ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол №2

Рабочая программа дисциплины (модуля)
«Дизайн компиляторов»

Направление подготовки: *02.03.01 Математика и компьютерные науки*

Направленность (профиль) подготовки: *Математика и компьютерные науки*

Квалификация (степень) выпускника: *бакалавр*

Форма обучения: *очная*

Срок освоения программы: *4 года*

Год набора: *2026*

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0215B0C000D1B3FEB44A37EE631E0C27D4
Владелец: Ивашкевич Евгений Васильевич
Действителен: с 13.01.2026 до 13.04.2027

Электронный документ

Москва
2026

Содержание

1. **Краткая характеристика дисциплины (модуля)** 3
2. **Перечень планируемых результатов обучения** 4
3. **Тематический план** 6
4. **Содержание дисциплины (модуля)** 6
5. **Учебно-методическое обеспечение** 7
6. **Материально-техническое обеспечение** 7
7. **Методические и оценочные материалы** 9

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Дизайн компиляторов» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению 02.03.01 «Математика и компьютерные науки», профиль «Математика и компьютерные науки», утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Дизайн компиляторов» важно для понимания принципов преобразования высокоуровневого кода в эффективный машинный код, что способствует оптимизации работы программ и систем. Это знание позволяет создавать новые языки программирования и инструменты разработки, повышая производительность и надежность программного обеспечения.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 «Математика и компьютерные науки», профиль «Математика и компьютерные науки» и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений, как дисциплина по выбору.

Дисциплина (модуль) является выборной и доступна для изучения на 3 или 4 курсе в 5, 6, 7, 8 семестрах на выбор после успешного изучения одной из дисциплин (модулей): «Разработка на Rust» или «Разработка на C++», или «Разработка на Java», «Теория формальных языков».

Цель изучения дисциплины (модуля): освоение методов и принципов разработки компиляторов для преобразования и оптимизации программного кода.

Задачи изучения дисциплины (модуля):

— изучить основные угрозы и уязвимости веб-приложений, включая OWASP Top 10 (такие как инъекции, XSS и CSRF), а также механизмы их эксплуатации и выявления с использованием инструментов анализа;

— практические методы защиты, включая аутентификацию, авторизацию, шифрование данных, безопасную разработку кода и конфигурацию серверов, с акцентом на фреймворки вроде Django, Spring или React;

— развить навыки проведения пентестов, аудита безопасности и реагирования на инциденты, включая создание политик безопасности и интеграцию инструментов мониторинга для минимизации рисков в реальных проектах.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

— принципы компиляции кода и устройства компиляторов;

— техники оптимизации сгенерированного кода;

— алгоритмы вывода типов для динамических языков программирования;

уметь:

— описывать формальную грамматику языка;

— использовать современные инструменты для генерации кода компилятора;

— анализировать внутреннее представление программ;

— выполнять преобразования внутреннего представления программ;

владеть:

— навыками реализации статического анализатор кода;

— практикой автоматического перевода кода из одного языка программирования в другой.

2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области разработки, основные принципы критической оценки источников информации и их релевантности
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-1.	Способен консультировать и использовать фундаментальные знания в области математического анализа, комплексного и функционального анализа алгебры, аналитической геометрии, дифференциальной геометрии и топологии, дифференциальных уравнений, дискретной математики и математической логики, теории вероятностей, математической статистики и случайных процессов, численных методов, теоретической механики в профессиональной деятельности	ОПК-1.1.	Знает основные концепции и теории в области математического анализа и смежных дисциплин; методы и подходы, используемые в различных областях математики
		ОПК-1.2.	Умеет применять математические методы для решения профессиональных задач
		ОПК-1.3.	Имеет практический опыт разработки и реализация математических моделей в профессиональной деятельности
ОПК-4.	Способен находить, анализировать, реализовывать программно и	ОПК-4.1.	Знает базовые основы современного математического аппарата,

	использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем		связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
		ОПК-4.2.	Умеет использовать этот математический аппарат в профессиональной деятельности
		ОПК-4.3.	Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ОПК-6.	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований
ПК-3.	Способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет разрабатывать и применять математические модели и алгоритмы для решения различных задач, анализируя полученные результаты
		ПК-3.3.	Имеет практический опыт использования методов математического и алгоритмического моделирования в реальных проектах или исследованиях

3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
		Лекции	Семинары (практические занятия)			
1	Формальные грамматики и описание структуры языков	6	6		24	Контрольная работа
2	Инструменты разбора грамматик и абстрактные синтаксические деревья (AST)	4	4		16	Контрольная работа
3	Построение и анализ потоков данных (DFG)	6	6		24	Контрольная работа
4	Статическая и динамическая типизация, алгоритмы вывода типов	6	6	2	24	Контрольная работа
5	Преобразования AST и генерация кода	8	8	2	36	Проект
	Зачет с оценкой			2		
	Итого:	***30***	***30***	***6***	***124***	
Объем дисциплины (модуля) (в ак. ч.)		***190***				
Объем дисциплины (модуля) (в зач. ед.)		***5***				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Формальные грамматики и описание структуры языков	Введение в компиляцию; регулярные и КС-грамматики BNF / EBNF, лексическая структура, конфликтные грамматики LL- vs LR-подходы; разбор простых грамматик вручную
2	Инструменты разбора грамматик и абстрактные синтаксические деревья (AST)	Лексер/парсер-генераторы, построение AST Обработка ошибок парсера, визуализация AST AST
3	Построение и анализ потоков данных (DFG)	Промежуточные представления, CFG, базовые DFG SSA-форма, анализ доступности и достижимости Оптимизации по данным: константная свёртка, устранение мёртвого кода
4	Статическая и динамическая типизация, алгоритмы вывода типов	Статические типовые системы, алгоритмы вывода типов Hindley–Milner Динамическая / гибридная типизация, проверки во время исполнения Практика вывода типов и аннотаций в IR.
5	Преобразования AST и генерация кода	Преобразования AST для упрощения структуры кода Преобразования AST для оптимизации кода Преобразование AST в текст на другом языке или в бинарный код

5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

Основная литература:

1. Разработка компиляторов : краткий курс / Н. Н. Вояковская, А. Е. Москаль, Д. Ю. Булычев [и др.] ; - Москва : ИНТУИТ, 2016. - 276 с. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2157793

2. Малявко, А. А. Формальные языки и компиляторы / А. А. Малявко. - Новосибирск : НГТУ, 2014. - 431 с. - SBN 978-5-7782-2318-9. - ISBN 978-5-7782-2318-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/548152

Дополнительная литература:

1. Вирт, Н. Построение компиляторов : практическое пособие / Н. Вирт ; пер. с англ. Е. В. Борисова, Л. Н. Чернышова. - 2-е изд. - Москва : ДМК Пресс, 2023. - 193 с. - ISBN 978-5-89818-573-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/2107934 .

6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование"	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное
Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
Пакеты программных средств и библиотек:		

AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

7. Методические и оценочные материалы

Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Дизайн компиляторов» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, контрольные работы, проект, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

Лекция – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удается, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

Семинар — это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

Контрольная работа – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы – получить специальные знания по одной или нескольким темам дисциплины и продемонстрировать навыки их практического применения.

Проект – исследовательская работа по курсу и презентация результатов.

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

Самостоятельная работа – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов, планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

Система оценивания результатов обучения по дисциплине (модулю)

Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Дизайн компиляторов»

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

Промежуточная аттестация по дисциплине (модулю) осуществляется в форме ***зачета с оценкой***, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом, если иное не указано в силлабусе дисциплины:

Десятибалльная оценка	**Пятибалльная оценка**	**Общая характеристика результата обучения по дисциплине (модулю)**
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
9	Отлично	
8	Отлично	
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно
6	Хорошо	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Дизайн компиляторов» оценивается следующим образом:

Активность	Вес	Количество	Описание
Контрольная работа	30%	3	Письменная работа с набором задач, которые нужно решить за ограниченное время
Проект	40%	1	Исследовательская работа по дисциплине (модулю) и презентация результатов
Зачет с оценкой	30%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

Формула расчёта итоговой оценки по дисциплине (модулю) «Дизайн компиляторов»: «0,3 ×контрольная работа + 0,4 × проект+ 0,3 × зачёт с оценкой».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные задания для контрольной работы

1. **Теория:** Определите разницу между регулярными грамматиками и контекстно-свободными (КС-грамматиками). Приведите пример регулярной грамматики для языка идентификаторов (буква, за которой следуют буквы/цифры).

2. **Практика:** Запишите КС-грамматику в BNF для арифметических выражений с операциями +, *, скобками и переменными. Преобразуйте её в EBNF, устранив левостороннюю рекурсию.

3. **Анализ:** Для грамматики S → A | B, A → aA | ε, B → bB | ε вычислите множества FIRST и FOLLOW. Объясните, почему она подходит для LL(1)-разбора.

4. **Сравнение:** Опишите ключевые различия между LL- и LR-подходами к разбору. Приведите пример грамматики, где LL(1) терпит неудачу, но LR(1) succeeds (например, неоднозначные выражения).

5. **Теория:** Что такое лексер-генератор (например, Flex) и парсер-генератор (например, Bison)? Опишите этапы их работы на примере простого токенизатора для ключевых слов и чисел.

6. **Практика:** Для входной строки "if (x > 0) { y = 1; }" постройте AST вручную: укажите узлы (IfNode, BinaryOpNode и т.д.) и их связи. Объясните, как AST отличается от конкретного синтаксического дерева (CST).

7. **Обработка ошибок:** Как парсер ANTLR обрабатывает синтаксические ошибки (например, пропуск токена)? Приведите пример recovery-стратегии для незакрытой скобки в выражении.

8. **Визуализация:** Опишите, как визуализировать AST с помощью Graphviz. Наришите схематично AST для цикла "for i in 1..10 { print(i); }".

9. **Теория:** Что такое CFG (Control Flow Graph) и DFG (Data Flow Graph)? Объясните, как CFG используется для анализа потоков в процедуре с ветвлениями и циклами.

10. **Практика:** Для кода на псевдокоде:

1: a = 5;
2: if (a > 0) b = a + 1; else b = 0;
3: c = b * 2;

Постройте CFG и DFG. Вычислите доступные определения (reaching definitions) для переменной b в узле 3.

11. **Анализ:** Опишите SSA-форму (Static Single Assignment). Преобразуйте приведенный выше код в SSA, введя φ-функции для слияния путей.

12. **Оптимизации:** Как анализ живости переменных (liveness analysis) помогает в устранении мертвого кода? Приведите пример: если переменная x присваивается, но не используется после, покажите оптимизацию.

13. **Теория:** Сравните статическую и динамическую типизацию. Приведите преимущества статической (например, в Java) и динамической (например, в Python) типизации для компиляторов.

14. **Практика:** Используя алгоритм Hindley–Milner, выведите тип для λ-выражения: let id = λx. x in id (id true). Укажите полиморфный тип (forall α. α → α).

15. **Гибридная типизация:** Объясните, как гибридная типизация (статическая + динамическая) работает в языках вроде TypeScript. Приведите пример аннотации типа и runtime-проверки.

16. **IR-практика:** В промежуточном представлении (IR) для "let y = x + 1" добавьте типовые аннотации. Что происходит при выводе типов, если x — int, a + — перегрузка?

17. **Теория:** Что такое десугаринг (desugaring)? Объясните, как он упрощает AST, превращая сахар (syntactic sugar) в базовые конструкции (например, for-loop в while).

18. **Практика:** Для AST выражения "if (cond) { a = b + c; } else { a = 0; }" выполните инлайнинг, если b и c — константы. Покажите преобразованный AST.

19. **Оптимизации:** Опишите частичную оценку (partial evaluation). Примените её к коду "f(x) = if (true) x*2 else x/2; call f(5)", вычислив статически.

20. **Теория и практика:** Опишите регистровое распределение (register allocation) в генерации кода. Для AST простого выражения "a = b + c" сгенерируйте машинный код (x86-подобный), предполагая регистры EAX, EBX, и укажите, как граф интерференции помогает распределению.

Примерное описание и критерии оценивания к проекту

Описание проекта:

В рамках итогового проекта студентам предлагается разработать прототип компилятора для простого императивного или функционального языка программирования, включающий следующие ключевые этапы:

1. **Определение формальной грамматики языка**
   — Описание синтаксиса с использованием контекстно-свободной грамматики (КС-грамматики).
   — Формализация правил синтаксиса и допустимых конструкций.
2. **Построение парсера и синтаксического анализа**
   — Реализация парсера с применением методов нисходящего (LL) или восходящего (LR) разбора.
   — Построение абстрактного синтаксического дерева (AST).
3. **Анализ потоков данных (DFG)**
   — Построение графа потоков данных на основе AST.
   — Анализ и оптимизация промежуточного представления.
4. **Типизация и вывод типов**
   — Внедрение статической типизации с использованием алгоритма вывода типов (например, Хиндли-Милнера).
   — Проверка корректности типов в программе.
5. **Преобразования AST**
   — Реализация локальных и глобальных преобразований AST для оптимизации и упрощения.
   — Удаление избыточных конструкций и применение правил преобразования.
6. **Генерация кода**
   — Генерация промежуточного или целевого кода с учётом архитектурных особенностей.
   — Обеспечение корректности и эффективности сгенерированного кода (управление регистрами, памятью и потоками).

Цели проекта:

— Продемонстрировать понимание и применение базовых концепций формальных грамматик и синтаксического анализа.
— Использовать инструменты и методы построения парсеров и AST.
— Реализовать анализ потоков данных для оптимизации кода.
— Применить алгоритмы вывода типов для обеспечения безопасности типов.
— Провести преобразования AST для улучшения качества промежуточного представления.
— Сгенерировать корректный и работоспособный код.

Критерии оценки:

— Корректность и полнота описания формальной грамматики языка.
— Правильная реализация парсера и построение AST без ошибок.
— Качество построения и анализа графа потоков данных (DFG), применение оптимизаций.
— Реализация статической типизации и корректный вывод типов.
— Эффективность и корректность преобразований AST.
— Корректность, эффективность и архитектурная адаптация генерируемого кода.
— Качество документации и презентации проекта, ясность пояснений и демонстраций.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1	Какой инструмент позволяет автоматически генерировать лексер на	Flex / lex	УК-1

	основе регулярных выражений в процессе разработки компилятора?		
2	Какой метод формального описания синтаксиса языка используется при проектировании парсера?	BNF / формальная грамматика	УК-1
3	Какой системный подход применяется для разрешения конфликтов в LR-парсерах (shift/reduce, reduce/reduce)?	Conflict resolution / разрешение конфликтов	УК-1
4	Какая математическая модель лежит в основе распознавания лексем в потоке символов?	Конечный автомат / finite automaton	ОПК-1
5	Какой тип логического вывода лежит в основе работы LL-парсеров при предсказании правила вывода?	Дедукция / deduction	ОПК-1
6	Какая структура данных используется для представления синтаксического дерева программы после разбора?	Дерево / AST	ОПК-1
7	Какой математический аппарат применяется для моделирования потока управления в программе?	Граф / control flow graph	ОПК-4
8	Какой алгоритм используется для построения промежуточного представления (IR) на основе синтаксического дерева?	Генерация IR / intermediate representation	ОПК-4
9	Какой метод оптимизации позволяет заменить выражение вида **2 + 3** на константу **5** на этапе компиляции?	Constant folding / константная свёртка	ОПК-4
10	Какой алгоритм удаляет код, который не влияет на результат выполнения программы?	Dead code elimination / удаление мёртвого кода	ОПК-4
11	Какой способ представления переменных в IR гарантирует, что каждая переменная присваивается только один раз?	SSA / static single assignment	ОПК-6
12	Какой алгоритм применяется для упрощения синтаксического сахара (например, циклов **for** в **while**)?	Desugaring / десугаринг	ОПК-6
13	Какой метод позволяет визуализировать структуру программы в виде дерева для отладки и анализа?	AST-диаграмма / графическое представление дерева	ОПК-6
14	Какой подход используется для генерации объектного кода из промежуточного представления?	Code generation / генерация кода	ОПК-6
15	Какой метод математического моделирования позволяет автоматически выводить типы переменных в языках с полиморфизмом?	Hindley–Milner / вывод типов	ПК-3
16	Какой алгоритм позволяет частично вычислить программу на этапе компиляции, упрощая её выполнение?	Partial evaluation / частичная оценка	ПК-3
17	Какой метод применяется для проверки корректности типов в программе на этапе компиляции?	Type checking / проверка типов	ПК-3
18	Какой способ анализа позволяет определить, какие переменные "живы" в определённой точке программы?	Liveness analysis / анализ живости	ПК-3

19	Какой системный подход используется для синтеза информации из нескольких источников при проектировании компилятора?	Системный анализ и синтез информации	УК-1
20	Какой математический метод применяется для оптимизации циклов на основе анализа индексов и границ?	Loop optimization / оптимизация циклов	ПК-3