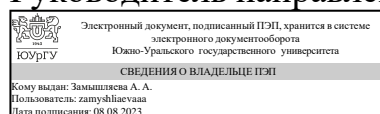


УТВЕРЖДАЮ:
Руководитель направления



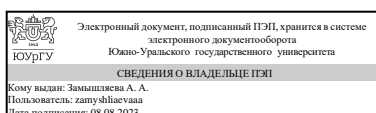
А. А. Замышляева

РАБОЧАЯ ПРОГРАММА

дисциплины ФД.02 Методы трансляции и формальные языки
для направления 01.03.02 Прикладная математика и информатика
уровень Бакалавриат
форма обучения очная
кафедра-разработчик Прикладная математика и программирование

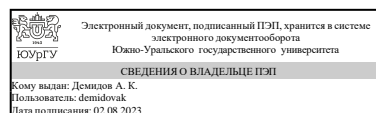
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 01.03.02 Прикладная математика и информатика, утверждённым приказом Минобрнауки от 10.01.2018 № 9

Зав.кафедрой разработчика,
д.физ.-мат.н., проф.



А. А. Замышляева

Разработчик программы,
доцент



А. К. Демидов

1. Цели и задачи дисциплины

Преподаваемая дисциплина является средством решения некоторых прикладных и системных задач. Преподавание и изучение дисциплины следует рассматривать как важную составляющую профессиональной подготовки. Целью дисциплины является обучение студентов применению теории формальных языков и автоматов, методов и алгоритмов лексического и синтаксически управляемого разбора при создании системного и прикладного обеспечения. Задачами дисциплины заключаются в том, чтобы ознакомить студентов с формальными методами описания структуры текстовой информации и синтаксиса языков программирования; дать представление о проблемах и направлениях исследований в области языков программирования; дать практические навыки по использованию современных библиотек и CASE-средств для обработки текстовой информации, для разработки компиляторов и интерпретаторов проблемно-ориентированных языков.

Краткое содержание дисциплины

Теория формальных языков, способы описания, классификация языков и распознавателей. Регулярные языки, использование регулярных выражений для обработки текстовой информации и лексического анализа. Контекстно-свободные языки, их классификация, виды распознавателей для синтаксического анализа. Трансляторы, структура компиляторов и интерпретаторов, этапы трансляции и промежуточные внутренние представления программы, принципы оптимизации кода. Современные CASE-средства, разработка интерпретатора и компилятора учебного языка с использованием LLVM и технологии JIT.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ПК-1 Способен активно участвовать в разработке системного и прикладного программного обеспечения	Знает: модели описания формальных языков, в том числе и языков программирования; задачи и этапы построения трансляторов Умеет: построить грамматику формального языка и преобразовать её к требуемому виду для построения лексического и синтаксического анализаторов Имеет практический опыт: проектирования компиляторов для архитектур семейства Intel

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
1.О.35 Современные технологии разработки программных систем искусственного интеллекта	Не предусмотрены

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Дисциплина	Требования
1.О.35 Современные технологии разработки программных систем искусственного интеллекта	<p>Знает: [УК-1.1. 3-3.] современное состояние информационно-коммуникационных технологий в мире и перспективы их развития, принципы работы современных информационных технологий, [ОПК-1.1. 3-1.] рынок информационных систем и информационно-коммуникационных технологий, автоматизирующих организационно-технические и экономические процессы, [ПК-2.1. 3-1.] основные программные платформы и компоненты систем искусственного интеллекта: механизмы логического вывода (рассуждений), объяснений, приобретения знаний, интеллектуальных интерфейсов, принципы Data Ops и Dev Ops, современные компьютерные технологии разработки программных систем</p> <p>Умеет: использовать современные информационные технологии для решения задач профессиональной деятельности, [ПК-2.1. У-1.] настраивать основные программные платформы и компоненты систем искусственного интеллекта: механизмов логического вывода (рассуждений), объяснений, приобретения знаний, интеллектуальных интерфейсов на особенности проблемной области, участвует в их разработке Имеет практический опыт: применения современных информационных технологий при проектировании систем искусственного интеллекта, участия в разработке прикладного программного обеспечения</p>

4. Объём и виды учебной работы

Общая трудоёмкость дисциплины составляет 4 з.е., 144 ч., 72,5 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах
		Номер семестра
		7
Общая трудоёмкость дисциплины	144	144
<i>Аудиторные занятия:</i>	64	64
Лекции (Л)	32	32
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	16	16
Лабораторные работы (ЛР)	16	16
<i>Самостоятельная работа (СРС)</i>	71,5	71,5
Изучение материала лекций и литературы, подготовка к практическим занятиям	20	20
Установка LLVM, разработка компилятора с языка высокого	38,5	38,5

уровня по вариантам с использованием LLVM		
Подготовка к диф.зачету	9	9
Подготовка к тесту по регулярным выражениям	4	4
Консультации и промежуточная аттестация	8,5	8,5
Вид контроля (зачет, диф.зачет, экзамен)	-	диф.зачет

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Формальные языки и грамматики	6	4	2	0
2	Регулярные языки и лексический анализ	22	8	6	8
3	Контекстно-свободные языки и синтаксический анализ	12	6	4	2
4	Структура компиляторов и интерпретаторов, этапы трансляции, CASE-средства для разработки	24	14	4	6

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Определение транслятора, компилятора и интерпретатора. Универсальные и проблемно-ориентированные языки программирования. Общая схема работы трансляторов. Способы определения языков. Язык как множество цепочек символов. Грамматики. Синтаксис и семантика языка программирования. Классификация грамматик и языков по Хомскому.	2
2	1	Задача распознавания принадлежности цепочки языку. Дерево вывода. Левосторонний и правосторонний выводу. Способы задания грамматик. Проблемы однозначности и эквивалентности грамматик. Классификация распознавателей.	2
3	2	Регулярные языки и грамматики. Свойства регулярных языков. Лемма о разрастании для регулярного языка. Проблемы, разрешимые для регулярных языков. Регулярные и автоматные грамматики. Детерминированные и недетерминированные конечные автоматы. Преобразование конечного автомата к детерминированному виду. Минимизация конечных автоматов.	2
4	2	Регулярные множества и регулярные выражения. Основные свойства, алгебра регулярных выражений. Эквивалентность регулярных грамматик, конечных автоматов и регулярных выражений.	2
5	2	Применение регулярных выражений для обработки текста в Python и C++. Основные библиотечные функции. Особенности реализации регулярных выражений в языках программирования.	2
6	2	Средство для автоматизации построения лексических анализаторов (сканеров) FLEX (RE/flex), назначение и принципы. Автономное использование FLEX для обработки текста.	2
6	3	Контекстно-свободные грамматики. Свойства контекстно-свободных языков. Назначение и принципы работы синтаксических анализаторов. Автоматы с магазинной памятью. Преобразование КС-грамматик. Классификация распознавателей для КС-языков.	2
8	3	Определение LL(k)-грамматики. Множества FIRST и FOLLOW. Построение нисходящего распознавателя для LL(1)-грамматики методом рекурсивного спуска.	2

9	3	Определение LR(k)-грамматики. Построение восходящего распознавателя для LR(1)-грамматики. Алгоритм сдвиг-свертка. Средство для автоматизации построения синтаксических анализаторов (парсеров) BISON.	2
10	4	Основные принципы построения трансляторов. Этапы трансляции. Особенности построения интерпретаторов. Технология JIT. Трансляторы с языка ассемблера. Макроязыки и макрогенерация. Таблицы идентификаторов.	2
11	4	Проблемы интеграции FLEX и BISON. Атрибутные грамматики. Абстрактное синтаксическое дерево (AST), представление и обработка.	2
12	4	CASE-средства для автоматизации разработки трансляторов, построения и обработки AST.	2
13	4	Семантический анализ. Методы генерации и оптимизации кода.	2
14	4	Назначение, основные принципы организации LLVM. Интерфейс LLVM API, основные классы	2
15	4	Промежуточное представление LLVM IR. Выбор целевой машины и генерация машинного кода.	2
16	4	JIT технология в JVM и LLVM. Инструменты LLVM JIT. Выполнение кода в LLVM.	2

5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов
1	1	Разработка грамматики для заданной конструкции языка. Отличия БНФ и РБНФ.	2
2	2	Построение конечного автомата (КА) для по заданной грамматике.	2
3	2	Тест-практика по регулярным выражениям	2
4	2	Использование FLEX для обработки текста. Порядок правил, выделение подавтоматов.	2
5	3	Разработка распознавателя для LL(1)-грамматик. Преобразование грамматики. Определение множеств FIRST и и FOLLOW.	2
6	3	Определение правил грамматики в BISON	2
7	4	Установка LLVM. Подключение LLVM к транслятору. Генерация кода для процессоров семейства Intel, выбор архитектуры.	2
8	4	Разработка интерпретатора в LLVM с использованием технологии JIT. Сравнение скорости работы с обычным интерпретатором.	2

5.3. Лабораторные работы

№ занятия	№ раздела	Наименование или краткое содержание лабораторной работы	Кол-во часов
1	2	Симуляция КА в Umlet. Генерация класса по диаграмме состояний.	2
2	2	Преобразование недетерминированного КА к детерминированному. Минимизация конечных автоматов.	2
3	2	Применение расширенных регулярных выражений в C++ и Python для обработки текста	2
4	2	Решение задач по обработке текста на FLEX.	2
5	3	Решение задач по синтаксическому анализу в BISON	2
6	4	Определение сканера для учебного языка. Проверка сканера.	2
7	4	Определение парсера для учебного языка. Проверка парсера.	2

8	4	Определение и обработка AST для учебного языка. Проверка интерпретатора.	2
---	---	--	---

5.4. Самостоятельная работа студента

Выполнение СРС			
Подвид СРС	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов
Изучение материала лекций и литературы, подготовка к практическим занятиям	ЭУМД, 2, с.10-340; ЭУМД, 3, разделы 1-6	7	20
Установка LLVM, разработка компилятора с языка высокого уровня по вариантам с использованием LLVM	ЭУМД, 1, главы 1,3,5,6,7	7	38,5
Подготовка к диф.зачету	ЭУМД, 2, с.10-340; ЭУМД, 1, гл. 3; ПУМД, доп.лит, 1, гл. 1-5	7	9
Подготовка к тесту по регулярным выражениям	ЭУМД, 4, раздел 1	7	4

6. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

6.1. Контрольные мероприятия (КМ)

№ КМ	Се-местр	Вид контроля	Название контрольного мероприятия	Вес	Макс. балл	Порядок начисления баллов	Учитывается в ПА
1	7	Текущий контроль	P1. Формальные языки и грамматики	10	10	Задание выполнено вовремя - 2 балла, иначе 0 баллов Учтены все варианты синтаксиса в правилах - 3 балла, оценка снижается на 1 балл за каждый отсутствующий вариант Нет синтаксических ошибок в формате правил и логических ошибок - 5 баллов, оценка снижается на 1 балл за каждую ошибку	дифференцированный зачет
2	7	Текущий контроль	P2. Конечные автоматы для распознавания цепочек регулярного	10	10	Выполнена проверка КА путем симуляции в UMLet- 3 балла, иначе 0 баллов	дифференцированный зачет

			языка			Выполнена генерация класса для КА по диаграмме состояний - 2 балла, иначе 0 баллов Нет синтаксических и логических ошибок в схеме КА - 5 баллов, оценка снижается на 1 балл за каждую ошибку	
3	7	Текущий контроль	P3 Регулярные выражения	10	10	Решить 5 задач на C++ или Python с использованием функций библиотеки для регулярных выражений По 2 баллу за каждую решенную задачу, за частично правильное решение 1 балл, иначе 0 баллов	дифференцированный зачет
4	7	Текущий контроль	Тест по регулярным выражениям	10	10	Тест включает 15 вопросов разной сложности, для выполнения задания необходимо ответить правильно на не менее 10 вопросов. По 1 баллу за правильный ответ на вопрос теста Баллы свыше 10 за правильные ответы переносятся в бонусы	дифференцированный зачет
5	7	Промежуточная аттестация	P4 Лексический анализ	-	10	Решить 5 задач с помощью инструмента для лексического анализа FLEX По 2 баллу за каждую решенную задачу, за частично правильное решение 1 балл, иначе 0 баллов	дифференцированный зачет
6	7	Текущий контроль	P5. Синтаксический анализ	10	10	Решить 5 задач с помощью инструмента для синтаксического анализа BISON По 2 баллу за каждую решенную задачу, за частично правильное решение 1 балл, иначе 0 баллов	дифференцированный зачет
7	7	Текущий контроль	P6. Разработка интерпретатора	20	10	Решена подзадача 1 (лексический анализ) -	дифференцированный зачет

						<p>3 балла, оценка снижается на 1 балл за каждую ошибку</p> <p>Решена подзадача 2 (синтаксический анализ) - 3 балла, оценка снижается на 1 балл за каждую ошибку</p> <p>Решена подзадача 3 (построение AST, обход AST и интерпретация) - 4 балла, оценка снижается на 1 балл за каждую ошибку</p>	
8	7	Текущий контроль	Р7. Разработка компилятора на LLVM	10	10	<p>Выполнена установка библиотек LLVM и подключение к компилятору - 2 балла, иначе 0 баллов</p> <p>Выполнена генерация кода для процессора Intel в компиляторе для учебного языка - 4 балла, оценка снижается на 1 балл за каждую ошибку</p> <p>Выполнено тестирование компилятора - 2 балла, иначе 0 баллов</p> <p>Выполнено сравнение скорости программы на учебном языке и на языке C - 2 балла, иначе 0 баллов</p>	дифференцированный зачет
9	7	Текущий контроль	Р8. Использование JIT-технологии	10	10	<p>Выполнена генерация внутреннего представления кода для JIT интерпретатора учебного языка - 4 балла, оценка снижается на 1 балл за каждую ошибку</p> <p>Выполнено тестирование интерпретатора - 3 балла, иначе 0 баллов</p> <p>Выполнено сравнение скорости работы скомпилированного кода, простого и JIT интерпретатора - 3 балла, если выполнено</p>	дифференцированный зачет

						сравнение только двух вариантов, то 2 балла, иначе 0 баллов	
10	7	Промежуточная аттестация	дифференцированный зачет	-	10	<p>Это контрольное мероприятие проводится в форме собеседования. Задаются два вопроса по пройденным темам. В первую очередь предлагаются вопросы по темам, которые были оценены на "неудовлетворительно" по текущему контролю. Каждый ответ оценивается от 0 до 5 баллов в зависимости от полноты ответа, знания терминов. Шкала оценивания Полный, правильный ответ - 5 баллов Одна неточность, неправильный термин - 4 балла Частичный ответ - 3 балла В ответе есть некоторые правильные определения - 2 балла Нет ответа - 0 баллов Оценка ставится как сумма баллов за оба ответа</p>	дифференцированный зачет
11	7	Бонус	Посещаемость и активность на занятиях	-	15	<p>Критерий Посещение занятий 100% - 3 балла 85% - 2 балла менее 85% - 0 баллов</p> <p>Критерий Активность на занятиях Правильный ответ у доски - 1 балл Помощь с места - 0,2 балла Не более 3 баллов в сумме</p> <p>Критерий Выполнено расширенное задание Р6 (часть I) - 5 баллов Критерий Дополнительные</p>	дифференцированный зачет

						баллы за тест (11+) - 1-5 баллов	
--	--	--	--	--	--	----------------------------------	--

6.2. Процедура проведения, критерии оценивания

Вид промежуточной аттестации	Процедура проведения	Критерии оценивания
дифференцированный зачет	Возможно определение рейтинга студента по дисциплине по результатам текущего контроля в соответствии с п.2.6. Если студент не набрал необходимый рейтинг по текущему контролю, то проводится устное собеседование. Студент должен ответить на 2 вопроса из вопросов к диф.зачету. В первую очередь предлагаются вопросы по темам, которые были оценены на "неудовлетворительно" по текущему контролю.	В соответствии с пп. 2.5, 2.6 Положения

6.3. Паспорт фонда оценочных средств

Компетенции	Результаты обучения	№ КМ											
		1	2	3	4	5	6	7	8	9	10	11	
ПК-1	Знает: модели описания формальных языков, в том числе и языков программирования; задачи и этапы построения трансляторов	+	+	+	+	+	+	+	+	+	+	+	+
ПК-1	Умеет: построить грамматику формального языка и преобразовать её к требуемому виду для построения лексического и синтаксического анализаторов	+	+	+	+	+	+				+	+	
ПК-1	Имеет практический опыт: проектирования компиляторов для архитектур семейства Intel										+	+	+

Типовые контрольные задания по каждому мероприятию находятся в приложениях.

7. Учебно-методическое и информационное обеспечение дисциплины

Печатная учебно-методическая документация

а) *основная литература:*

Не предусмотрена

б) *дополнительная литература:*

- Гордеев, А. В. Системное программное обеспечение Учеб. А. В. Гордеев, А. Ю. Молчанов. - СПб. и др.: Питер, 2001. - 708 с. ил.

в) *отечественные и зарубежные журналы по дисциплине, имеющиеся в библиотеке:*

Не предусмотрены

г) *методические указания для студентов по освоению дисциплины:*

- Конспект лекций и методические указания для выполнения СРС

из них: учебно-методическое обеспечение самостоятельной работы студента:

- Конспект лекций и методические указания для выполнения СРС

Электронная учебно-методическая документация

№	Вид литературы	Наименование ресурса в электронной форме	Библиографическое описание
1	Основная литература	Электронно-библиотечная система издательства Лань	Бруно, К. Л. LLVM: инфраструктура для разработки компиляторов / К. Л. Бруно, А. Рафаэль. — Москва : ДМК Пресс, 2015. — 342 с. — ISBN 978-5-97060-305-5. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/90119
2	Основная литература	Электронно-библиотечная система издательства Лань	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. https://e.lanbook.com/book/100452
3	Основная литература	Учебно-методические материалы кафедры	Разработка компиляторов и интерпретаторов — Текст : электронный https://ipc.susu.ru/38277.html
4	Основная литература	Учебно-методические материалы кафедры	Регулярные выражения — Текст : электронный https://ipc.susu.ru/28181.html
5	Дополнительная литература	Электронно-библиотечная система издательства Лань	Вирт, Н. Построение компиляторов / Н. Вирт. — Москва : ДМК Пресс, 2010. — 192 с. — ISBN 978-5-94074-585-3. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/1262

Перечень используемого программного обеспечения:

1. -MinIDE (сборка из SciTE, MinGW C/C++, GDB)(бессрочно)

Перечень используемых профессиональных баз данных и информационных справочных систем:

Нет

8. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Практические занятия и семинары	333 (36)	компьютеры, проектор
Лабораторные занятия	333 (36)	компьютеры с ПО для разработки компиляторов