

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  
решением Ученого совета ННГУ  
протокол №13 от 30.11.2022 г.

**Рабочая программа дисциплины**  
**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)

Форма обучения

очная

(очная / очно-заочная / заочная)

Нижегород  
2023 год

## 1. Место дисциплины в структуре ОПОП

№ варианта	Место дисциплины в учебном плане образовательной программы	Стандартный текст для автоматического заполнения в конструкторе РПД
2	Блок 1. Дисциплины (модули) Часть формируемая участниками образовательных отношений	Дисциплина <i>Б1.В.15 Введение в теорию трансляторов</i> относится к части ООП направления подготовки 02.03.02 Фундаментальная информатика и информационные технологии, формируемой участниками образовательных отношений.

## 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**	
<b>ПК-4.:</b> Способен проектировать программное обеспечение	<b>ПК-4.1:</b> Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения <b>ПК-4.2:</b> Знает методы и средства проектирования программного обеспечения <b>ПК-4.3.:</b> Знает методы и средства проектирования баз данных	<b>Знать</b> алгоритмические и математические методы, прикладное программное обеспечение теории трансляторов и их приложения к разработке компиляторов и трансляторов.	<i>Собеседование,</i>
	<b>ПК-4.4.:</b> Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения	Уметь применять на практике технологии, основанные на теории трансляторов: лексический и синтаксический анализ, алгоритмы оптимизации, генерация низкоуровневых инструкций. Уметь пользоваться навыками применения базовых технологий	<i>Задача</i>

	<b>ПК-4.5.:</b> Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных	трансляторов для разработки, отладки и поддержки трансляторов.	
--	--	--	--

### 3. Структура и содержание дисциплины

#### 3.1. Трудоемкость дисциплины

	<b>очная форма обучения</b>
<b>Общая трудоемкость</b>	<b>2 ЗЕТ</b>
<b>Часов по учебному плану</b>	<b>72</b>
<b>в том числе</b>	
<b>контактная работа:</b>	<b>25</b>
- занятия лекционного типа	<b>32</b>
- занятия лабораторного типа	<b>16</b>
- текущий контроль (КСР)	<b>1</b>
<b>самостоятельная работа</b>	<b>23</b>
<b>Промежуточная аттестация – зачет</b>	

#### 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.	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
Текущий контроль (КСР)	1				1	
Промежуточная аттестация –зачет						
<b>Итого</b>	<b>72</b>	<b>32</b>		<b>16</b>	<b>49</b>	<b>23</b>

Текущий контроль успеваемости реализуется в формах опросов на занятиях лабораторного 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.

	Раздел дисциплины	Наименование лабораторных работ
	Синтаксический анализ	1) Исследование различных алгоритмов синтаксического анализа 2) Исследование работы синтаксического анализатора языков семейства C/C++.
	Глобальные и цикловые оптимизации	1) Исследование техник оптимизации промежуточного представления кода. 2) Исследование возможностей оптимизации промежуточного представления кода в среде LLVM.
	Векторизация циклов и функций	1) Исследование техник и алгоритмов векторизации циклов. 2) Исследование возможностей векторизации циклов на уровне промежуточного представления кода в среде LLVM.
	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 (индикатор)	Шкала оценивания сформированности компетенций						
	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	Не зачтено		Зачтено				

а достижени я компетенц ий)							
<u>Знания</u>	Отсутств ие знаний теоретич еского материал а. Невозмо жность оценить полноту знаний вследств ие отказа обучающ егося от ответа	Уровень знаний ниже минималь ных требований. Имели место грубые ошибки.	Минимально допустимый уровень знаний. Допущено много негрубых ошибки.	Уровень знаний в объеме, соответству ющем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответству ющем программе подготовки. Допущено несколько несуществен ных ошибок	Уровень знаний в объеме, соответству ющем программе подготовки, без ошибок.	Уровень знаний в объеме, превышающ ем программу подготовки.
<u>Умения</u>	Отсутств ие минимал ьных умений. Невозмо жность оценить наличие умений вследств ие отказа обучающ егося от ответа	При решении стандартных задач не продемонстр ированы основные умения. Имели место грубые ошибки.	Продемонст рированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Продемонст рированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонст рированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонст рированы все основные умения, решены все основные задачи с отдельными несущест венным недочетами, выполнены все задания в полном объеме.	Продемонст рированы все основные умения, решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутств ие владения материал ом. Невозмо жность оценить наличие навыков вследств ие отказа обучающ егося от ответа	При решении стандартных задач не продемонстр ированы базовые навыки. Имели место грубые ошибки.	Имеется минимальны й набор навыков для решения стандартных задач с некоторыми недочетами.	Продемонст рированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонст рированы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продемонст рированы навыки при решении нестандартн ых задач без ошибок и недочетов.	Продемонст рирован творческий подход к решению нестандартн ых 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-4
2. Лексический анализ. Сканирование исходного кода. Лексемы и токены. Выбор токена. Сложности в сканировании: примеры из c++.	ПК-4
3. Формальные языки. Определение формального языка: грамматика, конечный автомат, регулярное выражение.	ПК-4
4. Связь лексемы и токена. Сложности в сканировании: лексическая двусмысленность. Принцип "наибольшего куска" (maximal munch). Детерминированный и недетерминированный автомат.	ПК-4
5. Синтаксический анализ. Построение абстрактного синтаксического дерева (Abstract Syntax Tree, AST). Синтаксический анализ.	ПК-4
6. Ограничения в применении регулярных выражений. Контекстно-свободные грамматики. Левосторонняя продукция (leftmost derivations).	ПК-4
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-4
8. Семантический анализ. Лексически- и синтаксически-корректная программа. Семантически корректная программа. Сложности в семантическом анализе.	ПК-4
9. Реализация семантического анализа: грамматика атрибутов, рекурсивный обход AST. Таблица символов. Проверка области видимости. Множественные проходы. Проверка типов.	ПК-4

Сильная/слабая система типов.	
10. Промежуточные представления (Intermediate Representation, IR). Среда исполнения. Представление Single Static Assignment (SSA). Построение SSA.	ПК-4
11. Промежуточное представление LLVM/clang, GCC. Среда исполнения. Представление данных. · Стек, дерево активации.	ПК-4
12. Генерация трехадресного кода (Three Address Code, TAC): выражения, операторы, высказывания, циклы.	ПК-4
13. Таксономия преобразований промежуточного уровня.	ПК-4
14. Оптимизация: техники и алгоритмы. Базовый блок. Граф потока управления (Control Flow Graph, CFG).	ПК-4
15. Локальные оптимизации. Проброс констант. Удаление неиспользуемого кода. Агрессивное удаление неиспользуемого кода. Анализ времени жизни. Анализ потока управления. Распознавание циклов.	ПК-4
16. Доминирование. Отношение доминирования. Дерево доминирования. Фронт доминирования (dominance frontier).	ПК-4
17. Глобальные и цикловые оптимизации. Глобальное удаление неиспользуемого кода. Глобальный проброс констант. Loop invariant code motion. Анализ алиасинга указателей.	ПК-4
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-4
19. Генерация машинного кода. Выбор инструкций. Распределение регистров.	ПК-4
20. Приложения компиляторных технологий (статический анализ, форматирование кода, clang-format, и т.д.).	ПК-4

### 5.2.2 Типовые задачи для оценки сформированности компетенции ПК-4

#### Задания для оценки сформированности компетенции ПК-4 в части «Уметь»

- 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.

## **6. Учебно-методическое и информационное обеспечение дисциплины**

### **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.

## **7. Материально-техническое обеспечение дисциплины**

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: операционная система Windows (лицензия), Microsoft Visual Studio (лицензия), Microsoft Office (лицензия), Matlab (лицензия)

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 02.03.02 Фундаментальная информатика и информационные технологии.

Автор Е. С. Тюрин

Рецензент (ы) \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_ Р.Г. Стронгин

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики 30.11.2022 года, протокол № 3