

**УТВЕРЖДЕНА**

Решением Ученого совета  
АНО ВО «Центральный университет»  
«07» марта 2024 г.  
Протокол №1

**Рабочая программа дисциплины (модуля)  
«Дизайн компиляторов»**

**Направление подготовки:** 38.03.05 Бизнес-информатика

**Направленность (профиль) подготовки:** Бизнес-аналитика

**Квалификация (степень) выпускника:** бакалавр

**Форма обучения:** очная

**Срок освоения программы:** 4 года

**Год набора:** 2024

**Москва  
2024**

## Содержание

<b>1. Краткая характеристика дисциплины (модуля)</b> .....	3
<b>2. Перечень планируемых результатов обучения</b> .....	4
<b>3. Тематический план</b> .....	6
<b>4. Содержание дисциплины (модуля)</b> .....	7
<b>5. Учебно-методическое обеспечение</b> .....	8
<b>6. Материально-техническое обеспечение</b> .....	8
<b>7. Методические и оценочные материалы</b> .....	10

## 1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Дизайн компиляторов» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 38.03.05 Бизнес-информатика, профиль Бизнес-аналитика, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 838 от 29.07.2020 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 38.03.05 Бизнес-информатика, профиль Бизнес-аналитика и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений.

Дисциплина (модуль) является выборной и доступна для изучения на 3 или 4 курсе в 5, 6, 7, 8 семестрах на выбор.

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

**Задачи изучения дисциплины (модуля)** направлены на формирование у студентов следующий знаний, умений и навыков:

- знание принципов компиляции кода и устройства компиляторов;
- знание техник оптимизации сгенерированного кода;
- знание алгоритмов вывода типов для динамических языков программирования;
- умение описывать формальную грамматику языка;
- умение использовать современные инструменты для генерации кода компилятора;
- умение анализировать внутреннее представление программ;
- умение выполнять преобразования внутреннего представления программ;
- навык реализации статического анализатора кода;
- навык реализации 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-6.	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	УК-6.1.	Знает основные принципы самовоспитания и самообразования, профессионального и личностного развития, исходя из этапов карьерного роста и требований рынка труда
		УК-6.2.	Умеет планировать свое рабочее время и время для саморазвития. формулировать цели личностного и профессионального развития и условия их достижения, исходя из тенденций развития области профессиональной деятельности, индивидуально-личностных особенностей
		УК-6.3.	Имеет практический опыт получения дополнительного образования, изучения дополнительных образовательных программ
ОПК-2.	Способен проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом	ОПК-2.1.	Знает основные тенденции и характеристики рынка информационных систем и информационно-коммуникационных технологий
		ОПК-2.2.	Умеет проводить исследование и анализ рыночной информации для оценки потребностей бизнеса и выбора оптимальных решений
		ОПК-2.3.	Имеет практический опыт в разработке и внедрении стратегий управления бизнесом на основе анализа рынка информационных технологий
ПК-2.	Способен использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования для решения задач профессиональной деятельности	ПК-2.1.	Знает основные математические методы и инструментальные средства, применяемые для обработки и анализа информации
		ПК-2.2.	Умеет эффективно использовать математический аппарат для систематизации данных и решения профессиональных задач

		ПК-2.3.	Имеет практический опыт работы с инструментами анализа информации в рамках исследовательских проектов
ПК-3.	Способен готовить научно-технические отчеты, презентации, научные публикации по результатам выполненных исследований	ПК-3.1.	Знает требования и стандарты оформления научно-технических отчетов, презентаций и публикаций
		ПК-3.2.	Умеет структурировать и представлять результаты исследований в ясной и доступной форме
		ПК-3.3.	Имеет практический опыт подготовки и публикации научных материалов, отражающих результаты выполненных исследований
ПК-8.	Способен под руководством специалиста более высокой категории осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-8.1.	Знает принципы и стандарты управления проектами
		ПК-8.2.	Умеет разрабатывать планы и организовывать проектную деятельность в соответствии с установленными стандартами

### 3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		<i>Очная форма</i>				
		Контактная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Формальные грамматики и описание структуры языков	5	5		22	Подготовка к семинару, Домашние задания, Контрольная работа
2	Инструменты разбора грамматик и абстрактные синтаксические деревья (AST)	6	6		23	Подготовка к семинару, Домашние задания, Контрольная работа
3	Построение и анализ потоков данных (DFG)	3	3		22	Подготовка к семинару, Домашние задания
4	Статическая и динамическая типизация, алгоритмы вывода типов	5	5		23	Подготовка к семинару, Домашние задания, Контрольная работа
5	Преобразования AST	4	4		22	Подготовка к семинару, Домашние задания
6	Генерация кода из AST	3	3		22	Подготовка к семинару, Домашние задания
	<i>Зачет с оценкой</i>			4		Проект
	<b><i>Итого:</i></b>	<b>26</b>	<b>26</b>	<b>4</b>	<b>134</b>	
	<b><i>Объем дисциплины (в ак. ч.)</i></b>	<b>190</b>				
	<b><i>Объем дисциплины (в зач. ед.)</i></b>	<b>5</b>				

#### 4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Формальные грамматики и описание структуры языков	Правила синтаксиса и допустимых конструкций языка. Контекстно-свободные грамматики. Производственные правила. Иерархия Хомского. Основа для построения парсеров и синтаксического анализа.
2	Инструменты разбора грамматик и абстрактные синтаксические деревья (AST)	Парсеры. Методы разбора: нисходящий (LL) и восходящий (LR) парсинг. Абстрактное синтаксическое дерево. AST — основа для анализа, оптимизации и генерации кода. Инструменты генерации парсеров (ANTLR, yacc).
3	Построение и анализ потоков данных (DFG)	Потоки данных. Анализ DFG. Использование DFG. Графы потоков данных. Ключевой этап оптимизации промежуточного представления.
4	Статическая и динамическая типизация, алгоритмы вывода типов	Статическая типизация. Динамическая типизация. Алгоритмы вывода типов (Хиндли-Милнер). Типизация. Комбинация статической типизации и вывода типов.
5	Преобразования AST	Преобразования AST. Удаление избыточных конструкций. Локальные и глобальные преобразования. Использование шаблонов и правил. Преобразования.
6	Генерация кода из AST	Генерация кода. Учет архитектурных особенностей. Инструкции управления памятью, регистрами и потоками. Эффективность и корректность исполняемой программы. Промежуточные представления.

## 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. Малявко А. А. Формальные языки и компиляторы : учебник для вузов / А. А. Малявко. — Москва : Издательство Юрайт, 2025. — 429 с. — (Высшее образование). — ISBN 978-5-534-04288-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562822>.

### *Дополнительная литература:*

1. Новожилов О. П. Архитектура ЭВМ и вычислительных систем : учебник для вузов / О. П. Новожилов. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 505 с. — (Высшее образование). — ISBN 978-5-534-20365-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/568920>.

2. Толстобров А. П. Архитектура ЭВМ : учебник для вузов / А. П. Толстобров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 162 с. — (Высшее образование). — ISBN 978-5-534-16839-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566711>.

## 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;
- 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 к 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	<a href="https://elibrary.ru/defaultx.asp">https://elibrary.ru/defaultx.asp</a>
2.	База данных для IT-специалистов	<a href="https://habr.com">https://habr.com</a>
3.	База данных ScienceDirect	<a href="https://www.sciencedirect.com">https://www.sciencedirect.com</a>
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	<a href="https://minobrnauki.gov.ru/">https://minobrnauki.gov.ru/</a>
5.	Федеральный портал «Российское образование»	<a href="https://www.edu.ru/">https://www.edu.ru/</a>
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	<a href="http://window.edu.ru/">http://window.edu.ru/</a>
7.	Единая коллекция цифровых образовательных ресурсов	<a href="http://school-collection.edu.ru/">http://school-collection.edu.ru/</a>
8.	Федеральный центр информационно - образовательных ресурсов	<a href="http://fcior.edu.ru/">http://fcior.edu.ru/</a>

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
<b>Операционные системы:</b>		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
<b>Браузеры:</b>		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
<b>Офисные приложения:</b>		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
<b>Программное обеспечение для планирования и учета времени:</b>		
Toggle app	зарубежное	свободно распространяемое
<b>Системы управления проектами:</b>		
Microsoft Imagine (Project)	зарубежное	лицензионное
<b>Системы управления базами данных:</b>		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
<b>Системы резервного копирования (backup):</b>		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
<b>Справочно-правовые системы:</b>		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
<b>Средства антивирусной защиты:</b>		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
<b>Среды разработки:</b>		
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
<b>Пакеты программных средств и библиотек:</b>		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
<b>Системы управления библиографической информацией:</b>		
Zotero	зарубежное	свободно распространяемое
<b>Сервисы и службы:</b>		
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, домашние задания, 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 удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

*Участие в семинаре (аудиторная работа)* – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

*Домашнее задание* – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

*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 (модуля) 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:

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Зачтено	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	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			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 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	Зачтено	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	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

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

Активность	Вес	Количество	Описание
Домашние задания	30%	13	Набор задач по темам недели
Аудиторная работа	10%	13	Активная работа студента на семинаре
Контрольные работы	20%	3	Письменная работа с набором задач, которые нужно решить за ограниченное время
Зачет с оценкой	40%	1	Защита итогового проекта

**Формула расчёта итоговой оценки по дисциплине (модулю) «Дизайн компиляторов»:** « $0,3 \times$  среднее за домашние задания +  $0,1 \times$  аудиторная работа +  $0,2 \times$  среднее за контрольные работы +  $0,4 \times$  зачет с оценкой».

## Текущий контроль успеваемости обучающихся по дисциплине (модулю)

### Примерные домашние задания

#### Домашнее задание по теме «Формальные грамматики и описание структуры языков»

1. Определите формальную грамматику (в форме Бэкуса-Наура, BNF) для простого языка арифметических выражений с операциями сложения, вычитания, умножения и деления, включая скобки. Приведите примеры корректных и некорректных строк.
2. Для заданной грамматики найдите множество FIRST и FOLLOW для всех нетерминальных символов. Объясните, как эти множества используются при построении парсеров.
3. Приведите пример грамматики с левой рекурсией. Выполните преобразование этой грамматики в эквивалентную без левой рекурсии.
4. Опишите, чем отличаются контекстно-свободные грамматики от регулярных. Приведите по одному примеру языка, который можно описать только с помощью контекстно-свободной грамматики.
5. Постройте конечный автомат, распознающий язык, описанный регулярным выражением  $(a|b)^*abb$ . Объясните соответствие между регулярными выражениями и конечными автоматами.

#### Домашнее задание по теме «Инструменты разбора грамматик и абстрактные синтаксические деревья (AST)»

1. Используя инструмент ANTLR (или другой парсер-генератор), создайте грамматику для простого языка выражений и сгенерируйте парсер. Напишите пример программы, которую этот парсер сможет обработать.
2. Для заданного исходного кода на простом языке (например, арифметические выражения с переменными) вручную постройте абстрактное синтаксическое дерево (AST). Объясните, какие узлы и связи в дереве соответствуют каким элементам исходного кода.
3. Объясните разницу между синтаксическим деревом разбора (parse tree) и абстрактным синтаксическим деревом (AST). Приведите примеры, когда AST проще для дальнейшей обработки.
4. Напишите функцию обхода AST (например, обход в глубину), которая вычисляет значение арифметического выражения, представленное деревом.
5. Исследуйте, как инструмент Bison (или другой генератор парсеров) обрабатывает грамматики с неоднозначностями. Приведите пример неоднозначной грамматики и предложите методы её устранения.

#### Домашнее задание по теме «Построение и анализ потоков данных (DFG)»

1. Для простой программы (например, вычисление выражения  $a = b + c * d$ ) постройте граф потоков данных (Data Flow Graph, DFG). Объясните, как узлы и ребра отражают зависимости между операциями.
2. Опишите основные типы зависимостей данных (например, истинная, анти-зависимость, выходная зависимость) и приведите примеры для каждой из них.
3. Рассмотрите программу с циклом и постройте её граф потоков данных. Объясните, как анализ потоков данных помогает в оптимизации циклов.
4. Исследуйте, как можно использовать DFG для выявления параллелизма в программе. Приведите пример, где параллельное выполнение возможно благодаря отсутствию зависимостей.
5. Напишите алгоритм обхода графа потоков данных для определения порядка вычисления операций, учитывая зависимости. Объясните, как этот порядок влияет на эффективность выполнения программы.

## Примерные вопросы для подготовки к семинарам

### Вопросы к семинару по теме «Статическая и динамическая типизация, алгоритмы вывода типов»

1. В чём основные различия между статической и динамической типизацией? Какие преимущества и недостатки у каждого подхода?
2. Как работает алгоритм вывода типов Хиндли-Милнера? Какие языки программирования используют этот алгоритм?
3. Что такое полиморфизм в контексте систем типов? Как он поддерживается в системах со статической типизацией?
4. Какие проблемы могут возникать при динамической типизации во время выполнения программы? Как их можно обнаружить и предотвратить?
5. Объясните понятия «тип», «подтип» и «совместимость типов». Как эти концепции влияют на безопасность и гибкость программ?

### Вопросы к семинару по теме «Преобразования AST»

1. Какие основные задачи решаются при преобразовании абстрактных синтаксических деревьев (AST) в компиляторах и интерпретаторах?
2. Что такое оптимизации на уровне AST? Приведите примеры простых оптимизаций, выполняемых на AST.
3. Как можно реализовать преобразование AST для устранения рекурсии или упрощения выражений?
4. Какие структуры данных и алгоритмы используются для обхода и модификации AST?
5. В чём разница между синтаксическим анализом и семантической обработкой в контексте работы с AST?

### Вопросы к семинару по теме «Генерация кода из AST»

1. Как абстрактное синтаксическое дерево (AST) используется при генерации машинного или промежуточного кода?
2. Какие этапы включает в себя процесс генерации кода из AST?
3. Как обеспечить корректное отображение управляющих конструкций (условия, циклы) из AST в целевой код?
4. Какие проблемы могут возникать при генерации кода в отношении управления памятью и оптимизации?
5. Объясните роль промежуточных представлений (IR) между AST и машинным кодом. Почему они важны?

## Примерные задания по контрольным работам

### Контрольная работа № 1

**Задание 1.** Дайте определение формальной грамматики. Какие компоненты входят в её структуру?

**Задание 2.** Объясните, что такое контекстно-свободная грамматика (КС-грамматика). Приведите пример.

**Задание 3.** Что такое производственные правила в грамматике? Как они используются для описания языка?

**Задание 4.** Опишите иерархию Хомского. Какие классы грамматик она включает?

**Задание 5.** Приведите пример грамматики, которая является контекстно-зависимой, но не контекстно-свободной.

**Задание 6.** Объясните роль грамматик в построении парсеров и синтаксическом анализе.

**Задание 7.** Напишите КС-грамматику для простого арифметического выражения,

содержащего числа и операции +, \*.

**Задание 8.** Чем отличается левая рекурсия от правой в грамматике и почему это важно для парсеров?

**Задание 9.** Опишите понятие «язык, порождаемый грамматикой». Как проверить, принадлежит ли строка этому языку?

**Задание 10.** Какие ограничения накладываются КС-грамматики на структуру языка по сравнению с более общими грамматиками?

### **Контрольная работа № 2**

**Задание 1.** Что такое парсер? Какова его роль в процессе компиляции?

**Задание 2.** Объясните разницу между нисходящим (LL) и восходящим (LR) разбором.

**Задание 3.** Какие преимущества и недостатки у LL-парсеров по сравнению с LR-парсерами?

**Задание 4.** Что такое абстрактное синтаксическое дерево (AST)? Чем оно отличается от конкретного синтаксического дерева?

**Задание 5.** Как AST используется в оптимизации и генерации кода?

**Задание 6.** Опишите принцип работы инструмента ANTLR. Для чего он применяется?

**Задание 7.** Что такое уасс? Как он помогает в построении парсеров?

**Задание 8.** Приведите пример правила грамматики и соответствующего ему узла AST.

**Задание 9.** Как в AST представляются операции с приоритетом и ассоциативностью?

**Задание 10.** Объясните, как можно преобразовать грамматику с левой рекурсией для использования в LL-парсере.

### **Контрольная работа № 3**

**Задание 1.** Дайте определение статической и динамической типизации. Приведите примеры языков с каждой из них.

**Задание 2.** Какие преимущества и недостатки у статической типизации по сравнению с динамической?

**Задание 3.** Объясните, что такое алгоритм вывода типов Хиндли-Милнера.

**Задание 4.** Как работает механизм вывода типов в функциональных языках программирования?

**Задание 5.** Что такое полиморфизм в контексте типизации? Приведите пример.

**Задание 6.** Опишите процесс статической проверки типов на примере простого выражения.

**Задание 7.** В чем заключается разница между явным объявлением типа и выводом типа?

**Задание 8.** Как сочетание статической типизации и вывода типов улучшает удобство разработки?

**Задание 9.** Приведите пример ошибки типизации, которая может быть обнаружена статической проверкой.

**Задание 10.** Объясните, как динамическая типизация влияет на производительность программ.

### **Примерное описание и критерии оценивания к итоговому 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 класс грамматик по Хомскому используется для описания синтаксиса большинства языков программирования? А. Регулярные грамматики Б. Контекстно-зависимые грамматики В. Рекурсивно-перечислимые грамматики Г. Контекстно-свободные грамматики	Г	ПК-2
2.	Как называется правило вида $A \rightarrow \alpha$ в формальной грамматике?	Продукция	ПК-2
3.	Какой метод парсинга использует разбор "сверху-вниз"?	LL	ПК-3

4.	Как называется дерево, отражающее структуру программы без синтаксических деталей?	AST	ПК-3
5.	Как называется граф, узлы которого представляют операции, а рёбра — потоки данных?	DFG	ОПК-2
6.	Какой алгоритм вывода типов используется в языках вроде Haskell?	Хиндли-Милнер	ОПК-2
7.	Как называется оптимизация, удаляющая мёртвый код из AST?	Dead-code elimination	УК-6
8.	Как называется промежуточное представление кода между AST и машинным кодом?	IR (Intermediate Representation)	УК-6
9.	Какой инструмент генерации парсеров использует грамматику ANTLR?	ANTLR	УК-6
10.	Какой стандарт управления проектами включает процессы инициации, планирования и исполнения?	PMBOK	ОПК-2