

Направление подготовки 09.03.01 «Информатика и вычислительная техника»
Профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»
РПД Б1.В.ДВ.04.02 «Теория формальных грамматик»



**Филиал федерального государственного бюджетного образовательного учреждения
высшего образования
«Национальный исследовательский университет «МЭИ»
в г. Смоленске**

УТВЕРЖДАЮ

Директор филиала

ФГБОУ ВО «НИУ «МЭИ» в г. Смоленске

д-р техн. наук, профессор



А.С. Федулов

2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ТЕОРИЯ ФОРМАЛЬНЫХ ГРАММАТИК

(НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ)

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Профиль **«Программное обеспечение средств вычислительной техники и автоматизированных систем»**

Уровень высшего образования: **бакалавриат**

Нормативный срок обучения: **4 года**

Форма обучения: **очная**

Год набора: **2023**

Смоленск – 2023 г.

Направление подготовки 09.03.01 «Информатика и вычислительная техника»
Профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»
РПД Б1.В.ДВ.04.02 «Теория формальных грамматик»



Программа составлена с учетом ФГОС ВО – бакалавриата по направлению подготовки 09.03.01 «Информатика и вычислительная техника», утвержденного приказом Минобрнауки России от «19» сентября 2017 г. № 929

Программу составил:
к.т.н., доцент кафедры «Вычислительная техника»

Я.А. Федулов

«05» июня 2023 г.

Программа обсуждена и одобрена на заседании кафедры «Вычислительная техника»
«07» июня 2023 г., протокол № 8

Заведующий кафедрой вычислительной техники
д.т.н., профессор

А.С. Федулов

«07» июня 2023 г.

РПД адаптирована для лиц с ограниченными возможностями здоровья и инвалидов

Ответственный в филиале по работе
с ЛОВЗ и инвалидами

Е.В. Зуева

«07» июня 2023 г.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины «Теория формальных грамматик» является формирование знаний, умений и навыков в определении требований к информационным системам, программному обеспечению и методам построения трансляторов с алгоритмических языков на основе современных технологий программирования.

Задачами дисциплины являются: изучение основных понятий и определений, классификаций программного обеспечения и алгоритмических языков; получение знаний об основных этапах создания трансляторов программных языков в рамках жизненного цикла, при современном состоянии технологий разработки программных продуктов; выработка умений в анализе исходную документацию и разработки документации по информационным системам и программному обеспечению; получение навыков оценки процессов создания трансляторов программного обеспечения; выработка практических навыков проектирования трансляторов программных языков.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Дисциплина «Теория формальных грамматик» относится к части, формируемой участниками образовательных отношений профессионального цикла Б1.В.ДВ.04.02 основной образовательной программы подготовки бакалавров по направлению «09.03.01 «Информатика и вычислительная техника».

Для изучения данной дисциплины в траектории формирования профессиональной компетенции ПК-8, необходимы знания, умения и навыки, формируемые предшествующими дисциплинами.

Б1.В.15 Технологии объектного программирования (ПК-8).

Перечень последующих дисциплин, для которых необходимы знания, умения и навыки, формируемые данной дисциплиной.

Б1.В.14 Тестирование программного обеспечения (ПК-8).

Б1.В.ДВ.04.02 Теория формальных грамматик (ПК-8).

Б2.В.02(П) Технологическая (проектно-технологическая) практика (УК-8; ПК-3; ПК-4; ПК-5; ПК-6; ПК-8).

Б2.В.04(Пд) Преддипломная практика (ПК-1; ПК-2; ПК-3; ПК-4; ПК-5; ПК-6; ПК-7; ПК-8).

Б3.01 Подготовка к защите и защита выпускной квалификационной работы (ПК-1; ПК-2; ПК-3; ПК-4; ПК-5; ПК-6; ПК-7; ПК-8).

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины направлено на формирование элементов следующих компетенций в соответствии с ФГОС ВО и ОП ВО по данному направлению подготовки (специальности):

ПК-8 - Способен выполнять работы по проектированию, интеграции и тестированию компонентов системного программного обеспечения.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы достижения компетенций	Результаты обучения
ПК-8. Способен выполнять работы по проектированию, интеграции и тестированию компонентов системного программного обеспечения	ПК-8.1. Выполняет работы по проектированию компонентов системного программного обеспечения	<p>Знает: возможности информационных систем и баз данных; устройство и функционирование современных информационных систем; современные стандарты информационного взаимодействия систем; инструменты и методы анализа требований; характеристики различных систем обеспечения безопасности, влияющие на производительность информационных систем и баз данных.</p> <p>Умеет: анализировать исходную документацию и разрабатывать документацию по информационным системам и базам данных; настраивать параметры инструментов системы безопасности в соответствии с установленными критериями.</p> <p>Владеет: навыками анализа функциональных и нефункциональных требований к информационным системам и базам данных; навыками документирования и верификации требований к информационным системам и базам данных.</p>
	ПК-8.2. Выполняет работы по интеграции компонентов системного программного обеспечения	<p>Знает: возможности информационных систем и баз данных; устройство и функционирование современных информационных систем; современные стандарты информационного взаимодействия систем; инструменты и методы анализа требований; характеристики различных систем обеспечения безопасности, влияющие на производительность</p>

	<p>информационных систем и баз данных.</p> <p><i>Умеет:</i> анализировать исходную документацию и разрабатывать документацию по информационным системам и базам данных; настраивать параметры инструментов системы безопасности в соответствии с установленными критериями.</p> <p><i>Владеет:</i> навыками анализа функциональных и нефункциональных требований к информационным системам и базам данных; навыками документирования и верификации требований к информационным системам и базам данных.</p>
<p>ПК-8.3. Выполняет работы по тестированию компонентов системного программного обеспечения</p>	<p><i>Знает:</i> возможности информационных систем и баз данных; устройство и функционирование современных информационных систем; современные стандарты информационного взаимодействия систем; инструменты и методы анализа требований; характеристики различных систем обеспечения безопасности, влияющие на производительность информационных систем и баз данных.</p> <p><i>Умеет:</i> анализировать исходную документацию и разрабатывать документацию по информационным системам и базам данных; настраивать параметры инструментов системы безопасности в соответствии с установленными критериями.</p> <p><i>Владеет:</i> навыками анализа функциональных и нефункциональных требований к информационным системам и базам данных; навыками документирования и верификации требований к информационным системам и базам данных.</p>

Направление подготовки 09.03.01 «Информатика и вычислительная техника»
 Профиль «Программное обеспечение средств вычислительной техники и автоматизированных систем»
 РПД Б1.В.ДВ.04.02 «Теория формальных грамматик»



4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Индекс	Наименование	Семестр 7										Итого за курс									
		Кон- троль	Академических часов								з.е.	Кон- троль	Академических часов							з.е.	
			Всего	Кон такт.	Лек	Лаб	Пр	КРП	СР	Кон- троль			Всего	Кон такт.	Лек	Лаб	Пр	КРП	СР		Кон- троль
Б1.В.Д В.04.02	Теория формальных грамматик	Экз КР	144	42	18	16		8	66	36	4	Экз КР	144	42	18	16		8	66	36	4

ОБОЗНАЧЕНИЯ:

Виды промежуточной аттестации (виды контроля):

Экз – экзамен;

ЗаО – зачет с оценкой;

За – зачет;

Виды работ:

Контакт. – контактная работа обучающихся с преподавателем;

Лек. – лекционные занятия;

Лаб.– лабораторные работы;

Пр. – практические занятия;

КРП – курсовая работа (курсовой проект);

РГР – расчетно-графическая работа (реферат);

СР – самостоятельная работа студентов;

з.е. – объем дисциплины в зачетных единицах.

Содержание дисциплины:

№	Наименование видов занятий и тематик, содержание
1	<p>Лекционные занятия 9 шт. по 2 часа:</p> <p><i>1.1. Представления языков программирования.</i> Место компилятора в программном обеспечении. Структура компилятора. Машины Тьюринга. Марковские модели. Конечные автоматы. Алфавиты, цепочки, языки. Представления языков. Типы грамматик. Линейно-ограниченные автоматы и их связь с контекстно-зависимыми грамматиками. Связь машин Тьюринга и грамматик типа 0.</p> <p><i>1.2. Формальные грамматики.</i> Алфавиты, цепочки, языки. Представления языков. Типы грамматик. Классификация грамматик по Хомскому. Линейно-ограниченные автоматы и их связь с контекстно-зависимыми грамматиками. Связь машин Тьюринга и грамматик типа 0.</p> <p><i>1.3 Регулярные множества и выражения.</i> Регулярные грамматики. Атрибутные грамматики. Грамматики Хомского, формы Бэкуса-Наура (БНФ). Лексические анализаторы, реализация. Конструирование таблиц анализатора. Разбор сверху-вниз (предсказывающий разбор). Рекурсивный спуск. Разбор снизу-вверх (сдвиг-свертка). Элементы теории перевода. Синтаксически управляемый переход.</p> <p><i>1.4. Контекстно-свободные грамматики.</i> Семантика контекстно-свободных языков. Алгоритм проверки на заикленность. Структура таблиц идентификаторов, таблиц расстановки со списками. Таблицы, основанные на деревьях, программирование. Сравнение методов реализации таблиц.</p> <p><i>1.5. Элементы теории трансляции.</i> Обобщенная структура транслятора. Назначение и необходимость фазы лексического анализа. Методы лексического анализа. Грамматики и распознаватели для лексического анализа. Лексический анализ на основе конечных автоматов. Преобразователи с магазинной памятью. Схемы синтаксически управляемого перехода. Описание областей видимости и блочной структуры. Линеаризованные представления. Атрибутная схема для алгоритма сопоставления образцов.</p> <p><i>1.6. Общие принципы организации синтаксического разбора.</i> Назначение синтаксического разбора. Классификация методов синтаксического разбора. Последовательность разбора. Использование просмотра вперед. Использование возвратов. Применение автоматов с магазинной памятью для нисходящего разбора слева направо. Общая связь между грамматиками и автоматами с магазинной памятью. Программная реализация нисходящего автомата с магазинной памятью.</p> <p><i>1.7. Семантический анализ.</i> Назначение семантического анализа. Семантическая модель программы. Построение элементов таблицы имен. Синтаксически управляемый перевод. Использование синтаксически управляемого перевода при построении таблиц имен. Использование синтаксически управляемого перевода при работе с таблицей имен. Синтаксически управляемый перевод при генерации промежуточного Представления. Организация промежуточного представления программы.</p> <p><i>1.8. Генерация кода.</i> Генерация кода, этапы, основные понятия. Динамическая организация памяти. Назначение адресов, трансляция переменных и выражений. Трансляция объектно-ориентированных свойств языков программирования. Генерация оптимального кода методами синтаксического анализа. Выбор дерева вывода наименьшей стоимости.</p> <p><i>1.9. Системы автоматизации построения трансляторов.</i> Структуры систем, основные термины и определения и части программного кода реализации систем автоматизации построения трансляторов. Система СУПЕР, система Yacc, LEX. Разделы типов, констант, файлов, библиотек. Атрибутная схема.</p>

2	<p>лабораторные работы 4 шт. по 4 часа:</p> <p>2.1. <i>Алгоритмы построения конечных автоматов.</i> Построение Машины Тьюринга, Марковских моделей. Составление отчета.</p> <p>2.2. <i>Построение лексического анализатора.</i> Формальные грамматики, задачи. Построение лексического анализатора. Полный список выбранных ключевых слов и стандартных функций. Внутренние таблицы сканера. Отладочные примеры работы сканера с выходными таблицами и дескрипторным текстом.</p> <p>2.3. <i>Построение распознавателя. Использование таблицы имен.</i> Синтаксический анализ, группирование лексем исходной программы в грамматические фразы, используемые компилятором для синтеза вывода. Реализация дерева разбора, синтаксическое дерево. Алгоритм устранения недостижимых символов. Алгоритм устранения бесполезных символов. Алгоритм преобразования грамматики в грамматику без пустых продукций. Алгоритм устранения цепных продукций. Устранение в КС-грамматике левой рекурсии. Детерминированный синтаксический анализ сверху вниз.</p> <p>2.4. <i>Семантический анализ и генерация промежуточного представления.</i> Внутренние формы представления программы. Постфиксная запись. Представление генерируемого кода в форме четверок. Организация генератора кода. Методы управления памятью. Фазы управления памятью.</p>
3	<p>практические занятия не предусмотрены в структуре дисциплины</p>
4	<p>В процессе курсового проектирования, студенты разрабатывают программный комплекс на языке высокого уровня по свободному выбору.</p> <p>Целью работы является освоение методологий проектирования и реализации трансляторов программных языков на основе формальных грамматик и конечных автоматов.</p> <p>Разработка программного комплекса направлена на приобретение студентом практических навыков анализа задачи, ее декомпозиции, работы с различными структурами данных, типовыми алгоритмами, подпрограммами, стандартными модулями, сетевыми компонентами, средствами операционной системы.</p> <p>Студент должен разобраться с поставленной задачей, отладить ее на персональном компьютере, продемонстрировать преподавателю работу отлаженных программ в соответствии с вариантом задания и выполнить отчет по работе в соответствии с требованиями.</p> <p>При выполнении расчетного задания студент должен обеспечить выполнение следующих требований:</p> <ol style="list-style-type: none"> 1. Все решаемые в системе задачи должны быть реализованы в виде отдельных модулей. 2. Каждая подпрограмма/модуль должен начинаться с комментария, объясняющего его назначение, назначение переменных, передаваемых в него и из него. 3. В работе должен быть предусмотрен «дружественный» пользовательский интерфейс, организовано меню решаемых задач и вызов подпрограмм по выбору пункта меню. 4. Ввод данных должен сопровождаться проверкой корректности. 5. Реализация разработанной системы должна начинаться с названия программы, информации об ее назначении, группы, фамилии и имени автора разработки. <p>В ходе выполнения работы для каждого задания необходимо выполнить по соответствующему номеру варианта задания. Номер варианта курсового проекта N выбирается в соответствии с порядковым номером фамилии студента в журнале группы. Номер N используется при конкретизации заданий.</p>
5	<p>расчетно-графическая работа не предусмотрена в структуре дисциплины</p>
6	<p>Самостоятельная работа студентов:</p> <ol style="list-style-type: none"> 6.1. 2 контрольных опроса после 4-й и 9-й лекций; 6.2. Закрепление материала по тематике лекционных занятий: закрепление изучения материалов лекций 1.1-1.9 – основы программирования на языке вы-

	сокого уровня; классификация программного обеспечения и формальных грамматик; проектирование транслятора языка высокого уровня. 6.3. Подготовка к экзамену по дисциплине (оценочные материалы приведены в разделе 6 настоящей РПД).
--	---

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Таблица - Образовательные технологии, используемые при реализации различных видов учебной занятий по дисциплине

№ п/п	Виды учебных занятий	Образовательные технологии
1	Лекции	Классическая (традиционная, информационная) лекция. Интерактивная лекция (лекция-визуализация). Индивидуальные и групповые консультации по дисциплине.
2	Лабораторная работа	Технология выполнения лабораторных заданий индивидуально. Технология выполнения лабораторных заданий в малой группе (в бригаде). Технология проблемного обучения на основе анализа результатов лабораторной работы: индивидуальный опрос, собеседование в малой группе (бригаде), обсуждение результатов командной работы, групповая дискуссия, метод «круглого стола», представление студентом или группой студентов (бригадой) результатов лабораторной работы в форме отчета и мультимедийной презентации. Проектная технология.
3	Самостоятельная работа студентов (внеаудиторная)	Информационно-коммуникационные технологии (доступ к ЭИОС филиала, к ЭБС филиала, доступ к информационно-методическим материалам по дисциплине).
4	Контроль (промежуточная аттестация: зачет или экзамен)	Технология письменного контроля, в том числе тестирование. Рейтинговая система контроля.

6. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ – ДЛЯ ОЦЕНКИ КАЧЕСТВА ОСВОЕНИЯ ДИСЦИПЛИНЫ

К промежуточной аттестации студентов по дисциплине могут привлекаться представители работодателей, преподаватели последующих дисциплин, заведующие кафедрами.

Оценка качества освоения дисциплины включает как текущий контроль успеваемости, так и промежуточную аттестацию.

Оценочные средства текущего контроля успеваемости:

Примеры вопросов к контрольному опросу после 4-й лекции:

1. Даны грамматика и цепочка. Построить вывод заданной цепочки.

a) $S \rightarrow T \mid T+S \mid T-S$ $T \rightarrow F \mid F*T$ $F \rightarrow a \mid b$ Цепочка: $a-b*a+b$	б) $S \rightarrow aSBC \mid abC$ $CB \rightarrow BC$ $bB \rightarrow bb$ $bC \rightarrow bccC \rightarrow cc$ Цепочка: $aaabbbccc$
--	--

2. Построить все сентенциальные формы для грамматики с правилами:

$$S \rightarrow A+B \mid B+A$$

$$A \rightarrow a$$

$$B \rightarrow b$$

3. К какому типу по Хомскому относится данная грамматика? Какой язык она порождает? Каков тип языка? Указать максимально возможный номер типа грамматики и языка.

a) $S \rightarrow APA$ $P \rightarrow + \mid -$ $A \rightarrow a \mid b$	б) $S \rightarrow A \mid SA \mid SB$ $A \rightarrow a$ $B \rightarrow b$
--	--

4. Построить грамматику, порождающую заданный язык L . Каков тип построенной грамматики? Каков тип языка?

a) $L = \{ a^n b^m \mid n, m \geq 1 \};$

б) $L = \{ \alpha\beta\gamma \mid \alpha, \beta, \gamma \in \{a,b\}^* \text{ (т.е. } \alpha, \beta, \gamma \text{ — любые цепочки из } a \text{ и } b \text{)} \};$

в) $L = \{ a_1 a_2 \dots a_n a_n \dots a_2 a_1 \mid a_i \in \{0, 1\}, n \geq 1 \}.$

5. К какому типу по Хомскому относится данная грамматика (указать максимально возможный номер)? Какой язык она порождает? Каков тип языка? Выписать подтверждающую ответ грамматику, в состав которой входит только один нетерминал — цель грамматики.

а) $S \rightarrow AB \mid ASB$ $A \rightarrow a$ $aB \rightarrow b$ $bB \rightarrow bb$	б) $S \rightarrow 1A0$ $1A \rightarrow 11A0 \mid 01$
--	---

6. Эквивалентны ли грамматики с правилами:

а) $S \rightarrow AB$ $A \rightarrow a \mid Aa$ $B \rightarrow b \mid Bb$	и	$S \rightarrow AS \mid SB \mid AB$ $A \rightarrow a$ $B \rightarrow b$
---	---	--

$$\begin{array}{l} \text{b) } S \rightarrow aSL \mid aL \\ L \rightarrow Kc \\ cK \rightarrow Kc \\ K \rightarrow b \end{array} \quad \text{и} \quad \begin{array}{l} S \rightarrow aSBc \mid abc \\ cB \rightarrow Bc \\ bB \rightarrow bb \end{array}$$

7. Построить КС-грамматику, эквивалентную грамматике с правилами:

$$\begin{array}{l} \text{a) } S \rightarrow aAb \\ aA \rightarrow aaAb \\ A \rightarrow \varepsilon \\ A \rightarrow a \end{array} \quad \text{b) } \begin{array}{l} S \rightarrow AB \mid ABS \\ AB \rightarrow BA \\ BA \rightarrow AB \\ B \rightarrow b \end{array}$$

8. Построить регулярную грамматику, эквивалентную грамматике с правилами:

$$\begin{array}{l} \text{a) } S \rightarrow A \mid AS \\ A \rightarrow a \mid bb \end{array} \quad \text{b) } \begin{array}{l} S \rightarrow A.A \\ A \rightarrow B \mid BA \\ B \rightarrow 0 \mid 1 \end{array}$$

9. Привести пример грамматики, каждое правило которой относится к одному из трех видов $A \rightarrow Bt$, либо $A \rightarrow tB$, либо $A \rightarrow t$, для которой не существует эквивалентной регулярной грамматики.

10. Доказать, что грамматика с правилами:

$$\begin{array}{l} S \rightarrow aSBC \mid abC \\ CB \rightarrow BC \\ bB \rightarrow bb \\ bC \rightarrow bc \\ cC \rightarrow cc \end{array}$$

порождает язык $L = \{a^n b^n c^n \mid n \geq 1\}$.

Примеры вопросов к контрольному опросу после 9-й лекции:

1. Общие сведения о трансляторах. Основные понятия и определения.
2. Общие особенности языков программирования и трансляторов. Обобщенная структура транслятора. Варианты взаимодействия блоков транслятора.
3. Многопроходная организация взаимодействия блоков транслятора. Однопроходная организация взаимодействия блоков транслятора. Комбинированные взаимодействия блоков транслятора
4. Основы теории языков и формальных грамматик. Способы определения языков. Формальные грамматики. Грамматики с ограничениями на правила. Способы записи синтаксиса языка. Метаязык Хомского. Метаязык Хомского-Щутценберже. Бэкуса-Наура формы (БНФ). Распознаватели
5. Демонстрационный язык программирования. Источники вдохновения для создания демонстрационного языка. Синтаксис и семантика DPL. Элементарные конструкции. Составные конструкции. Организация программы. Краткое описание семантики языка.
6. Примеры программ на DPL. Алгоритм Евклида (нахождение наибольшего общего делителя). Одновременное нахождение наибольшего общего делителя (НОД) и наименьшего общего кратного (НОК).

7. Примеры программ на DPL. Суммирование n элементов из входного потока Сортировка элементов вектора. Описание пользовательского синтаксиса с использованием диаграмм Вирта.
8. Организация лексического анализа. Назначение и необходимость фазы лексического анализа.
9. Транслитератор. Грамматика и распознаватели для лексического анализа. Связь между диаграммой Вирта и конечным автоматом. Связь между диаграммами Вирта и праволинейными грамматиками. Преобразование правой рекурсии в итерацию. Связь между диаграммами Вирта и грамматиками с левой рекурсией. Преобразование левой рекурсии в итерацию
10. Методы лексического анализа. Организация непрямого лексического анализатора. Организация прямого лексического анализатора

Примеры алгоритма самостоятельной работы по закреплению материала по тематике лекционных занятий:

В ходе изучения дисциплины «Теория формальных грамматик» студенты могут посещать аудиторские занятия (лекции, лабораторные занятия, консультации). Особенность изучения дисциплины состоит в выполнении комплекса лабораторных работ, главной задачей которого является получение навыков самостоятельной работы на компьютерах с использованием современных компьютерных программ, предназначенных для решения определенного круга профессиональных задач.

Важное место в овладении тем данной дисциплины отводится самостоятельной работе, при этом во время аудиторных занятий могут быть рассмотрены и проработаны наиболее важные и трудные вопросы по той или иной теме дисциплины, а более легкие вопросы могут быть изучены студентами самостоятельно.

Методика закрепления материалов лекционных занятий 1.1-1.9:

Закрепление знаний в области данной дисциплины, приобретение практических навыков проектирования программных систем осуществляется путем разработки программных средств по заданной предметной области:

1. Анализ требований на разработку.

Формирование требований к разрабатываемой системе; обследование объекта и обоснование необходимости создания программных средств; формирование требований пользователя к разрабатываемой системе; оформление отчета о выполнении работ и заявки на разработку программных средств; разработка концепции программного обеспечения; изучение объекта проектирования; Проведение необходимых исследовательских работ; разработка вариантов концепции и выбор варианта концепции программных средств, удовлетворяющего требованиям пользователей; оформление отчета о проделанной работе.

2. Разработка и утверждение технического задания.

Разработка и утверждение технического задания включает в себя: определение требований к программе; разработку технического обоснования разработки программы; определение стадий, этапов и сроков разработки программы и документации на нее; выбор языков программирования; определение необходимости проведения исследовательских работ на последующих стадиях; согласование и утверждение технического задания.

Результатом данной фазы является общее описание системы, включающее в себя требования к программе и требования к надежности программы. Требования к программному обеспечению включают: описание входных данных (корректных и ошибочных); роли при использовании программного обеспечения, и их интерфейсы (средств общения с пользователями); упрощения, предположения и допущения по отношению к программам; описание выходных данных; требования к надежности функционирования программы. Определяются ошибки, которые необходимо выявлять, и сообщения, которые желательно выдавать пользователю при наличии ошибок. Перечисляются все особые ситуации, требующие дополнительного учета и специального рассмотрения.

3. Проектирование программного обеспечения.

Проектирование включает в себя два этапа: эскизный проект и технический проект. Эскизный проект заключается в предварительной разработке структуры входных и выходных данных и общего описания алгоритма решения задачи. При разработке технического проекта структуры входных и выходных данных уточняются и определяются их формы представления. Уточняется алгоритм решения задачи, определяется семантика и синтаксис языка программирования и разрабатывается структура программы. Оба этапа сопровождаются пояснительной запиской и технико-экономическим обоснованием. В соответствии с технологией нисходящего структурного программирования программный комплекс разбивается на небольшие части – программные модули. Каждая отдельная подзадача должна быть относительно независимой и представлять собой некоторый законченный модуль программы. По итогу этапа должны быть спроектированы: словарь терминов, функциональные диаграммы IDEF0 или диаграммы потоков данных (DFD). Не менее трех уровней иерархии диаграмм (лучше больше). Оценка качества проектирования (коэффициент декомпозиции, коэффициент сбалансированности); диаграммы переходов состояний STD; схемы алгоритмов; диаграммы Джексона, структурные карты Константайна.

4. Реализация рабочего проекта.

Рабочий проект включает в себя разработку программы и программной документации, а также испытание программы. Этап кодирования алгоритмов заключается в переводе алгоритмов, разработанных для каждого программного модуля, в программы на конкретном языке программирования. Кодирование должно быть простым. Изошренное программирование может обойтись слишком дорого при отладке и модификации программы. Необычное кодирование (например, использование скрытых возможностей машины) часто препятствует отладке программы и, конечно, затрудняет ее использование другими программистами. Входные форматы должны быть разработаны с учетом максимального удобства для пользователя и минимальной возможности ошибок. Порядок переменных и форматы данных, привычные для пользователя, помогут избежать ошибок и облегчат использование программ. Форматы выходных данных могут сильно различаться. Иногда даются четкие инструкции, и выходные данные подгоняются под определенный стандарт. Результаты расчета должны быть удобочитаемыми и понятными для непрограммиста. По итогу этапа должны быть разработаны: структурные и функциональные схемы; описание разработанных процедур и функций (название, входные данные, выходные данные, ограничения); структурное тестирование (тестирование базового пути, тестирование условий, тестирование циклов), функциональное тестирование (разбиение на классы эквивалентности, анализ граничных значений, анализ причинно-следственных связей); реализация пользовательского интерфейса (построение графа диалога).

В заключении поводится оценка качества разработанного программного обеспечения.

Оценочные средства для промежуточной аттестации:

Примеры вопросов к экзамену по дисциплине:

1. Общие сведения о трансляторах. Основные понятия и определения.
2. Общие особенности языков программирования и трансляторов. Обобщенная структура транслятора. Варианты взаимодействия блоков транслятора.
3. Многопроходная организация взаимодействия блоков транслятора. Однопроходная организация взаимодействия блоков транслятора. Комбинированные взаимодействия блоков транслятора.
4. Основы теории языков и формальных грамматик. Способы определения языков. Формальные грамматики. Грамматики с ограничениями на правила. Способы записи синтаксиса языка. Метаязык Хомского. Бэкуса-Наура формы (БНФ).
5. Классификация грамматик по Хомскому. Иерархия Хомского.

6. Разбор по регулярным грамматикам. Алгоритм разбора по диаграмме состояний.
7. Разбор цепочек с помощью детерминированного и недетерминированного конечного автомата (ДКА и НКА).
8. Демонстрационный язык программирования (DPL). Источники вдохновения для создания демонстрационного языка. Синтаксис и семантика DPL. Элементарные конструкции. Составные конструкции. Организация программы. Краткое описание семантики языка.
9. Примеры программ на DPL. Алгоритм Евклида (нахождение наибольшего общего делителя). Одновременное нахождение наибольшего общего делителя (НОД) и наименьшего общего кратного (НОК).
10. Организация лексического анализа. Назначение и необходимость фазы лексического анализа.
11. Транслитератор. Грамматики и распознаватели для лексического анализа. Связь между диаграммой Вирта и конечным автоматом. Связь между диаграммами Вирта и праволинейными грамматиками. Преобразование правой рекурсии в итерацию. Связь между диаграммами Вирта и грамматиками с левой рекурсией. Преобразование левой рекурсии в итерацию.
12. Методы лексического анализа. Организация непрямого лексического анализатора. Организация прямого лексического анализатора.
13. Лексический анализатор демонстрационного языка программирования. Транслитератор DPL. Общая организация транслитератора. Программная реализация транслитератора. Непрямой лексический анализатор DPL.
14. Диаграммы Вирта для отдельных автоматов непрямого лексического анализатора. Программная реализация отдельных автоматов. Общая структура непрямого лексического анализатора. Прямой лексический анализатор DPL.
15. Общие принципы организации синтаксического разбора. Назначение синтаксического разбора. Классификация методов синтаксического разбора.
16. Методы разбора. Последовательность разбора. Использование просмотра вперед. Использование возвратов.
17. Применение автоматов с магазинной памятью для нисходящего разбора слева направо. Необходимость использования автоматов с магазинной памятью. Организация автомата с магазинной памятью.
18. Общая связь между грамматиками и автоматами с магазинной памятью. Связь между S-грамматикой и автоматом с магазинной памятью. Обобщенный алгоритм построения нисходящего АМП для S – грамматики. S-грамматика и распознавание вложенности скобок.
19. Построение автомата с магазинной памятью по q-грамматике. Построение нисходящего автомата. Примеры построения АМП по q-грамматике. Распознавание вложенности скобок и q-грамматика.
20. LL(1) – грамматики. Программная реализация нисходящего автомата с магазинной памятью. Разработка программы по таблице переходов АМП. Разработка программы с использованием метода рекурсивного спуска.
21. Использование динамически порождаемых автоматов для нисходящего разбора слева направо. Семантический разрыв между формальными грамматиками и автоматами с магазинной памятью. Модель динамически порождаемых конечных автоматов. Графический метаязык для описания динамически порождаемых конечных автоматов. Использование диаграмм Вирта для представления динамически порождаемых конечных автоматов, распознающих КС(1) грамматику.
22. Применение диаграмм Вирта для построения КС(1) грамматик, используемых при нисходящем разборе слева направо. Избавление от левых или правых рекурсий. Преобразование исходных синтаксических правил к более простому виду. Преобразование КС(n) грамматик к грамматикам с просмотром вперед на меньшее число символов.

23. Проверка грамматик на эквивалентность. Освобождение от пустых правил. Непосредственное использование правил с левой рекурсией. Использование синтаксиса для определения семантики. Доказательство принадлежности к КС(1) грамматике.

24. Разработка синтаксических диаграмм для распознавателя DPL. Преобразование пользовательского синтаксиса к КС(1) грамматике. Замена обозначений нетерминалов в исходных диаграммах Вирта. Диаграммы Вирта, полученные после подстановки лексем и замены названий на идентификаторы.

25. Освобождение от сквозных связей в демонстрационном языке. Модификация правил с одинаковыми начальными лексемами. Проверка принадлежности к КС(1) грамматике.

26. Получение окончательных диаграмм Вирта демонстрационного языка. Окончательные диаграммы Вирта, описывающие синтаксис демонстрационного языка программирования DPL. Написание кода по разработанным диаграммам Вирта

27. Семантический анализ. Назначение семантического анализа. Семантическая модель программы. Построение дерева разбора. Построение объектной модели. Связь между построением семантической модели программы и генерацией кода.

28. Синтаксически управляемый перевод. Использование синтаксически управляемого перевода при построении таблиц имен. Построение таблицы имен для демонстрационного языка.

29. Программная реализация построения таблицы имен для демонстрационного языка. Использование синтаксически управляемого перевода при работе с таблицей имен. Использование синтаксически управляемого перевода для работы с таблицей имен в трансляторе с демонстрационного языка.

30. Синтаксически управляемый перевод при генерации промежуточного представления. Организация промежуточного представления программы. Формирование команд виртуальной машины. Основные отношения на уровне команд. Структура команд языка промежуточного представления. Формат команды промежуточного представления и его реализация. Структура операнда демонстрационного языка.

31. Генерация промежуточного представления. Генерация структур, представляющих константные операнды. Генерация промежуточного представления для переменной. Генерация промежуточного представления для выражений.

Примеры практических заданий к экзамену по дисциплине:

1. Задача: Даны грамматика G :

$$S \rightarrow 0A1$$

$$G: A \rightarrow 0A0$$

$$A \rightarrow \varepsilon$$

Какого типа язык $L(G)$? Каков тип языка? Указать максимально возможный номер типа грамматики и языка.

2. Задача: Даны грамматика G :

$$G: S \rightarrow aSb \mid \varepsilon$$

Какого типа язык $L(G)$? Каков тип языка? Указать максимально возможный номер типа грамматики и языка.

3. Построить грамматику, порождающую заданный язык L .

$$L = \{a^n b^m \mid n, m \geq 1\}$$

Каков тип построенной грамматики?

4. Дана грамматика с правилами:

$$S \rightarrow S0 \mid S1 \mid D0 \mid D1$$

$$D \rightarrow H$$

$$H \rightarrow 0 \mid 1 \mid H0 \mid H1$$

Построить восходящим и нисходящим методами дерево вывода для цепочки: 10.1001.

5.

Дана регулярная грамматика с правилами:

$$S \rightarrow S0 \mid S1 \mid P0 \mid P1$$

$$P \rightarrow N.$$

$$N \rightarrow 0 \mid 1 \mid N0 \mid N1$$

Построить по ней диаграмму состояний (ДС) и использовать ДС для разбора цепочек: 11.010, 0.1 и , 100.

6. Задан АМП, проверяющий правильность расстановки скобок: Множество входных символов: { (,), \perp }; Множество магазинных символов: { A, ∇ }; Множество состояний: S, где S - единственное состояние автомата; 4) В начальном состоянии магазин содержит только маркер дна (∇). 5) Переходы:

$$(\downarrow, A, S \Rightarrow \downarrow A, S, \rightarrow \quad), \nabla, S \Rightarrow \text{Отвергнуть}$$

$$(\downarrow, \nabla, S \Rightarrow \downarrow A, S, \rightarrow \quad \perp, A, S \Rightarrow \text{Отвергнуть}$$

$$), A, S \Rightarrow \uparrow, S, \rightarrow \quad \perp, \nabla, S \Rightarrow \text{Допустить}$$

Продолжить таблицу переходов для проверки ()) \perp (3 строки)

Номер	Содержимое стека	Состояние АМП	Остаток входной цепочки	Номер применяемого правила
1	∇	S	()) \perp	2

7. Для следующих выражений в ПОЛИЗе дать обычную инфиксную запись:

$$ab + bc - / a + \quad 2x + 2x^* \leq$$

$$a \text{ not } b \text{ and not } abca \text{ and or and}$$

8. Используя стек, вычислить следующие выражения в ПОЛИЗе:

$$x \text{ y } *x \text{ y } / + \quad \text{при } x = 8, y = 2;$$

$$a \text{ 2+b / b 4*+} \quad \text{при } a = 4, b = 3;$$

$$a \text{ b not and a or not} \quad \text{при } a = b = \text{true.}$$

В филиале используется система с традиционной шкалой оценок – "отлично", "хорошо", "удовлетворительно", "неудовлетворительно", "зачтено", "не зачтено" (далее - пятибалльная система).

Форма промежуточной аттестации по настоящей дисциплине – экзамен в 7-м семестре.

Применяемые критерии оценивания по дисциплинам (в соответствии с инструктивным письмом НИУ МЭИ от 14 мая 2012 года № И-23):

Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
«отлично»/ «зачтено (отлично)»/ «зачтено»	Выставляется обучающемуся, обнаружившему всестороннее, систематическое и глубокое знание материалов изученной дисциплины, умение свободно выполнять задания, предусмотренные программой, усвоивший основную и знакомый с дополнительной литературой, рекомендованной рабочей программой дисциплины; проявившему творческие способности в понимании, изложении и использовании материалов изученной дисциплины, безупречно ответившему не только на вопросы билета, но и на дополнительные вопросы в рамках рабочей программы дисциплины, правильно выполнившему практическое задание. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «эталонный».
«хорошо»/ «зачтено (хорошо)»/ «зачтено»	Выставляется обучающемуся, обнаружившему полное знание материала изученной дисциплины, успешно выполняющему предусмотренные задания, усвоившему основную литературу, рекомендованную рабочей программой дисциплины; показавшему систематический характер знаний по дисциплине, ответившему на все вопросы билета, правильно выполнивший практическое задание, но допустивший при этом не принципиальные ошибки. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «продвинутый».
«удовлетворительно»/ «зачтено (удовлетворительно)»/ «зачтено»	Выставляется обучающемуся, обнаружившему знание материала изученной дисциплины в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, знакомому с основной литературой, рекомендованной рабочей программой дисциплины; допустившему погрешность в ответе на теоретические вопросы и/или при выполнении практических заданий, но обладающему необходимыми знаниями для их устранения под руководством преподавателя, либо неправильно выполнившему практическое задание, но по указанию преподавателя выполнившему другие практические задания из того же раздела дисциплины. Компетенции, закреплённые за дисциплиной, сформированы на уровне – «пороговый».
«неудовлетворительно»/ не зачтено	Выставляется обучающемуся, обнаружившему серьезные пробелы в знаниях основного материала изученной дисциплины, допустившему принципиальные ошибки в выполнении заданий, не ответившему на все вопросы билета и дополнительные вопросы и неправильно выполнившему практическое задание (неправильное выполнение только практического задания не является однозначной причиной для выставления оценки «неудовлетворительно»). Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине. Оценка по дисциплине выставляется обучающемуся с учётом результатов текущего контроля. Компетенции на уровне «пороговый», закреплённые за дисциплиной, не сформированы.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Учебное и учебно-лабораторное оборудование

Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная: специализированной мебелью; доской аудиторной; демонстрационным оборудованием: персональным компьютером (ноутбуком); переносным (стационарным) проектором.

Учебная аудитория для лабораторных работ, выполняемых в компьютерном классе, оснащенная: специализированной мебелью; доской аудиторной; персональными компьютерами с подключением к сети "Интернет" и доступом в ЭИОС филиала.

Для самостоятельной работы обучающихся по дисциплине используется помещение для самостоятельной работы обучающихся, оснащенное: специализированной мебелью; доской аудиторной; персональными компьютерами с подключением к сети "Интернет" и доступом в ЭИОС филиала.

Программное обеспечение (Операционная система 64 или 32 разрядной версии, офисный пакет программ, среды разработки и компиляторы языков *C#* и *python*, средство отрисовки графических векторных изображений).

8. ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

для слепых и слабовидящих:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
- обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
- для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
- письменные задания оформляются увеличенным шрифтом;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

для глухих и слабослышащих:

- лекции оформляются в виде электронного документа;
- письменные задания выполняются на компьютере в письменной форме;
- экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

для лиц с нарушениями опорно-двигательного аппарата:

- лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
- письменные задания выполняются на компьютере со специализированным программным обеспечением;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере;
- используется специальная учебная аудитория для лиц с ЛОВЗ – ауд. 106 главного учебного корпуса по адресу 214013, г. Смоленск, Энергетический пр-д, д.1, здание энергетического института (основной корпус).

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены филиалом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

для слепых и слабовидящих:

- в печатной форме увеличенным шрифтом;
- в форме электронного документа;
- в форме аудиофайла.

для глухих и слабослышащих:

- в печатной форме;
- в форме электронного документа.

для обучающихся с нарушениями опорно-двигательного аппарата:

- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

9. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература.

1. Вояковская, Н. Н. Разработка компиляторов: учебное пособие / Н. Н. Вояковская, А. Е. Москаль, Д. Ю. Булычев, А. А. Терехов. – 2-е изд. – Москва: ИНТУИТ, 2016. – 374 с. – Текст : электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/100452>.

2. Малявко, А. А. Формальные языки и компиляторы: учебное пособие / А. А. Малявко. – Новосибирск : Новосибирский государственный технический университет, 2014. – 431 с. : табл., схем. – (Учебники НГТУ). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=436055>.

3. Пентус, А. Е. Математическая теория формальных языков: учебное пособие / А. Е. Пентус, М. Р. Пентус. – 2-е изд. – Москва : ИНТУИТ, 2016. – 218 с. – ISBN 5-9556-0062-0. – Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100633>.

Дополнительная литература.

1. Златопольский, Д. М. Основы программирования на языке Python / Д. М. Златопольский. – 2-ое изд., испр. и доп. – Москва: ДМК Пресс, 2018. – 396 с. – ISBN 978-5-97060-641-4. – Текст : электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/131683>.

2. Серебряков, В. А. Теория и реализация языков программирования: курс учебное пособие / В. А. Серебряков, М. П. Галочкин, Д. Р. Гончар, М. Г. Фуругян. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2017. – 323 с. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=234669>.

3. Шарден, Б. Крупномасштабное машинное обучение вместе с Python : учебное пособие / Б. Шарден, Л. Массарон, А. Боскетти ; перевод с английского А. В. Логунова. – Москва : ДМК Пресс, 2018. – 358 с. – ISBN 978-5-97060-506-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/105836>.

Список авторских методических разработок.

Федулов Я.А. Комплект мультимедийных презентаций к лекциям по дисциплине «Теория формальных грамматик» (расположен в ЭИОС филиала и передается обучающимся на 1-й лекции для подготовки к лекциям и самостоятельного изучения дисциплины).



ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Но- мер изме- мене- ния	Номера страниц				Всего стра- ниц в доку- менте	Наименование и № документа, вводящего изменения	Подпись, Ф.И.О. внесшего измене- ния в данный эк- земпляр	Дата внесения из- менения в данный эк- земпляр	Дата введения из- менения
	изме- нен- ных	заме- нен- ных	но- вых	анну- лиро- ван- ных					
1	2	3	4	5	6	7	8	9	10