Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 07.11.2023 14:27:43
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет информационных технологий

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____ / Д.Г. Демидов /
«16» 02 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Методы трансляции и компиляции»

Направление подготовки
09.03.01 Информатика и вычислительная техника

Профиль
«Системная и программная инженерия»

Квалификация
Бакалавр

Формы обучения
очная

Москва, 2023 г.

Разработчик(и):

старший преподаватель кафедры «Информационная безопасность»

/А.Ю. Гневшев/

Согласовано:
И.о. заведующего кафедрой «Информационная безопасность»,

/А.Ю. Гневшев/

Руководитель образовательной программы,

/А.Ю. Гневшев/

Заведующий кафедрой «Инфокогнитивные технологии»,

доцент, к.т.н. /Е.А.Пухова/

1 Цели, задачи и планируемые результаты обучения по дисциплине

Цель изучения дисциплины – обучение студентов навыкам использования машинного обучения и анализа данных для последующей разработки стратегий продвижения продуктов и услуг в цифровом пространстве.

Обучение по дисциплине «Методы трансляции и компиляции» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать программы на основе спроектированного алгоритма и проводить отладку программы, применять методы системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектно-ориентированный анализ иерархии классов. осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к базовой части основной профессиональной образовательной программы бакалавриата.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Языки программирования.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, т.е. 144 академических часа (из них 72 часа – самостоятельная работа студентов и 72 часа – аудиторные занятия).

Разделы дисциплины изучаются в шестом семестре обучения, т.е. на третьем курсе. Форма контроля – экзамен.

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			6	
1	**Аудиторные занятия**	**72**	72	
	В том числе:			
1.1	Лекции	4	4	
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	68	68	
2	**Самостоятельная работа**	**72**	72	
3	**Промежуточная аттестация**			
	Экзамен		экзамен	
	Итого:	**144**	144	

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Введение в компиляцию	28	2		16		18
2	Лексический анализ	44	2		18		18
3	Синтаксический анализ	46			18		18
4	Генерация промежуточного кода, генерация кода	26			16		18
Итого		**144**	**4**		**68**		**72**

3.3 Содержание дисциплины

1. Введение в компиляцию и интерпретацию: определение, основные этапы процесса компиляции, сравнение компиляции и интерпретации.
2. Синтаксический анализ: основные структуры данных и алгоритмы синтаксического анализа, LL(1) и LR(1) методы разбора, префиксные деревья, обработка синтаксических ошибок.
3. Семантический анализ: лексический анализ, синтаксические диаграммы, контекстно-свободные грамматики, прямая и обратная польская запись, абстрактное синтаксическое дерево, присваивание имён.
4. Оптимизация кода: понятие стоимости, виды оптимизации (удаление мёртвого кода, удаление избыточного кода, оптимизация констант, инлайнинг функций), различные методы оценки стоимости.
5. Генерация кода: представление кода на языке ассемблера, размещение и связывание, генерация кода для различных архитектур, оптимизация кода на уровне ассемблерных инструкций.
6. Трансляция в машинный код: понятие виртуальной машины, JIT-компиляция, динамическая трансляция, компиляция «на лету».
7. Параллельная компиляция и оптимизация: принципы и подходы к параллельной компиляции, разделение работы между процессорами, анализ и оптимизация параллельного кода.
8. Языки программирования и стандарты: стандарты языков программирования (ANSI C, C++, Java, C# и др.), их особенности и отличия, принципы разработки компиляторов для этих языков.
9. Инструменты для разработки компиляторов: обзор существующих инструментов и библиотек для разработки компиляторов (Lex/Yacc, ANTLR, SWIG, GCC и др.), описание их возможностей и ограничений.
10. Практические проекты и исследования: разработка и анализ реальных проектов компиляции и трансляции, подготовка отчётов о проделанной работе, участие в научных исследованиях в области компиляции и оптимизации кода.

4 Учебно-методическое и информационное обеспечение

4.1 Основная литература

1. Малявко, А. А. Формальные языки и компиляторы : учебное пособие для вузов / А. А. Малявко. — Москва : Издательство Юрайт, 2022. — 429 с. — (Высшее образование). — ISBN 978-5-534-04288-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/492129

4.2 Дополнительная литература

1. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2021. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/469759

2. Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для вузов / О. В. Казарин, А. С. Забабурин. — Москва : Издательство Юрайт, 2023. — 312 с. — (Высшее образование). — ISBN 978-5-9916-9043-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/513300

5 Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Практические занятия (семинары) и самостоятельная работа студентов должна проводиться в специализированных аудиториях с комплектом мультимедийного оборудования и/или доской для записей материалов. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

5.2 Требования к программному обеспечению

Для проведения практических занятий (семинаров) специального программного обеспечения для освоения дисциплины не требуется.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, семинары и практики.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется на зачете в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;

- сформированность компетенций;
- оформление материала в соответствии с требованиями.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- экзамен.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения				
ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать программы на основе спроектированного алгоритма и проводить отладку программы, применять методы	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины.	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.

системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектно-ориентированный анализ иерархии классов. осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.				

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: экзамен.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом

по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине.

Шкала оценивания	Описание
Отлично	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

7.3.1 Экзаменационные вопросы

1. Определение и основные задачи компиляции.
2. Основные этапы трансляции программы.
3. Синтаксические анализаторы и их виды.
4. LL(1)-анализаторы и их построение.
5. LR(1)-анализаторы и их особенности.

6. Префиксные деревья и их применение в синтаксическом анализе.
7. Лексический анализ и его функции.
8. Синтаксическая диаграмма и ее роль в семантическом анализе.
9. Контекстно-свободная грамматика и ее применение в семантическом анализе.
10. Абстрактное синтаксическое дерево и его свойства.
11. Прямая и обратная польская записи и их использование в семантическом анализе.
12. Понятие стоимости и его использование в оптимизации кода.
13. Виды оптимизации и их роль в улучшении кода.
14. Удаление мёртвого кода и его роль в оптимизации.
15. Удаление избыточного кода и его влияние на оптимизацию.
16. Оптимизация констант и ее значение для оптимизации кода.
17. Инлайнинг функций и его применение для оптимизации.
18. Различные методы оценки стоимости и их влияние на оптимизацию кода.
19. Представление кода на языке ассемблера и его особенности.
20. Размещение и связывание и их значение для генерации кода.
21. Генерация кода для различных архитектурных платформ и ее особенности.
22. Оптимизация машинного кода на уровне ассемблерных инструкций и ее влияние на производительность.
23. Виртуальная машина и ее использование в трансляции в машинный код.
24. JIT-компиляция и ее роль в трансляции в машинный код.
25. Динамическая трансляция и ее применение при компиляции “на лету”.
26. Принципы параллельной компиляции и их реализация.
27. Разделение работы между процессорами и его значение для параллельной оптимизации.
28. Анализ и оптимизация параллельного кода и их роль для повышения производительности.
29. Стандарты языков программирования ANSI C и их основные особенности.
30. Стандарты языков программирования C++ и их отличительные характеристики.
31. Стандарты языков программирования Java и их основные свойства.
32. Стандарты языков программирования C# и их ключевые моменты.
33. Обзор инструментов для разработки компиляторов Lex/Yacc и их возможности.
34. Обзор инструментов для разработки компиляторов ANTLR и их ограничения.
35. Обзор инструментов для разработки компиляторов SWIG и их функции.
36. Обзор инструментов для разработки компиляторов GCC и их характеристики.
37. Разработка и анализ реального проекта компиляции на примере языка C.
38. Разработка и анализ реального проекта трансляции на примере языка Java.
39. Подготовка отчета о проделанной работе в рамках практического проекта в области компиляции.
40. Участие в научном исследовании в области оптимизации кода и его результаты.
41. Компиляция для микроконтроллеров и ее особенности.
42. Компиляция под мобильные платформы и ее специфические черты.
43. Компиляция на платформе .NET и ее основные характеристики.
44. Компиляция в среде LLVM и ее преимущества.
45. Технологии автоматического распараллеливания и их применение.
46. Использование библиотек для автоматического распараллеливания кода и их возможности.
47. Применение автоматических оптимизаторов кода и их эффективность.
48. Оптимизация времени выполнения и ее значение в компиляции и трансляции.
49. Оптимизация памяти и ее роль при генерации машинного кода.
50. Оценка качества работы компилятора и методы ее проведения.

7.3.2 Типовые практические задания

1. Разработать компилятор для простого языка программирования, который поддерживает арифметические операции, переменные и условные операторы.
2. Написать программу для синтаксического анализа выражения на языке LL(1).
3. Реализовать алгоритм LR(1) для разбора выражения, содержащего переменные и арифметические операции.
4. Создать программу для генерации машинного кода из абстрактного синтаксического дерева.
5. Разработать программу для оптимизации констант в исходном коде на языке программирования.
6. Написать программу для генерации ассемблерного кода из абстрактного синтаксического дерева с использованием виртуальной машины.
7. Реализовать параллельную компиляцию исходного кода на нескольких процессорах.
8. Написать программу для создания компилятора для языка программирования с использованием инструмента Lex/Yacc.
9. Написать программу для оптимизации параллельного кода с использованием автоматического распараллеливания.

7.3.3 Типовой экзаменационный билет

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1
по дисциплине
«Методы трансляции и компиляции»
направление подготовки 09.03.01 Информатика и вычислительная техника

ВОПРОСЫ:

1. Лексический анализ и его функции.
2. Оптимизация памяти и ее роль при генерации машинного кода.
3. Динамическая трансляция и ее применение при компиляции “на лету”.

Утверждено: _____ / _____ / «__» _____ 20__г.