

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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)

УТВЕРЖДАЮ:

Директор _____ В.П. Гергель

« ____ » _____ 2018г.

Рабочая программа дисциплины (модуля)

Введение в теорию трансляторов

(наименование дисциплины (модуля))

Уровень высшего образования

бакалавриат

(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность

02.03.02 Фундаментальная информатика и информационные технологии

(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы

Инженерия программного обеспечения

(указывается профиль / магистерская программа / специализация)

Квалификация (степень)

бакалавр

(бакалавр / магистр / специалист)

Форма обучения

очная

(очная / очно-заочная / заочная)

Нижегород

2018

1. Место и цели дисциплины (модуля) в структуре ОПОП

Дисциплина Б1.В.ДВ.14.01 «Введение в теорию трансляторов» относится к дисциплинам по выбору вариативной части ОПОП по направлению подготовки «Фундаментальная информатика и информационные технологии (Профиль: Инженерия программного обеспечения)». Трудоемкость дисциплины составляет 2 зачетные единицы.

Цель освоения дисциплины

Целью освоения дисциплины «Введение в теорию трансляторов» является изучение основных моделей и методов теории трансляторов. Рассматриваются основные этапы создания транслятора, изучаются возникающие проблемы и методы их решения. Основное внимание уделяется развитию навыков разработки отдельных подсистем компилятора.

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников)

Формируемые компетенции	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций
<i>ПК-1: способность к ведению научно-исследовательской деятельности в области информатики и информационных технологий; способность использовать современные инструментальные и вычислительные средства (ПК-1-4); (завершающий этап)</i>	Знать З1 (ПК-1-4) Алгоритмические и математические методы, прикладное программное обеспечение теории трансляторов и их приложения к разработке компиляторов и трансляторов. Уметь У1 (ПК-1-4) Применять на практике технологии, основанные на теории трансляторов: лексический и синтаксический анализ, алгоритмы оптимизации, генерация низкоуровневых инструкций. Владеть В1 (ПК-1-4) Навыками применения базовых технологий трансляторов для разработки, отладки и поддержки трансляторов.

3. Структура и содержание дисциплины Введение в теорию трансляторов

Объем дисциплины «Введение в теорию трансляторов» составляет 2 зачетные единицы, всего 72 часа, из которых 21 час составляет контактная работа обучающегося с преподавателем (10 часов занятия лекционного типа, 10 часов занятия лабораторного типа, 1 час– мероприятия промежуточной аттестации), 51 час составляет самостоятельная работа обучающегося.

Содержание дисциплины (модуля)

Наименование и краткое содержание разделов и тем дисциплины (модуля), форма промежуточной аттестации по дисциплине	Всего (часы)	В том числе					Самостоятельная работа обучающегося, часы
		Контактная работа (работа во взаимодействии с преподавателем), часы из них				Всего	
		Занятия лекционного типа	Занятия семинарского типа	Занятия лабораторного типа	Консультации		
Введение. Компилятор, интерпретатор.	3	1	–	0	–	1	2
Лексический анализ.	7	1	–		–	1	6
Синтаксический анализ.	9	1	–	2	–	3	6
Семантический анализ.	9	1	–		–	1	8
Промежуточное представление.	8	1	–		–	2	6
Оптимизация: техники и алгоритмы.	11	1	–	4	–	5	6
Векторизация циклов и функций.	9	1	–	2	–	3	6
Генерация машинного кода.	9	1	–	2	–	3	6
Приложения компиляторных технологий.	7	2	–		–	2	5
В т.ч. текущий контроль 2 ч.							
Промежуточная аттестация <u>зачет</u>							

4. Образовательные технологии

Используются образовательные технологии в форме лекции-информации, лабораторных работ и самостоятельных практических работ. Лабораторные занятия - одна из форм учебного занятия, направленная на развитие самостоятельности обучающихся и приобретение умений и навыков. Данные учебные занятия углубляют, расширяют, детализируют полученные на лекции знания. Лабораторное занятие предполагает выполнение студентами по заданию и под руководством преподавателей нескольких практических работ. При выполнении лабораторных работ, при самостоятельной работе и

подготовке к промежуточной аттестации студенты имеют доступ к материалам курса, размещенным на Интернет-сайтах, режим доступа – свободный.

5. 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.

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

6.1. Перечень компетенций выпускников образовательной программы с указанием результатов обучения (знаний, умений, владений), характеризующих этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования.

Оценка уровня формирования компетенции ОПК-2

Индикаторы компетенции	Критерии оценивания (дескрипторы)	Шкала оценивания
Знать З1 (ПК-1-4)	Полное отсутствие навыков, предусмотренных компетенцией.	Плохой уровень формирования

Алгоритмические и математические методы, прикладное программное обеспечение теории трансляторов и их приложения к разработке компиляторов и трансляторов. Уметь У1 (ПК-1-4) Применять на практике технологии, основанные на теории трансляторов: лексический и синтаксический анализ, алгоритмы оптимизации, генерация низкоуровневых инструкций. Владеть В1 (ПК-1-4) Навыками применения базовых технологий трансляторов для разработки, отладки и поддержки трансляторов.		компетенции. «Плохо»
	Отсутствие навыков, предусмотренных данной компетенцией	Неудовлетворительный уровень формирования компетенции. «неудовлетворительно»
	Знать некоторые основные понятия, определения, математические методы, прикладное программное обеспечение теории трансляторов и их приложения к разработке компиляторов и трансляторов. Уметь У1 с погрешностями. Владеть некоторыми основными навыками, демонстрируя их в стандартных ситуациях.	Удовлетворительный уровень формирования компетенции. «Удовлетворительно»
	Знать основные понятия, определения, математические методы, прикладное программное обеспечение теории трансляторов и их приложения к разработке компиляторов и трансляторов с некоторыми погрешностями. Уметь У1 с незначительными погрешностями. Владеть основными навыками, демонстрируя их в стандартных ситуациях	Хороший уровень формирования компетенции. «Хорошо»
	Знать основные понятия, определения, математические методы, прикладное программное обеспечение теории трансляторов и их приложения к разработке компиляторов и трансляторов с незначительными погрешностями Уметь У1 с незначительными погрешностями. Владеть всеми основными навыками, демонстрируя их в стандартных ситуациях	Очень хороший уровень формирования компетенции «Очень хорошо»
	Знать основные понятия, определения, математические методы, прикладное программное обеспечение теории трансляторов и их приложения к разработке компиляторов и трансляторов без погрешностей Уметь У1 в полном объеме. Владеть всеми навыками, демонстрируя их в стандартных ситуациях.	Отличный уровень формирования компетенции «Отлично»
	Знать основной и дополнительный материал без ошибок и погрешностей. Уметь У1 в полном объеме. Свободно Владеть всеми навыками, демонстрируя их в стандартных и нестандартных ситуациях.	Превосходный уровень формирования компетенции «Превосходно»

6.2. Описание шкал оценивания

Для оценивания результатов учебной деятельности студентов при изучении дисциплины используется балльная система оценки учебной работы студентов. По результатам промежуточной аттестации: в случае зачета проставляются оценки «Зачтено» (соответствует уровням оценки компетенций «удовлетворительно» и выше) и «Не зачтено» (соответствует уровням оценки компетенций «плохо» и «неудовлетворительно»).

6.3. Критерии и процедуры оценивания результатов обучения по дисциплине (модулю), характеризующих этапы формирования компетенций

Для оценивания результатов обучения в виде знаний используются следующие процедуры и технологии:

- индивидуальное собеседование.

Для оценивания результатов обучения в виде умений и владений используются следующие процедуры и технологии:

- лабораторные работы, включающие постановку одной сложной учебной задачи в виде краткой формулировки действий, которые следует выполнить, и описания результата, который нужно получить.
- индивидуальное собеседование по результатам лабораторной работы.

6.4. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций и (или) для промежуточного контроля сформированности компетенции (ПК-1).

Задания для оценки сформированности компетенции ПК-1-4 в части «Уметь», «Владеть»

- 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 (оценивания результатов сформированности компетенции ПК-1-4 в виде знаний):

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. Фронтир 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.).

6.5. Методические материалы, определяющие процедуры оценивания.

Положение о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся в ННГУ от 13.02.2014.

http://www.unn.ru/pages/general/norm-acts/attest_stud%202014.pdf

7. Учебно-методическое и информационное обеспечение дисциплины «Введение в теорию трансляторов»

а) основная литература

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.

8. Материально-техническое обеспечение дисциплины (модуля)

Имеются в наличии учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, промежуточной аттестации, а также помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет». Компьютерный класс с доступом в интернет для демонстрации результатов домашних практических работ. Учебная и научная литература, учебно-методические материалы, представленные в библиотечном фонде, в электронных библиотеках.

Для лабораторных работ: Компьютерный класс ПЭВМ, удовлетворяющий следующим требованиям (в компьютерном классе установлена операционная система Windows (лицензия), Microsoft Visual Studio (лицензия), Microsoft Office (лицензия), Matlab (лицензия)) :.

Минимальные	Оптимальные
Процессор - не ниже Pentium IV; Графический процессор - не ниже CUDA или MS DirectX10; Программное обеспечение: MS Visual Studio .NET не ниже 2008 версии.	Процессор - Intel® Core™ i7 Графический процессор – NVIDIA GeForce 580 и выше, Программное обеспечение: MS Visual Studio .NET 2015.

Программа составлена в соответствии с требованиями ОС ННГУ с учетом рекомендаций ОПОП ВО по направлению 02.02.02 - Фундаментальная информатика и информационные технологии.

Автор Е. С. Тюрин

Заведующий кафедрой _____ Р.Г. Стронгин

Программа одобрена методической комиссией Института информационных технологий, математики и механики ННГУ им. Н.И. Лобачевского

от 20 июня 2018 года, протокол № 10