

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

(факультет / институт / филиал)

УТВЕРЖДЕНО
президиумом Ученого совета ННГУ
от 14.12.2021 г протокол № 4

Рабочая программа дисциплины

Введение в теорию трансляторов

(наименование дисциплины (модуля))

Уровень высшего образования

бакалавриат

(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность

09.03.04 Программная инженерия

(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы

Разработка программно-информационных систем

(указывается профиль / магистерская программа / специализация)

Форма обучения

очная

(очная / очно-заочная / заочная)

Нижегород

2022 год

Место дисциплины в структуре ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений.

| № варианта | Место дисциплины в учебном плане образовательной программы | Стандартный текст для автоматического заполнения в конструкторе РПД |
|------------|--|--|
| 3 | ФТД. Факультативы | Дисциплина <i>ФТД.01 Введение в теорию трансляторов</i> является факультативом в ООП направления подготовки 09.03.04. Программная инженерия |

1. 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 | |
| <i>ПК-12. Способен осуществлять концептуальное, функциональное и логическое проектирование систем и подсистем малого и среднего масштаба и сложности.</i> | <i>ПК-12.1 Знает методы концептуального, функционального и логического проектирования</i> | <i>Знать</i> алгоритмические и математические методы, прикладное программное обеспечение теории трансляторов и их приложения к разработке компиляторов и трансляторов. | <i>Собеседование,</i> |
| | <i>ПК-12.3. Умеет применять архитектурные стили и паттерны проектирования при решении типовых задач</i> | <i>Уметь</i> применять на практике технологии, основанные на теории трансляторов: лексический и синтаксический анализ, алгоритмы оптимизации, генерация низкоуровневых инструкций. | <i>Задача</i> |
| | <i>ПК-12.4. Владеет навыками разработки проекта программной</i> | <i>пользоваться</i> навыками применения базовых технологий трансляторов для разработки, отладки и поддержки трансляторов. | <i>Задача</i> |

3. Структура и содержание дисциплины

3.1. Трудоемкость дисциплины

| | |
|---|-----------------------------|
| | очная форма обучения |
| Общая трудоемкость | 1 ЗЕТ |
| Часов по учебному плану | 36 |
| в том числе | |
| контактная работа: | 25 |
| - занятия лекционного типа | 12 |
| - занятия семинарского типа | 12 |
| - текущий контроль (КСР) | 1 |
| самостоятельная работа | 11 |
| Промежуточная аттестация – зачет | |

3.2. Содержание дисциплины

| Наименование и краткое содержание 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 | Занятия лабораторного типа | Всего | |
| Введение. Компилятор, интерпретатор. | 2 | 1 | 0 | | 1 | 1 |
| Лексический анализ. | 2 | 1 | 0 | | 1 | 1 |
| Синтаксический анализ. | 5 | 1 | 2 | | 3 | 2 |
| Семантический анализ. | 3 | 1 | 0 | | 1 | 2 |
| Промежуточное представление. | 2 | 1 | 0 | | 1 | 1 |
| Оптимизация: техники и алгоритмы. | 6 | 1 | 4 | | 5 | 1 |
| Векторизация циклов и функций. | 4 | 1 | 2 | | 3 | 1 |
| Генерация машинного кода. | 4 | 1 | 2 | | 3 | 1 |
| Приложения компиляторных технологий. | 7 | 4 | 2 | | 6 | 1 |
| | | | | | | |
| Текущий контроль (КСР) | 1 | | | | 1 | |
| Промежуточная аттестация –зачет | | | | | | |
| Итого | 36 | 12 | 12 | | 25 | 11 |

Текущий контроль успеваемости реализуется в формах опросов на занятиях семинарского типа.

Промежуточная аттестация проходит в традиционной форме (зачет).

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Для выполнения программы самостоятельной работы дисциплины «Введение в теорию трансляторов» достаточно: (а) самостоятельной проработки лекционного и дополнительного материала и (б) выполнить по согласованию с преподавателем N лабораторных работ на темы, представленные ниже в таблице. Лабораторные работы выполняются в среде Microsoft Visual Studio и инфраструктуры разработки компилятора LLVM.

| № п / п | Раздел дисциплины | Наименование лабораторных работ |
|------------------|-----------------------------------|--|
| 1 | Синтаксический анализ | 1) Исследование различных алгоритмов синтаксического анализа 2) Исследование работы синтаксического анализатора языков семейства C/C++. |
| 2 | Глобальные и цикловые оптимизации | 1) Исследование техник оптимизации промежуточного представления кода. 2) Исследование возможностей оптимизации промежуточного представления кода в среде LLVM. |
| 3 | Векторизация циклов и функций | 1) Исследование техник и алгоритмов векторизации циклов. 2) Исследование возможностей векторизации циклов на уровне промежуточного представления кода в среде LLVM. |
| 4 | Генерация машинного кода | 1) Исследование алгоритмов и техник генерации машинного кода. 2) Исследование возможностей генерации машинного кода в среде LLVM. |

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю), включающий:

5.1. Описание шкал оценивания результатов обучения по дисциплине

| Уровень сформированности компетенций (индикатора) | Шкала оценивания сформированности компетенций | | | | | | |
|---|---|---------------------|-------------------|--------|--------------|---------|-------------|
| | плохо | неудовлетворительно | удовлетворительно | хорошо | очень хорошо | отлично | превосходно |

| достижения компетенций) | 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 ошибки. | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки, без ошибок. | Уровень знаний в объеме, превышающем программу подготовки. |
| <u>Умения</u> | Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа | При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки. | Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме. | Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами. | Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами. | Продemonстрированы все основные умения, решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме. | Продemonстрированы все основные умения, решены все основные задачи. Выполнены все задания, в полном объеме без недочетов |
| <u>Навыки</u> | Отсутствие владения материалом. 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 некоторыми недочетами. | Продemonстрированы базовые навыки при решении стандартных задач с некоторыми недочетами | Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов. | Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов. | Продemonстрированы творческий подход к решению нестандартных задач. |

Шкала оценки при промежуточной аттестации

| Оценка | | Уровень подготовки |
|------------|---------------------|--|
| зачтено | Превосходно | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно» |
| | Отлично | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично» |
| | Очень хорошо | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо» |
| | Хорошо | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «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», ни одна из компетенций не сформирована на уровне «плохо» |
| | Плохо | Хотя бы одна компетенция сформирована на уровне «плохо» |

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

5.2.1 Контрольные вопросы

| Вопросы | Код компетенции (согласно РПД) |
|---|-----------------------------------|
| 1. Введение в компиляторы. Языковой процессор. Компилятор, интерпретатор, гибридный компилятор. Стадии компиляции на примере. | ПК-12 |
| 2. Лексический анализ. Сканирование исходного кода. Лексемы и токены. Выбор токена. Сложности в сканировании: примеры из c++. | ПК-12 |
| 3. Формальные языки. Определение формального языка: грамматика, конечный автомат, регулярное выражение. | ПК-12 |
| 4. Связь лексемы и токена. Сложности в сканировании: лексическая двусмысленность. Принцип "наибольшего куска" (maximal munch). Детерминированный и недетерминированный автомат. | ПК-12 |
| 5. Синтаксический анализ. Построение абстрактного синтаксического дерева (Abstract Syntax Tree, AST). Синтаксический анализ. | ПК-12 |
| 6. Ограничения в применении регулярных выражений. | ПК-12 |

| | |
|--|-------|
| 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. | ПК-12 |
| 8. Семантический анализ. Лексически- и синтаксически-корректная программа. Семантически корректная программа. Сложности в семантическом анализе. | ПК-12 |
| 9. Реализация семантического анализа: грамматика атрибутов, рекурсивный обход AST. Таблица символов. Проверка области видимости. Множественные проходы. Проверка типов. Сильная/слабая система типов. | ПК-12 |
| 10. Промежуточные представления (Intermediate Representation, IR). Среда исполнения. Представление Single Static Assignment (SSA). Построение SSA. | ПК-12 |
| 11. Промежуточное представление LLVM/clang, GCC. Среда исполнения. Представление данных. · Стек, дерево активации. | ПК-12 |
| 12. Генерация трехадресного кода (Three Address Code, TAC): выражения, операторы, высказывания, циклы. | ПК-12 |
| 13. Таксономия преобразований промежуточного уровня. | ПК-12 |
| 14. Оптимизация: техники и алгоритмы. Базовый блок. Граф потока управления (Control Flow Graph, CFG). | ПК-12 |
| 15. Локальные оптимизации. Проброс констант. Удаление неиспользуемого кода. Агрессивное удаление неиспользуемого кода. Анализ времени жизни. Анализ потока управления. Распознавание циклов. | ПК-12 |
| 16. Доминирование. Отношение доминирования. Дерево доминирования. Фронт доминирования (dominance frontier). | ПК-12 |
| 17. Глобальные и цикловые оптимизации. Глобальное удаление неиспользуемого кода. Глобальный проброс констант. Loop invariant code motion. Анализ алиасинга указателей. | ПК-12 |
| 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. | ПК-12 |
| 19. Генерация машинного кода. Выбор инструкций. Распределение регистров. | ПК-12 |
| 20. Приложения компиляторных технологий (статический анализ, форматирование кода, clang-format, и т.д.). | ПК-12 |

5.2.2 Типовые задачи для оценки сформированности компетенции ПК-12

- 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.

Вопросы для индивидуального собеседования

1. Введение в компиляторы.

- Зачем нужно изучать компиляторы. Краткая история. Как быть успешным в компиляторах, необходимые знания и навыки.
- 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++.
- Связь лексемы и токена. Формальные языки. Определение формального языка: грамматика, конечный автомат, регулярное выражение.
 - Сложности в сканировании: лексическая двусмысленность. Принцип "наибольшего куска" (maximal munch). Детерминированный и недетерминированный автомат.

3. Синтаксический анализ. Построение абстрактного синтаксического дерева (Abstract Syntax Tree, AST).

- Синтаксический анализ. Ограничения в применении регулярных выражений.
- Контекстно-свободные грамматики. Левосторонняя продукция (leftmost derivations). Дерево разбора (parse tree). Цель разбора.
- Сложности в синтаксическом разборе - двусмысленность.
- Абстрактное синтаксическое дерево (AST). Типы синтаксического разбора: LL, LR, SLR, LALR, LR.

4. Семантический анализ.

- Лексически- и синтаксически-корректная программа.
- Семантически корректная программа.

- Сложности в семантическом анализе.
- Реализация семантического анализа: грамматика атрибутов, рекурсивный обход AST.

- Таблица символов. Проверка области видимости. Множественные проходы.

- Проверка типов. Сильная/слабая система типов.

5. Промежуточные представления (Intermediate Representation, IR). Среда исполнения.

- Представление Single Static Assignment (SSA). Построение SSA.

- Промежуточное представление LLVM/clang, GCC.

- Среда исполнения. Представление данных.

- Стек, дерево активации.

- Замыкания (closure). Ко-рутины (coroutines).

- Реализация объектов: классы, наследование, указатель this, таблица виртуальных функций (Vtable), динамическая проверка типов.

- Генерация трехадресного кода (Three Address Code, TAC): выражения, операторы, высказывания, циклы.

6. Таксономия преобразований промежуточного уровня.

7. Оптимизация: техники и алгоритмы.

- Базовый блок. Граф потока управления (Control Flow Graph, CFG).

- Локальные оптимизации.

- § Проброс констант. Удаление неиспользуемого кода. Агрессивное удаление неиспользуемого кода. Анализ времени жизни.

- Анализ потока управления. Распознавание циклов.

- § Доминирование. Отношение доминирования. Дерево доминирования. Фронт доминирования (dominance frontier).

- Глобальные и цикловые оптимизации.

- § Глобальное удаление неиспользуемого кода. Глобальный проброс констант.

- § Loop invariant code motion.

- Анализ алиасинга указателей.

- 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.

8. Генерация машинного кода.

- Выбор инструкций.
- Распределение регистров.

9. Приложения компиляторных технологий (статический анализ, форматирование кода, clang-format, и т.д.).

6. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература

1. Галочкин М., Гончар Д., В. Серебряков, Фуругян М. Теория и реализация языков программирования. <http://www.intuit.ru/studies/courses/1157/173/info>
2. Булычев Д., Вояковская Н., Москаль А., Терехов А. Разработка компиляторов. <http://www.intuit.ru/studies/courses/26/26/info>
3. Малявко, А. А. Формальные языки и компиляторы : учебное пособие для вузов / А. А. Малявко. — М. : Издательство Юрайт, 2017. — 429 с. — (Серия : Университеты России). — ISBN 978-5-534-04288-7. — Режим доступа : www.biblio-online.ru/book/B4D96654-71D5-4748-986D-66E8309C25E3.

б) дополнительная литература:

1. Ануфриенко А., Идрисов Р. Введение в оптимизацию приложений с использованием компиляторов Intel. <http://www.intuit.ru/studies/courses/707/563/info>
2. Трофимов, В. В. Алгоритмизация и программирование: учебник для академического бакалавриата / В. В. Трофимов, Т. А. Павловская ; под ред. В. В. Трофимова. — М. : Издательство Юрайт, 2017. — 137 с. — (Серия : Бакалавр. Академический курс. Модуль.). — ISBN 978-5-9916-9866-5. — Режим доступа : www.biblio-online.ru/book/B08DB966-3F96-4B5A-B030-E3CD9085CED4.

в) Интернет-ресурсы

1. Единое окно доступа к образовательным ресурсам (Программирование) <http://window.edu.ru>
2. Электронно-библиотечная система <https://e.lanbook.com/>

7. Материально-техническое обеспечение дисциплины

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой (лекционного и семинарского типа), оснащенные оборудованием и техническими средствами обучения.

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 **09.03.04 Программная инженерия**.

Автор _____ Е. С. Тюрин

Рецензент (ы) _____

Заведующий кафедрой _____ Р.Г. Стронгин

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики

от 1.12.2021 года, протокол № 2.