не изменяются
- 2) Функция, возвращающая значение истинности
- 3) Описание состояний вычислительного процесса
- 4) Совокупность правил для построения выражений

18. Что такое язык программирования с динамической типизацией?

- 1) Язык, в котором типы данных определяются на стадии компиляции
- 2) Язык, в котором типы данных определяются во время выполнения
- 3) Язык, который не поддерживает типизацию
- 4) Язык, в котором все переменные строго типизированы

19. Какая модель вычислений используется для описания алгоритмов, которые работают с потоками данных?

- 1) Модель Лямбда-выражений
- 2) Модель потока данных
- 3) Модель программирования с событиями
- 4) Модель Дейкстры

20. Что такое алгоритм?

- 1) Набор случайных операций, выполняемых в программе
- 2) Структура данных, которая хранит результаты работы программы
- 3) Последовательность шагов, решающих определенную задачу
- 4) Формула для вычисления математических значений

21. Что такое синтаксический анализ?

- 1) Процесс определения правильности исполнения программы

- 2) Процесс создания исходного кода программы
- 3) Процесс проверки структуры исходного кода по грамматике
- 4) Процесс выполнения программы

22. Что такое теория вычислительных процессов?

- 1) Теория, изучающая работу аппаратных средств компьютера
- 2) Теория, которая занимается разработкой языков программирования
- 3) Теория, изучающая способы решения задач с использованием вычислительных машин
- 4) Теория, описывающая методы разработки программных приложений

23. Что такое теория вычислительных сложностей?

- 1) Теория, изучающая количество шагов, необходимых для решения задачи
- 2) Теория, изучающая типы данных в языке программирования
- 3) Теория, анализирующая возможность использования разных операционных систем
- 4) Теория, которая исследует синтаксические структуры программ

24. Что такое язык с определенной семантикой?

- 1) Язык, в котором операторы имеют четкие и фиксированные значения
- 2) Язык, в котором невозможно задать выражения с неопределенным значением
- 3) Язык, где можно изменять правила интерпретации команд
- 4) Язык, не поддерживающий операторы

25. Что такое автоматное представление данных?

- 1) Представление данных в виде диаграмм состояний и переходов
- 2) Представление данных в виде таблиц
- 3) Представление данных в виде массивов
- 4) Представление данных в виде многозадачных операций

26. Что такое абстракция в программировании?

- 1) Скрытие сложных деталей реализации
- 2) Описание всех деталей работы программы
- 3) Реализация конкретных алгоритмов
- 4) Применение функций к данным

27. Какую задачу решает теорема о Тьюринговой машине?

- 1) Задачу синтаксического анализа
- 2) Задачу о вычислимости
- 3) Задачу о многозадачности
- 4) Задачу об эффективности алгоритмов

28. Что такое логика программирования?

- 1) Метод написания кода без использования выражений
- 2) Метод использования математических теорий для оптимизации кода
- 3) Система правил и принципов, используемая для описания алгоритмов
- 4) Метод автоматического тестирования программ

29. Что такое язык программирования с жесткой типизацией?

- 1) Язык, в котором переменные не требуют определения типов
- 2) Язык, в котором типы переменных определяются заранее и строго проверяются
- 3) Язык, не поддерживающий типизацию
- 4) Язык, который не требует объявления переменных

30. Что такое алгебра типов в языках программирования?

- 1) Теория, описывающая математические функции в программировании
- 2) Теория, изучающая правила работы с различными типами данных
- 3) Теория, описывающая правила компиляции кода
- 4) Теория, описывающая структуру данных

31. Что такое формальная грамматика?

- 1) Грамматика, определяющая правила написания алгоритмов
- 2) Грамматика, определяющая синтаксис языка программирования
- 3) Грамматика, ограничивающая использование переменных
- 4) Грамматика, описывающая поведение функций

32. Что такое эквивалентность языков программирования?

- 1) Принцип, при котором программы на разных языках дают одинаковые результаты
- 2) Принцип, при котором программы на разных языках имеют одинаковую синтаксическую структуру
- 3) Принцип, при котором можно изменить синтаксис языка
- 4) Принцип, при котором используются одинаковые алгоритмы для различных языков

33. Что такое инкрементальный компилятор?

- 1) Компилятор, который анализирует код построчно
- 2) Компилятор, который компилирует только измененные части программы
- 3) Компилятор, который выполняет программу после компиляции
- 4) Компилятор, который переводит программу на ассемблер

34. Что такое модель вычислений с памятью?

- 1) Модель, где вычисления выполняются только в процессе выполнения программы
- 2) Модель, где вычисления выполняются с использованием памяти машины

- 3) Модель, где нет необходимости в хранении данных
- 4) Модель, где данные вычисляются в реальном времени

35. Что такое домен языков программирования?

- 1) Область, для которой предназначен язык программирования
- 2) Пространство имен, в котором работает программа
- 3) Типы данных, используемые в языке программирования
- 4) Совокупность всех алгоритмов на языке программирования

36. Что такое нормальная форма грамматики?

- 1) Форма, в которой правила грамматики представляют собой простые цепочки символов
- 2) Форма, в которой грамматика описывает несколько операций
- 3) Форма, в которой грамматика не допускает произвольных преобразований
- 4) Форма, в которой грамматика не имеет символов

37. Что такое модель Лямбда-выражений?

- 1) Математическая модель вычислений на основе функций
- 2) Модель для работы с данными в реальном времени
- 3) Модель для многозадачности
- 4) Модель для создания алгоритмов на основе множества данных

38. Что такое трансцендентное вычисление?

- 1) Процесс вычислений, связанных с бесконечными рядами и функциями
- 2) Процесс, связанный с локальными вычислениями
- 3) Процесс использования переменных
- 4) Процесс описания математических алгоритмов

39. Что такое параллельные вычисления?

- 1) Вычисления, выполняющиеся с использованием нескольких процессоров одновременно
- 2) Вычисления, выполняющиеся на одном процессоре, но с множеством задач
- 3) Вычисления, выполняющиеся по очереди
- 4) Вычисления, связанные с математическими уравнениями

40. Что такое теория сложности алгоритмов?

- 1) Теория, изучающая, какие задачи могут быть решены на определенном языке программирования
- 2) Теория, изучающая, сколько времени или памяти необходимо для решения задачи
- 3) Теория, изучающая принципы работы с базами данных
- 4) Теория, описывающая, как преобразовывать данные

41. Что такое рекурсивное определение?

- 1) Определение, в котором описание включает само себя
- 2) Определение, в котором описание использует другие функции
- 3) Определение, описывающее процесс выполнения программы
- 4) Определение, которое описывает конечный процесс

42. Что такое полиморфизм в контексте языков программирования?

- 1) Способ, при котором объект может быть представлен разными способами
- 2) Способ, при котором объекты имеют одинаковую структуру данных
- 3) Способ, при котором все функции используются одинаково
- 4) Способ, при котором типы данных не зависят от контекста

43. Что такое динамическая типизация?

- 1) Преобразование типов данных во время компиляции
- 2) Преобразование типов данных во время выполнения программы
- 3) Определение типов данных на основе анализа кода
- 4) Статическое определение типов данных

44. Что такое лексический анализ?

- 1) Процесс перевода исходного кода в машинный код
- 2) Процесс выделения лексем из исходного кода программы
- 3) Процесс выполнения программы
- 4) Процесс оптимизации кода

45. Что такое многозадачность в вычислениях?

- 1) Способ выполнения нескольких программ одновременно
- 2) Способ использования многократных ресурсов для обработки данных
- 3) Способ улучшения работы с памятью
- 4) Способ исполнения одной программы с многими вариантами

Ключ к тесту:

1.2	2.3	3.2	4.2	5.2	6.1	7.1	8.1	9.2
10.1	11.1	12.2	13.3	14.1	15.2	16.1	17.2	18.2
19.2	20.3	21.3	22.3	23.1	24.1	25.1	26.1	27.2
28.3	29.2	30.2	31.2	32.1	33.2	34.2	35.1	36.1
37.1	38.1	39.1	40.2	41.1	42.1	43.2	44.2	45.1

Шкала оценивания результатов тестирования

% верных решений (ответов)	Шкала оценивания
85 - 100	отлично
70 - 84	хорошо
50- 69	удовлетворительно

6.2.4. Примеры задач при разборе конкретных ситуаций

Тема 1. История языков высокого уровня.

ПК-6.

1. Написать отчет о развитии языков высокого уровня, начиная с FORTRAN до современных языков, таких как Python и Go. Описать ключевые особенности и достижения каждого языка.

2. Сравнить два языка высокого уровня (например, C и Java) с точки зрения их применения в различных областях программирования. Привести примеры программ, написанных на каждом языке.

3. Провести исследование о влиянии теории вычислений на разработку языков высокого уровня. Описать, какие концепции были заимствованы из теории и как они повлияли на языки.

Тема 2. Представление о назначении и структуре языка Бэкуса-Наура.

ПК-6.

1. Написать описание языка Бэкуса-Наура (BNF) и его применения в описании синтаксиса языков программирования. Привести примеры грамматик на BNF для простых языков.

2. Создать грамматику на языке Бэкуса-Наура для описания математических выражений (например, выражения с операциями сложения и умножения). Описать, как грамматика обрабатывает приоритет операций.

3. Сравнить BNF с другими формальными языками, такими как EBNF и ABNF. Описать, в чем заключаются преимущества и недостатки каждого подхода.

Тема 3. Формальные языки и грамматики

ПК-6.

1. Определить основные понятия формальных языков и грамматик. Привести примеры формальных языков и их грамматик, а также объяснить их применение в программировании.

2. Создать контекстно-свободную грамматику для описания языка, содержащего простые арифметические выражения. Описать, как такая грамматика может быть использована в компиляторе.

3. Исследовать связь между формальными языками и конечными автоматами. Написать отчет о том, как формальные языки могут быть распознаны конечными автоматами.

Тема 4. Классификация по Хомскому.

ПК-6.

1. Описать классификацию формальных языков по Хомскому и привести примеры для каждого класса (регулярные, контекстно-свободные, контекстно-зависимые, рекурсивно перечислимые языки).

2. Исследовать применение классификации Хомского в компиляторах и интерпретаторах. Описать, как различные классы языков влияют на выбор алгоритмов анализа.

3. Провести исследование о том, как классификация по Хомскому помогает в понимании сложности вычислений и алгоритмов.

Тема 5. Преобразования грамматик

ПК-6.

1. Описать различные виды преобразований грамматик (например, нормализация, упрощение). Привести примеры преобразования грамматик в нормальную форму Хомского.

2. Создать программу, которая автоматически преобразует заданную грамматику в нормальную форму Хомского. Описать алгоритм, используемый для преобразования.

3. Исследовать влияние преобразования грамматик на эффективность синтаксического анализа. Написать отчет с примерами.

Тема 6. Дерево разбора (дерево вывода)

ПК-6.

1. Описать структуру дерева разбора и его назначение в синтаксическом анализе. Привести пример дерева разбора для арифметического выражения.

2. Создать программу, которая строит дерево разбора для заданного арифметического выражения. Описать алгоритм построения дерева.

3. Исследовать, как деревья разбора могут быть использованы для выполнения семантического анализа. Привести примеры использования деревьев в компиляторах.

Тема 7. Ограниченность контекстно-свободной грамматики

ПК-6.

1. Описать ограничения контекстно-свободных грамматик и привести примеры языков, которые не могут быть описаны такими грамматиками.

2. Исследовать проблемы, связанные с обработкой контекстно-свободных языков в компиляторах. Описать, как эти проблемы могут быть решены.

3. Провести эксперимент по сравнению контекстно-свободных и контекстно-зависимых грамматик. Написать отчет с выводами о том, какие задачи можно решить с помощью каждой из них.

Тема 8. Конечные автоматы

ПК-6.

1. Описать концепцию конечных автоматов и их применение в теории формальных языков. Привести примеры конечных автоматов для различных языков.

2. Создать программу, которая реализует конечный автомат для распознавания двоичных чисел, делящихся на 3. Описать переходы и состояния автомата.

3. Исследовать связь между регулярными выражениями и конечными автоматами. Написать отчет о том, как регулярные выражения могут быть преобразованы в конечные автоматы.

Тема 9. Минимизация конечных автоматов

ПК-6.

1. Описать алгоритмы минимизации конечных автоматов (например, алгоритм Хопкрофта). Привести примеры применения этих алгоритмов.

2. Создать программу, которая минимизирует конечный автомат и выводит результат в виде нового автомата. Описать алгоритм, использованный для минимизации.

3. Исследовать, как минимизация конечных автоматов влияет на производительность программ, использующих их. Написать отчет с примерами.

Тема 10. Построение лексического анализатора.

ПК-6.

1. Описать процесс построения лексического анализатора и его роль в компиляции. Привести примеры лексических анализаторов для различных языков.

2. Создать простой лексический анализатор для языка, поддерживающего арифметические выражения. Описать, как анализатор распознает токены.

3. Исследовать различные методы реализации лексических анализаторов (например, таблицы символов, конечные автоматы). Написать отчет с выводами о преимуществах и недостатках каждого метода.

Шкала оценивания

Шкала оценивания	Критерии оценивания
«Отлично»	обучающийся ясно изложил условие задачи, решение обосновал
«Хорошо»	обучающийся ясно изложил условие задачи, но в обосновании решения имеются сомнения;
«Удовлетворительно»	обучающийся изложил решение задачи, но обосновал его формулировками обыденного мышления;
«Неудовлетворительно»	обучающийся не уяснил условие задачи, решение не обосновал либо не сдал работу на проверку (в случае проведения решения задач в письменной форме).

6.2.5. Индивидуальные задания для курсовой работы (проекта)

КР и КП по дисциплине «Теория вычислительных процессов и языков программирования» рабочей программой и учебным планом не предусмотрены.

6.3. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

Вопросы для подготовки к промежуточной аттестации по итогам освоения дисциплины Теория вычислительных процессов и языков программирования:

ПК-6.

1. История языков высокого уровня.
2. Фортран (Fortran).
3. Алгол-60.
4. Кобол. ПЛ/1.
5. Бейсик..
6. Паскаль.
7. Ада.

8. Модуль-2.
9. Си++.
10. Схема работы интерпретатора. Схема компилятора/интерпретатора
11. Лексический анализатор
12. Синтаксический анализатор Синтаксический анализ
13. Преобразования грамматик
14. Исключение цепных правил.
15. Исключение бесполезных символов.
16. Нормальная форма Хомского.
17. Алгоритм удаления недостижимых символов. Алгоритм удаления бесплодных символов.
18. Алгоритм приведения грамматики.
19. КС-грамматика
20. Ограниченность контекстно-свободной грамматики
21. Таблица разбора.
22. Теорема (о накачке КС-языка).
23. Нормальная форма Грейбах.
24. Регулярные грамматики
25. Минимизация конечных автоматов
26. Построение лексического анализатора
27. Графы (диаграммы) состояний.
28. Детерминированные и недетерминированные автоматы.
29. Цепочки вывода. Построение таблиц идентификаторов по методу бинарного дерева. Хэш-функции и хэш-адресация.
30. Принципы работы хэш-функций. Построение таблиц идентификаторов на основе хэш-функции.
31. Построение таблиц идентификаторов по методу цепочек.
32. Комбинированные способы построения таблиц идентификаторов.
33. Формальные языки и грамматики: понятия и классификация.
34. Автоматы и их связь с формальными языками.
35. Регулярные выражения и их применение.
36. Конечные автоматы: определение, свойства и виды.
37. Недетерминированные конечные автоматы (НКА) и детерминированные конечные автоматы (ДКА).
38. Методы преобразования НКА в ДКА.
39. Минимизация автоматов: задачи и алгоритмы.
40. Понятие контекстно-свободной грамматики (КС-грамматики).
41. Построение синтаксических деревьев.
42. Парсеры: нисходящие и восходящие методы.
43. LL(1)-парсеры и их свойства.
44. LR- и LALR-парсеры.
45. Семантический анализ и его роль в компиляции.
46. Области видимости и лексические окружения.
47. Типизация и типовые системы в языках программирования.
48. Динамическая и статическая типизация.

49. Механизмы управления памятью: стек, куча.
50. Особенности интерпретируемых и компилируемых языков.
51. Основные этапы компиляции.
52. Средства оптимизации в компиляторах.
53. Модель вычислений: машина Тьюринга.
54. Алгоритмическая сложность и вычислимость.
55. Нерекурсивные и рекурсивные функции.
56. Понятие и примеры неразрешимых задач.
57. Конвейерная обработка данных в вычислительных системах.
58. Понятие трансляторов и ассемблеров.
59. Макроязыки и препроцессоры.
60. Языки сценариев и их применение.
61. Сравнение императивных и функциональных языков.
62. Функциональные языки: основные принципы.
63. Лямбда-исчисление и его роль в теории языков.
64. Типы данных в языках программирования.
65. Особенности объектно-ориентированных языков.
66. Методы обработки исключений.
67. Концепция абстракции в языках программирования.
68. Модульность и системы сборки.
69. Системы типов и проверка типов во время компиляции и выполнения.
70. Понятие языка высокого уровня и низкого уровня.
71. Виртуальные машины и байт-код.
72. Механизмы отражения и метапрограммирования.
73. Принципы работы Just-In-Time (JIT) компиляции.
74. Языки программирования с автоматическим управлением памятью.
75. Перспективы развития языков программирования.

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

Основной целью проведения промежуточной аттестации является определение степени достижения целей по учебной дисциплине или ее разделам. Осуществляется это проверкой и оценкой уровня теоретической знаний, полученных обучающимися, умения применять их в решении практических задач, степени овладения обучающимися практическими навыками и умениями в объеме требований рабочей программы по дисциплине, а также их умение самостоятельно работать с учебной литературой.

Организация проведения промежуточной аттестации регламентирована «Положением об организации образовательного процесса в федеральном государственном автономном образовательном учреждении «Московский политехнический университет»

6.4.1. Показатели оценивания компетенций на различных этапах их формирования, достижение обучающимися планируемых результатов обучения по дисциплине

Код и наименование компетенции ПК-6. Способен организовывать и технологически обеспечивать кодирование на языках программирования.				
Этап (уровень)	Критерии оценивания			
	неудовлетворительно	удовлетворительно	хорошо	отлично
знать	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: -основные результаты теории формальных языков; - формальные модели основных вычислительных процессов, методы анализа вычислительных процессов; - теоретические основы методов проектирования и способы описания языков программирования, стандарты, используемые для языков программирования	Обучающийся демонстрирует неполное соответствие следующих знаний: -основные результаты теории формальных языков; - формальные модели основных вычислительных процессов, методы анализа вычислительных процессов; - теоретические основы методов проектирования и способы описания языков программирования, стандарты, используемые для языков программирования	Обучающийся демонстрирует частичное соответствие следующих знаний: -основные результаты теории формальных языков; - формальные модели основных вычислительных процессов, методы анализа вычислительных процессов; - теоретические основы методов проектирования и способы описания языков программирования, стандарты, используемые для языков программирования	Обучающийся демонстрирует полное соответствие следующих знаний: -основные результаты теории формальных языков; - формальные модели основных вычислительных процессов, методы анализа вычислительных процессов; - теоретические основы методов проектирования и способы описания языков программирования, стандарты, используемые для языков программирования
уметь	Обучающийся не умеет или в недостаточной степени умеет - создавать распознаватели, интерпретаторы и трансляторы информационных потоков; - находить и устранять проблем взаимодействия вычислительных процессов; - моделировать работу алгоритмов взаимодействия процессов и ресурсов в работе вычислительной системы;	Обучающийся демонстрирует неполное соответствие следующих умений: - создавать распознаватели, интерпретаторы и трансляторы информационных потоков; - находить и устранять проблем взаимодействия вычислительных процессов; - моделировать работу алгоритмов взаимодействия процессов и ресурсов в	Обучающийся демонстрирует частичное соответствие следующих умений: - создавать распознаватели, интерпретаторы и трансляторы информационных потоков; - находить и устранять проблем взаимодействия вычислительных процессов; - моделировать работу алгоритмов взаимодействия	Обучающийся демонстрирует полное соответствие следующих умений: - создавать распознаватели, интерпретаторы и трансляторы информационных потоков; - находить и устранять проблем взаимодействия вычислительных процессов; - моделировать работу алгоритмов

	-разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик	работе вычислительной системы; -разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик	процессов и ресурсов в работе вычислительной системы; -разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик	взаимодействия процессов и ресурсов в работе вычислительной системы; -разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик
владеть	Обучающийся не владеет или в недостаточной степени владеет - навыками программного построения распознавателей, интерпретаторов и трансляторов информационных потоков; - навыками использования стандартных инструментальных средств построения распознавателей и трансляторов в системном программировании; -навыками практического анализа проблем взаимодействия вычислительных процессов.	Обучающийся владеет в неполном объеме и проявляет недостаточность владения - навыками программного построения распознавателей, интерпретаторов и трансляторов информационных потоков; - навыками использования стандартных инструментальных средств построения распознавателей и трансляторов в системном программировании; - н а в ы к а м и практического анализа проблем взаимодействия вычислительных процессов.	Обучающимся допускаются незначительные ошибки, неточности, затруднения, частично владеет - навыками программного построения распознавателей, интерпретаторов и трансляторов информационных потоков; - навыками использования стандартных инструментальных средств построения распознавателей и трансляторов в системном программировании; - н а в ы к а м и практического анализа проблем взаимодействия вычислительных процессов.	Обучающийся свободно применяет полученные навыки, в полном объеме владеет - навыками программного построения распознавателей, интерпретаторов и трансляторов информационных потоков; - навыками использования стандартных инструментальных средств построения распознавателей и трансляторов в системном программировании; - н а в ы к а м и практического анализа проблем взаимодействия вычислительных процессов.

6.4.2. Методика оценивания результатов промежуточной аттестации

Показателями оценивания компетенций на этапе промежуточной аттестации по дисциплине «Теория вычислительных процессов и языков программирования» являются результаты обучения по дисциплине.

Оценочный лист результатов обучения по дисциплине

Код компетенции	Знания	Умения	Навыки	Уровень
-----------------	--------	--------	--------	---------

				сформированности компетенции на данном этапе / оценка
ПК-6 Способен организовывать и технологически обеспечивать кодирование на языках программирования	на уровне знаний: знать основные результаты теории формальных языков; знать формальные модели основных вычислительных процессов, методы анализа вычислительных процессов; знать теоретические основы методов проектирования и способы описания языков программирования, стандарты, используемые для языков программирования	на уровне умений: уметь создавать распознаватели, интерпретаторы и трансляторы информационных потоков; уметь находить и устранять проблем взаимодействия вычислительных процессов; уметь моделировать работу алгоритмов взаимодействия процессов и ресурсов в работе вычислительной системы; уметь разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик	на уровне навыков: навыки программного построения распознавателей, интерпретаторов и трансляторов информационных потоков; навыки использования стандартных инструментальных средств построения распознавателей и трансляторов в системном программировании; навыки практического анализа проблем взаимодействия вычислительных процессов.	
Оценка по дисциплине (среднее арифметическое)				

Оценка «отлично» выставляется, если среднее арифметическое находится в интервале от 4,5 до 5,0.

Оценка «хорошо» выставляется, если среднее арифметическое находится в интервале от 3,5 до 4,4.

Оценка «удовлетворительно» выставляется, если среднее арифметическое находится в интервале от 2,5 до 3,4.

Оценка «неудовлетворительно» выставляется, если среднее арифметическое находится в интервале от 0 до 2,4.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по дисциплине «Теория вычислительных процессов и языков программирования», при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем,

ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7. Электронная информационно-образовательная среда

Каждый обучающийся в течение всего периода обучения обеспечивается индивидуальным неограниченным доступом к электронной информационно-образовательной среде Чебоксарского института (филиала) Московского политехнического университета из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), как на территории филиала, так и вне ее.

Электронная информационно-образовательная среда – совокупность информационных и телекоммуникационных технологий, соответствующих технологических средств, обеспечивающих освоение обучающимися образовательных программ в полном объеме независимо от места нахождения обучающихся.

Электронная информационно-образовательная среда обеспечивает:

а) доступ к учебным планам, рабочим программам дисциплин (модулей), практик, электронным учебным изданиям и электронным образовательным ресурсам, указанным в рабочих программах дисциплин (модулей), практик;

б) формирование электронного портфолио обучающегося, в том числе сохранение его работ и оценок за эти работы;

в) фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения программы бакалавриата;

г) проведение учебных занятий, процедур оценки результатов обучения, реализация которых предусмотрена с применением электронного обучения, дистанционных образовательных технологий;

д) взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействия посредством сети «Интернет».

Функционирование электронной информационно-образовательной среды обеспечивается соответствующими средствами информационно-коммуникационных технологий и квалификацией работников, ее использующих и поддерживающих.

Функционирование электронной информационно-образовательной среды соответствует законодательству Российской Федерации.

Основными составляющими ЭИОС филиала являются:

а) сайт института в сети Интернет, расположенный по адресу www.polytech21.ru, <https://chebpolytech.ru/> который обеспечивает:

- доступ обучающихся к учебным планам, рабочим программам дисциплин, практик, к изданиям электронных библиотечных систем, электронным информационным и образовательным ресурсам, указанных в рабочих программах (разделы сайта «Сведения об образовательной организации»);

- информирование обучающихся обо всех изменениях учебного процесса (новостная лента сайта, лента анонсов);

- взаимодействие между участниками образовательного процесса (подразделы сайта «Задать вопрос директору»);

б) официальные электронные адреса подразделений и сотрудников института с Яндекс-доменом @polytech21.ru (список контактных данных подразделений Филиала размещен на официальном сайте Филиала в разделе «Контакты», списки контактных официальных электронных данных преподавателей размещены в подразделах «Кафедры») обеспечивают взаимодействие между участниками образовательного процесса;

в) личный кабинет обучающегося (портфолио) (вход в личный кабинет размещен на официальном сайте Филиала в разделе «Студенту» подразделе «Электронная информационно-образовательная среда») включает в себя портфолио студента, электронные ведомости, рейтинг студентов и обеспечивает:

- фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися,

- формирование электронного портфолио обучающегося, в том числе с сохранение работ обучающегося, рецензий и оценок на эти работы,

г) электронные библиотеки, включающие электронные каталоги, полнотекстовые документы и обеспечивающие доступ к учебно-методическим материалам, выпускным квалификационным работам и т.д.:

Чебоксарского института (филиала) - «ИРБИС»

д) электронно-библиотечные системы (ЭБС), включающие электронный каталог и полнотекстовые документы:

- ЭБС «ЛАНЬ» -<https://e.lanbook.com/>

- Образовательная платформа Юрайт - <https://urait.ru>

- IPR SMART -<https://www.iprbookshop.ru/>

е) платформа цифрового образования Политеха -<https://lms.mospolytech.ru/>

ж) система «Антиплагиат» -<https://www.antiplagiat.ru/>

з) система электронного документооборота DIRECTUM Standard — обеспечивает документооборот между Филиалом и Университетом;

и) система «IC Управление ВУЗом Электронный деканат» (Московский политехнический университет) обеспечивает фиксацию хода образовательного процесса, результатов промежуточной аттестации и результатов освоения образовательных программ обучающимися;

к) система «POLYTECH systems» обеспечивает информационное, документальное автоматизированное сопровождение образовательного процесса;

л) система «Абитуриент» обеспечивает документальное автоматизированное сопровождение работы приемной комиссии.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература

1. Волкова, В. Н. Теория информационных процессов и систем : учебник и практикум для академического бакалавриата / В. Н. Волкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 447 с. — (Высшее образование). — ISBN 978-5-534-05621-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/580893>.

2. Кудрявцев, В. Б. Дискретная математика. Теория однородных структур : учебник для вузов / В. Б. Кудрявцев, А. С. Подколзин, А. А. Болотов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2026. — 295 с. — (Высшее образование). — ISBN 978-5-534-02901-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/584578>.

3. Воропаева, О. Ф. Основы численного анализа динамических систем : учебник для вузов / О. Ф. Воропаева. — Москва : Издательство Юрайт, 2025. — 164 с. — (Высшее образование). — ISBN 978-5-534-18818-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/568925>.

Дополнительная литература

1. Подбельский, В. В. Программирование. Базовый курс C# : учебник для вузов / В. В. Подбельский. — Москва : Издательство Юрайт, 2025. — 369 с. —

(Высшее образование). — ISBN 978-5-534-10616-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560848>.

Периодика:

1. Известия Тульского государственного университета. Технические науки: Научный рецензируемый журнал. <https://tidings.tsu.tula.ru/tidings/index.php?id=technical&lang=ru&year=1>. - Текст : электронный.

2. Научный периодический журнал «Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика» : Научный рецензируемый журнал. <https://vestnik.susu.ru/cmi> - Текст : электронный.

3. Научный периодический журнал «Вестник Южно-Уральского государственного университета. Серия «Компьютерные технологии, управление, радиоэлектроника»: Научный рецензируемый журнал. <https://vestnik.susu.ru/ctcr> - Текст : электронный.

9. Профессиональные базы данных и информационно-справочные системы

Профессиональная база данных и информационно-справочные системы	Информация о праве собственности (реквизиты договора)
Университетская информационная система РОССИЯ https://uisrussia.msu.ru/	Тематическая электронная библиотека и база для прикладных исследований в области экономики, управления, социологии, лингвистики, философии, филологии, международных отношений, права. свободный доступ
научная электронная библиотека Elibrary http://elibrary.ru/	Научная электронная библиотека eLIBRARY.RU - это крупнейший российский информационно-аналитический портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты более 26 млн научных статей и публикаций, в том числе электронные версии более 5600 российских научно-технических журналов, из которых более 4800 журналов в открытом доступе свободный доступ
сайт Института научной информации по общественным наукам РАН. http://www.inion.ru	Библиографические базы данных ИНИОН РАН по социальным и гуманитарным наукам ведутся с начала 1980-х годов. Общий объём массивов составляет более 3 млн. 500 тыс. записей (данные на 1 января 2012 г.). Ежегодный прирост — около 100 тыс. записей. В базы данных включаются аннотированные описания книг и статей из журналов и сборников на 140 языках, поступивших в Фундаментальную библиотеку ИНИОН РАН. Описания статей и книг в базах данных снабжены шифром хранения и ссылками на полные тексты источников из Научной электронной библиотеки.
Федеральный портал «Российское образование»	Федеральный портал «Российское образование» – уникальный интернет-ресурс в сфере образования и науки.

образование» [Электронный ресурс] – http://www.edu.ru	Ежедневно публикует самые актуальные новости, анонсы событий, информационные материалы для широкого круга читателей. Еженедельно на портале размещаются эксклюзивные материалы, интервью с ведущими специалистами – педагогами, психологами, учеными, репортажи и аналитические статьи. Читатели получают доступ к нормативно-правовой базе сферы образования, они могут пользоваться самыми различными полезными сервисами – такими, как онлайн-тестирование, опросы по актуальным темам и т.д.
Информационные технологии – периодическое научно-техническое издание в области информационных технологий, автоматизированных систем и использования информатики в различных приложениях novtex.ru	Издательство выпускает теоретические и прикладные научно-технические журналы, обеспечивающие научной, производственной, обзорно-аналитической и образовательной информацией руководящих работников и специалистов промышленных предприятий, научных академических и отраслевых организаций, а также учебных заведений в области приоритетных направлений развития науки и технологий.
Ассоциация инженерного образования России http://www.ac-raee.ru/	Совершенствование образования и инженерной деятельности во всех их проявлениях, относящихся к учебному, научному и технологическому направлениям, включая процессы преподавания, консультирования, исследования, разработки инженерных решений, оказания широкого спектра образовательных услуг, обеспечения связей с общественностью, производством, наукой и интеграции в международное научно-образовательное пространство. свободный доступ

10. Программное обеспечение (лицензионное и свободно распространяемое), используемое при осуществлении образовательного процесса

Аудитория	Программное обеспечение	Информация о праве собственности (реквизиты договора, номер лицензии и т.д.)
№ 2196 Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин	Windows 7 OLPNLAcdmc	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License	Сублицензионный договор №977_1049.ЕП/25 от 10.12.2025
	Yandex браузер	Свободное распространяемое программное обеспечение (бессрочная лицензия)
	МТС Линк	Договор №2/2026 (091_168.ЕП/26) от 27.03.2026
	Microsoft Office Standard	номер лицензии-42661846 от

<p>(модулей) <u>Кабинет систем управления</u> <u>ООО «НПО «Каскад-ГРУП»</u></p>	<p>2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic (Microsoft Open License</p>	<p>30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)</p>
<p>№ 2066 Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/специалитета/магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) <u>Лаборатория «Программирования и баз данных»</u> <u>Лаборатория информационных технологий</u></p>	<p>Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year Educational Renewal License</p>	<p>Сублицензионный договор №977_1049.ЕП/25 от 10.12.2025</p>
	<p>Windows 7 OLPNLAcdmc Windows Server 2012</p>	<p>договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)</p>
	<p>MS Windows 10 Pro</p>	<p>договор № 392_469.223.3К/19 от 17.12.19 (бессрочная лицензия)</p>
	<p>Microsoft Office Standard 2019(Microsoft DreamSpark Premium Electronic Software Delivery Academic(Microsoft Open License</p>	<p>номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)</p>
	<p>Eclipse IDE for Java EE Developers, NET Framework, JDK 8, Microsoft SQL Server Express Edition, Microsoft Visio Professional, Microsoft Visual Studio, MySQL Installer for Windows, NetBeans, SQL Server Management Studio, Microsoft SQL Server Java Connector, Android Studio, IntelliJ IDEA.</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
	<p>КОМПАС-3D v20 и v21</p>	<p>Сублицензионный договор № Нп-22-00044 от 21.03.2022 (бессрочная лицензия)</p>
	<p>MathCADv.15</p>	<p>Сублиц.договор №39331/МОС2286 от 6.05.2013) номер лицензии-42661846 от 30.08.2007) (бессрочная лицензия)</p>
	<p>SimInTech</p>	<p>Отечественное программное обеспечение</p>
	<p>AdobeReader</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
	<p>AdobeFlashPlayer</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>
<p>Microsoft Visual Studio 2019</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>	
<p>Python 3.7</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>	
<p>PascalABC</p>	<p>свободно распространяемое программное обеспечение (бессрочная лицензия)</p>	
<p>AIMP</p>	<p>отечественное свободно распространяемое программное обеспечение (бессрочная лицензия)</p>	
<p>№ 1126 Помещение для</p>	<p>Kaspersky Endpoint Security для бизнеса – Расширенный Russian Edition. 150-249 Node 2 year</p>	<p>Сублицензионный договор №977_1049.ЕП/25 от 10.12.2025</p>

самостоятельной работы обучающихся	Educational Renewal License	
	Windows 7 OLPNLAcdmc	договор №Д03 от 30.05.2012) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	AdobeReader	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Гарант-справочно-правовая система	Договор №С-002-2025 от 09.01.2025
	Yandex браузер	свободно распространяемое программное обеспечение (бессрочная лицензия)
	Microsoft Office Standard 2007(Microsoft DreamSpark Premium Electronic Software Delivery Academic (Microsoft Open License	номер лицензии-42661846 от 30.08.2007) с допсоглашениями от 29.04.14 и 01.09.16 (бессрочная лицензия)
	МТС Линк	Договор №2/2026 (091_168.ЕП/26) от 27.03.2026
AIMP	отечественное свободно распространяемое программное обеспечение (бессрочная лицензия)	

11. Материально-техническое обеспечение дисциплины

Тип и номер помещения	Перечень основного оборудования и технических средств обучения
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Кабинет систем управления ООО «НПО «Каскад-ГРУП» № 2196 (г. Чебоксары, ул. К.Маркса, 60)	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; стенды <u>Технические средства обучения:</u> мультимедийное оборудование (проектор, экран)
Учебная аудитория для проведения учебных занятий всех видов, предусмотренных программой среднего профессионального образования/бакалавриата/ специалитета/ магистратуры, оснащенная оборудованием и техническими средствами обучения, состав которых определяется в рабочих программах дисциплин (модулей) Лаборатория «Программирования и баз данных» Лаборатория информационных технологий № 2066 (г. Чебоксары, ул. К.Маркса, 60)	<u>Оборудование:</u> комплект мебели для учебного процесса; доска учебная; автоматизированные рабочие места, автоматизированное рабочее место преподавателя, проектор и экран; маркерная доска; <u>Технические средства обучения:</u> компьютерная техника (процессор Core i3, оперативная память объемом не менее 8 Гб); сервер в лаборатории (8-ядерный процессор с частотой не менее 3 ГГц, оперативная память объемом не менее 16 Гб, жесткие диски общим объемом не менее 1 Тб; мультимедийное оборудование (телевизор)
Помещение для самостоятельной работы обучающихся № 1126 (г. Чебоксары, ул. К.Маркса, 60)	<u>Оборудование:</u> комплект мебели для учебного процесса; <u>Технические средства обучения:</u> компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Филиала

12. Методические указания для обучающегося по освоению дисциплины

Методические указания для занятий лекционного типа

В ходе лекционных занятий обучающемуся необходимо вести конспектирование учебного материала, обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Необходимо задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Целесообразно дорабатывать свой конспект лекции, делая в нем соответствующие записи из основной и дополнительной литературы, рекомендованной преподавателем и предусмотренной учебной программой дисциплины.

Методические указания для занятий лабораторного и семинарского (практического) типа.

Выполнению лабораторных (практических) работ предшествует проверка знаний студентов – их теоретической готовности к выполнению задания. Проверка знаний проводится в форме, которую определяет преподаватель дисциплины (тестирование, опрос).

При проведении лабораторных (практических) занятий выделяют следующие разделы:

- общие положения (перечень лабораторных или практических занятий);
- общие требования к выполнению работ, общие требования к выполнению отчета);
- инструкция по каждой работе;
- справочные материалы и т. д.

Лабораторные занятия позволяют развивать у обучающегося творческое теоретическое мышление, умение самостоятельно изучать литературу, анализировать практику; учат четко формулировать мысль, вести дискуссию, то есть имеют исключительно важное значение в развитии самостоятельного мышления.

Подготовка к практическому занятию включает два этапа. На первом этапе обучающийся планирует свою самостоятельную работу, которая включает: уяснение задания на самостоятельную работу; подбор основной и дополнительной литературы; составление плана работы, в котором определяются основные пункты предстоящей подготовки. Составление плана дисциплинирует и повышает организованность в работе.

Второй этап включает непосредственную подготовку к занятию, которая начинается с изучения основной и дополнительной литературы. Особое внимание при этом необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. Далее следует подготовить тезисы для выступлений по всем учебным вопросам, выносимым на практическое

занятие или по теме, вынесенной на дискуссию (круглый стол), продумать примеры с целью обеспечения тесной связи изучаемой темы с реальной жизнью.

Готовясь к докладу или выступлению в рамках интерактивной формы (дискуссия, круглый стол), при необходимости, следует обратиться за помощью к преподавателю.

Методические указания к самостоятельной работе.

Самостоятельная работа обучающегося является основным средством овладения учебным материалом во время, свободное от обязательных учебных занятий. Самостоятельная работа обучающегося над усвоением учебного материала по учебной дисциплине может выполняться в библиотеке университета, учебных кабинетах, компьютерных классах, а также в домашних условиях. Содержание и количество самостоятельной работы обучающегося определяется учебной программой дисциплины, методическими материалами, практическими заданиями и указаниями преподавателя.

Самостоятельная работа в аудиторное время может включать:

- 1) конспектирование (составление тезисов) лекций;
- 2) выполнение контрольных работ;
- 3) решение задач;
- 4) работу со справочной и методической литературой;
- 5) работу с нормативными правовыми актами;
- 6) выступления с докладами, сообщениями на семинарских занятиях;
- 7) защиту выполненных работ;
- 8) участие в оперативном (текущем) опросе по отдельным темам изучаемой дисциплины;
- 9) участие в беседах, деловых (ролевых) играх, дискуссиях, круглых столах, конференциях;
- 10) участие в тестировании и др.

Самостоятельная работа во внеаудиторное время может состоять из:

- 1) повторения лекционного материала;
- 2) подготовки к практическим занятиям;
- 3) изучения учебной и научной литературы;
- 4) изучения нормативных правовых актов (в т.ч. в электронных базах данных);
- 5) решения задач, и иных практических заданий;
- 6) подготовки к контрольным работам, тестированию и т.д.;
- 7) подготовки к практическим занятиям устных докладов (сообщений);
- 8) выполнения курсовых работ, предусмотренных учебным планом;
- 9) выполнения выпускных квалификационных работ и др.;
- 10) выделения наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями на консультациях;
- 11) проведения самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, докладов;

12) текущий контроль осуществляется в форме устных, тестовых опросов, докладов;

В случае пропусков занятий, наличия индивидуального графика обучения и для закрепления практических навыков студентам могут быть выданы типовые индивидуальные задания, которые должны быть сданы в установленный преподавателем срок.

13. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Обучение по данной дисциплине инвалидов и лиц с ограниченными возможностями здоровья (далее ОВЗ) осуществляется преподавателем с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

Для студентов с нарушениями опорно-двигательной функции и с ОВЗ по слуху предусматривается сопровождение лекций и практических занятий мультимедийными средствами, раздаточным материалом.

Для студентов с ОВЗ по зрению предусматривается применение технических средств усиления остаточного зрения, а также предусмотрена возможность разработки аудиоматериалов.

По данной дисциплине обучение инвалидов и лиц с ограниченными возможностями здоровья может осуществляться как в аудитории, так и с использованием электронной информационно-образовательной среды, образовательного портала и электронной почты.

ЛИСТ ДОПОЛНЕНИЙ И ИЗМЕНЕНИЙ

рабочей программы дисциплины

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от « _____ » _____ 202__ г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от « _____ » _____ 202__ г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от « _____ » _____ 202__ г.

Внесены дополнения и изменения _____

Рабочая программа дисциплины рассмотрена, обсуждена и одобрена для исполнения в 202__-202__ учебном году на заседании кафедры, протокол № от « _____ » _____ 202__ г.

Внесены дополнения и изменения _____
