

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО

решением президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

Рабочая программа дисциплины

Введение в теорию трансляторов

Уровень высшего образования

Бакалавриат

Направление подготовки / специальность

02.03.02 - Фундаментальная информатика и информационные технологии

Направленность образовательной программы

Инженерия программного обеспечения

Форма обучения

очная

г. Нижний Новгород

2024 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.15 Введение в теорию трансляторов относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-4: Способен проектировать программное обеспечение	ПК-4.1: Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения ПК-4.2: Знает методы и средства проектирования программного обеспечения ПК-4.3: Знает методы и средства проектирования баз данных ПК-4.4: Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения ПК-4.5: Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных	ПК-4.1: Знать типовое прикладное программное обеспечение, используемое при разработке компиляторов и трансляторов. ПК-4.2: Знать основные алгоритмические и математические методы теории трансляторов и их приложения к разработке компиляторов и трансляторов. ПК-4.3: Знать основные алгоритмические и математические методы теории трансляторов и их приложения к разработке компиляторов и трансляторов. ПК-4.4: Уметь применять на практике технологии, основанные на теории трансляторов: лексический и синтаксический анализ, алгоритмы оптимизации, генерация низкоуровневых инструкций.	Практическое задание	Зачёт: Контрольные вопросы

		ПК-4.5: Уметь применять основные алгоритмические и математические методы технологии трансляторов для разработки, отладки и поддержки трансляторов.		
--	--	---	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	2
Часов по учебному плану	72
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	32
- занятия семинарского типа (практические занятия / лабораторные работы)	16
- КСР	1
самостоятельная работа	23
Промежуточная аттестация	0 Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/ лабора- торные работы), часы	Всего	
	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0	0 Ф 0
Введение. Компилятор, интерпретатор.	4	3	0	3	1
Лексический анализ.	5	3	0	3	2
Синтаксический анализ.	7	3	2	5	2
Семантический анализ.	6	3	0	3	3
Промежуточное представление.	7	4	0	4	3
Оптимизация: техники и алгоритмы.	11	4	4	8	3
Векторизация циклов и функций.	9	4	2	6	3
Генерация машинного кода.	11	4	4	8	3

Приложения компиляторных технологий.	11	4	4	8	3
Аттестация	0				
КСР	1			1	
Итого	72	32	16	49	23

Содержание разделов и тем дисциплины

Введение. Компилятор, интерпретатор.
 Лексический анализ.
 Синтаксический анализ.
 Семантический анализ.
 Промежуточное представление.
 Оптимизация: техники и алгоритмы.
 Векторизация циклов и функций.
 Генерация машинного кода.
 Приложения компиляторных технологий.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

1.<http://www.intuit.ru/studies/courses/1157/173/info> Галочкин М., Гончар Д., В. Серебряков, Фуругян М. Теория и реализация языков программирования.

2.<http://www.intuit.ru/studies/courses/26/26/info> Булычев Д., Вояковская Н., Москаль А., Терехов А. Разработка компиляторов.

б) дополнительная литература:

1.<http://www.intuit.ru/studies/courses/707/563/info> : Ануфриенко А., Идрисов Р. Введение в оптимизацию приложений с использованием компиляторов Intel.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ПК-4:

- 1) Исследование различных алгоритмов синтаксического анализа
- 2) Исследование работы синтаксического анализатора языков семейства C/C++. Освоение процедур дискретного вейвлет-анализа и синтеза (реконструкции) изображений в среде Matlab
- 3) Исследование техник оптимизации промежуточного представления кода.

- 4) Исследование возможностей оптимизации промежуточного представления кода в среде LLVM.
- 5) Исследование возможностей редактирования сигналов на основе основных классов вейвлетов в среде Matlab.
- 6) Исследование техник и алгоритмов векторизации циклов.
- 7) Исследование возможностей векторизации циклов на уровне промежуточного кода.
- 8) Исследование алгоритмов и техник генерации машинного кода.
- 9) Исследование возможностей генерации машинного кода в среде LLVM.

Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
зачтено	Выполнены все или большая часть этапов решения задачи или задача решена с незначительными недочетами. Код и результаты работы представлены преподавателю в срок.
не зачтено	Выполнены не все лабораторные работы или выполнены не в полном объеме (представлено не полное описание этапов выполнения заданий, код работает некорректно, результаты работы не представлены преподавателю).

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено		зачтено				
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие	При решении	Продемонс	Продемонс	Продемонс	Продемонс	Продемонстр

	минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	трированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	трированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	трированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	трированы все основные умения. Решены все основные задачи с отдельными и несущественными недочетами, выполнены все задания в полном объеме	трированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-4

1. Введение в компиляторы. Языковой процессор. Компилятор, интерпретатор, гибридный компилятор. Стадии компиляции на примере.
2. Лексический анализ. Сканирование исходного кода. Лексемы и токены. Выбор токена. Сложности в сканировании: примеры из c++.
3. Формальные языки. Определение формального языка: грамматика, конечный автомат, регулярное выражение.
4. Связь лексемы и токена. Сложности в сканировании: лексическая двусмысленность. Принцип "наибольшего куска" (maximal munch). Детерминированный и недетерминированный автомат.
5. Синтаксический анализ. Построение абстрактного синтаксического дерева (Abstract Syntax Tree, AST). Синтаксический анализ.
6. Ограничения в применении регулярных выражений. Контекстно-свободные грамматики. Левосторонняя продукция (leftmost derivations).
7. Дерево разбора (parse tree). Цель разбора. Сложности в синтаксическом разборе - двусмысленность. Абстрактное синтаксическое дерево (AST). Типы синтаксического разбора: LL, LR, SLR, LALR, LR.
8. Семантический анализ. Лексически- и синтаксически-корректная программа. Семантически корректная программа. Сложности в семантическом анализе.
9. Реализация семантического анализа: грамматика атрибутов, рекурсивный обход AST. Таблица символов. Проверка области видимости. Множественные проходы. Проверка типов. Сильная/слабая система типов.
10. Промежуточные представления (Intermediate Representation, IR). Среда исполнения. Представление Single Static Assignment (SSA). Построение SSA.
11. Промежуточное представление LLVM/clang, GCC. Среда исполнения. Представление данных. Стек, дерево активации.
12. Генерация трехадресного кода (Three Address Code, TAC): выражения, операторы, высказывания, циклы.
13. Таксономия преобразований промежуточного уровня.
14. Оптимизация: техники и алгоритмы. Базовый блок. Граф потока управления (Control Flow Graph, CFG).

15. Локальные оптимизации. Проброс констант. Удаление неиспользуемого кода. Агрессивное удаление неиспользуемого кода. Анализ времени жизни. Анализ потока управления. Распознавание циклов.
16. Доминирование. Отношение доминирования. Дерево доминирования. Фронтир доминирования (dominance frontier).
17. Глобальные и цикловые оптимизации. Глобальное удаление неиспользуемого кода. Глобальный проброс констант. Loop invariant code motion. Анализ алиасинга указателей.
18. Векторизация. Векторизация циклов. Векторизация функции целиком. Связь векторизации функции целиком и векторизации внешнего цикла. Преобразования потока управления в поток данных: предикатирование и маскирование побочных эффектов.
19. Генерация машинного кода. Выбор инструкций. Распределение регистров.
20. Приложения компиляторных технологий (статический анализ, форматирование кода, clang-format, и т.д.).

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
зачтено	Студент ответил на большую часть вопросов возможно с незначительными недочетами.
не зачтено	При ответе студент допускает грубые ошибки в основном материале и решении стандартных задач.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Теория и реализация языков программирования / Серебряков В.А., Галочкин М.П., Гончар Д.Р., Фуругян М.Г. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663596&idb=0>.
2. Разработка компиляторов / Вояковская Н.Н., Москаль А.Е., Булычев Д.Ю., Терехов А.А. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=663586&idb=0>.

Дополнительная литература:

1. Введение в оптимизацию приложений с использованием компиляторов Intel / Идрисов Р.И., Ануфриенко А.В. - Москва : ИНТУИТ, 2016., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=662966&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Операционные системы семейства MicrosoftWindows, лицензия по подписке MicrosoftImagine.
2. Браузер Google Chrome, предоставляется бесплатно на условиях лицензионных соглашений на программное обеспечение с открытым исходным кодом.
3. Среда разработки семейства MicrosoftVisualStudio, лицензия по подписке MicrosoftImagine.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 02.03.02 - Фундаментальная информатика и информационные технологии.

Автор(ы): Мееров Иосиф Борисович, кандидат технических наук, доцент.

Заведующий кафедрой: Мееров Иосиф Борисович, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 13.12.2023, протокол № 3.