

МИНОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук



Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине
(Оценочные средства по дисциплине)

Методы компиляции

по направлению подготовки

01.03.02 Прикладная математика и информатика

Направленность (профиль) подготовки:

Математическое моделирование и информационные системы

ОС составил:
ассистент кафедры компьютерной безопасности

 В.А. Провкин


Рецензент:
канд. техн. наук, доцент,
заведующий кафедрой компьютерной безопасности

 С.А. Останин

Оценочные средства одобрены на заседании учебно-методической комиссии института прикладной математики и компьютерных наук (УМК ИПМКН).

Протокол от 08.06.2023 г. № 02

Председатель УМК ИПМКН,
д-р техн. наук, профессор

 С.П. Сущенко

Оценочные средства (ОС) являются элементом системы оценивания сформированности компетенций у обучающихся в целом или на определенном этапе ее формирования.

ОС разрабатывается в соответствии с рабочей программой (РП) дисциплины.

1. Компетенции и результаты обучения, формируемые в результате освоения дисциплины

Компетенция	Индикатор компетенции	Код и наименование результатов обучения (планируемые результаты обучения, характеризующие этапы формирования компетенций)	Критерии оценивания результатов обучения			
			Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач.	ИОПК-2.1. Обладает навыками объектно-ориентированного программирования для решения прикладных задач в профессиональной деятельности.	ОР-ОПК2-2.1.1. Уметь разрабатывать архитектуру основных блоков компилятора с использованием методов объектно-ориентированного проектирования.	Студент умеет разрабатывать архитектуру основных блоков компилятора с использованием методов объектно-ориентированного проектирования.	Студент в целом умеет разрабатывать архитектуру основных блоков компилятора, однако к построенной архитектуре имеются замечания	Студент испытывает значительные затруднения при разработке архитектуры основных блоков компилятора.	Студент не умеет разрабатывать архитектуру основных блоков компилятора.

	<p>ИОПК-2.2. Проявляет навыки использования основных языков программирования, основных методов разработки программ, стандартов оформления программной документации.</p>	<p>ОР-ОПК2-2.2.1. Уметь реализовывать структуры данных, необходимых для работы компилятора, на выбранном языке программирования.</p>	<p>Студент умеет реализовывать структуры данных, необходимых для работы компилятора, на выбранном языке программирования.</p>	<p>Студент умеет реализовывать большую часть структур данных, необходимых для работы компилятора.</p>	<p>Студент испытывает значительные затруднения при реализации структуры данных, необходимых для работы компилятора.</p>	<p>Студент не умеет реализовывать структуры данных, необходимых для работы компилятора.</p>
	<p>ИОПК-2.3. Демонстрирует умение отбора среди существующих математических методов, наиболее подходящих для решения конкретной прикладной задачи.</p>	<p>ОР-ОПК2-2.3.1. Знать основные методы лексического и синтаксического анализа и оптимизации программ. Уметь выбирать наиболее эффективный метод синтаксического анализа для заданной формальной грамматики.</p>	<p>Студент знает основные методы лексического и синтаксического анализа и оптимизации программ, и умеет выбирать наиболее эффективный метод синтаксического анализа для заданной формальной грамматики.</p>	<p>Студент знает основные методы лексического и синтаксического анализа и оптимизации программ, но испытывает затруднения при выборе наиболее эффективного метода синтаксического анализа.</p>	<p>Студент имеет поверхностное представление об основных методах лексического и синтаксического анализа, и не умеет выбирать наиболее эффективный метод синтаксического анализа.</p>	<p>Студент не знает основных методов лексического и синтаксического анализа и оптимизации программ, не умеет выбирать наиболее эффективный метод синтаксического анализа.</p>

	ИОПК-2.4. Демонстрирует умение адаптировать существующие математические методы для решения конкретной прикладной задачи.	ОР-ОПК2-2.4.1. Уметь составлять грамматику для описания формального языка. Уметь модифицировать грамматику для возможности применения определённого метода синтаксического анализа.	Студент способен без ошибок составить формальную грамматику, которая описывает требуемый язык, а также модифицировать грамматику для возможности применения определённого метода синтаксического анализа предложений в языке.	Студент способен без ошибок составить формальную грамматику, которая описывает требуемый язык, но испытывает трудности при выполнении преобразований грамматики для возможности применения определённого метода синтаксического анализа.	Студент допускает некоторые ошибки при составлении формальной грамматики и её преобразовании и для возможности применения определённого метода синтаксического анализа.	Студент не способен без ошибок составить формальную грамматику и модифицировать её для возможности применения определённого метода синтаксического анализа.
ПК-2. Способен формализовать и алгоритмизировать поставленную задачу, написать программный код, а также верифицировать работоспособность программного обеспечения и исправить дефекты.	ИПК-2.1. Осуществляет построение формальной модели и алгоритма для поставленной задачи, написание программного кода с использованием языков программирования, верификацию работоспособности программного обеспечения и исправление дефектов.	ОР-ПК2-2.1.1. Владеть практическими навыками реализации блоков компилятора на выбранном языке программирования.	Студент способен без ошибок реализовать основные блоки компилятора: лексический и синтаксический анализаторы.	Студент способен реализовать основные блоки компилятора, однако в процессе реализации допускает некоторые ошибки.	Студент испытывает серьёзные затруднения при реализации основных блоков компилятора.	Студент не способен реализовать основные блоки компилятора.

	<p>ИПК-2.2. Осуществляет оформление программного кода в соответствии с установленными требованиями, разработку процедур верификации работоспособности и измерения характеристик программного обеспечения, разработку тестовых наборов данных.</p>	<p>ОР-ПК2-2.2.1. Владеть методами доказательства свойств языков, описываемых формальными грамматиками. Уметь строить тесты, обнаруживающие ошибки в реализации компилятора.</p>	<p>Студент владеет методами доказательства свойств языков, описываемых формальными грамматиками, и умеет строить тесты, обнаруживающие ошибки в реализации компилятора.</p>	<p>Студент владеет методами доказательства свойств языков, описываемых формальными грамматиками, однако построенные тесты не способны обнаружить некоторые ошибки.</p>	<p>Студент испытывает затруднения при доказательстве свойств языков, описываемых формальными грамматиками, построенные тесты не способны обнаружить большую часть ошибок.</p>	<p>Студент не владеет методами доказательства свойств языков, описываемых формальными грамматиками, не умеет строить тесты, обнаруживающие ошибки в реализации компилятора.</p>
	<p>ИПК-2.3. Осуществляет работу с системой контроля версий, рефакторинг и оптимизацию программного кода.</p>	<p>ОР-ПК2-2.3.1. Уметь эффективно реализовывать наиболее критичные к производительности части компилятора.</p>	<p>Студент умеет эффективно реализовывать наиболее критичные к производительности части компилятора.</p>	<p>Студент в целом умеет реализовывать наиболее критичные к производительности части компилятора, однако их эффективность может быть недостаточно высокой.</p>	<p>Студент испытывает значительные затруднения при реализации критичных к производительности частей компилятора.</p>	<p>Студент не умеет эффективно реализовывать наиболее критичные к производительности части компилятора.</p>

2. Этапы формирования компетенций и виды оценочных средств

№	Этапы формирования компетенций (разделы дисциплины)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Разделы 1, 2, 3	ОР-2, ОР-3	Лабораторные задания
2.	Разделы 1-5	ОР-1, ОР-2, ОР-3	Вопросы на зачёте

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

3.1. Типовые задания для проведения текущего контроля успеваемости по дисциплине: опрос на лекциях, проверка лабораторных работ.

3.2. Типовые задания для проведения промежуточной аттестации по дисциплине (вопросы к экзамену):

1. Задачи и этапы трансляции. Типы трансляторов.
2. Формальные грамматики и языки: грамматика, вывод, язык, разбор. Метаязык БНФ.
3. Способы задания выводов и разборов. Деревья вывода/разбора.
4. Классификация грамматик и языков по Хомскому.
5. Две стратегии распознавания контекстно-свободных языков. Неформальное описание нисходящей стратегии.
6. Две стратегии распознавания контекстно-свободных языков. Неформальное описание восходящей стратегии.
7. Допустимые преобразования контекстно-свободных грамматик. Типы контекстно-свободных грамматик.
8. Необходимое и достаточное условие порождения грамматикой бесконечного языка.
9. Преобразование укорачивающей контекстно-свободной грамматики в контекстно-свободную.
10. Назначение сканера, две стратегии использования. Теорема о преобразовании линейной грамматики в автоматную.
11. Диаграмма состояний лексем. Построение диаграммы состояний по правилам грамматики класса 3. Пример для лексемы "текстовая константа"
12. Диаграмма состояний лексем. Построение диаграммы состояний по правилам грамматики класса 3. Пример для лексемы "целочисленная константа"
13. Блок лексического анализа: дескрипторы, лексическая свёртка, таблицы.
14. Блок лексического анализа: задачи, структура, результат работы.
15. Метод простого предшествования: определение грамматики предшествования.
16. Метод простого предшествования: алгоритм работы распознавателя.
17. Метод простого предшествования: определение множеств $L(U)$ и $R(U)$, алгоритм их построения.
18. Метод простого предшествования: построение матрицы предшествования.
19. Метод простого предшествования: функции предшествования. Алгоритм Флойда построения функций предшествования.

20. Метод простого предшествования: функции предшествования. Построение функций предшествования по графу линейаризации.
21. Метод простого предшествования: структура транслятора.
22. Метод операторного предшествования: определение грамматики.
23. Метод операторного предшествования: первичная фраза, работа распознавателя.
24. Метод операторного предшествования: определение множеств $Lt(U)$ и $Rt(U)$, алгоритм их построения.
25. Метод операторного предшествования: матрица операторного предшествования, функции операторного предшествования.
26. LR(k)-грамматики. Определение.
27. LR(1)-распознаватель, его структура и работа.
28. LL(k)-грамматики. Определение. Иллюстрация на деревьях вывода.
29. 1-предсказывающий алгоритм разбора: структура и работа.
30. Метод рекурсивного спуска: структура анализатора и его работа.
31. Факторизация грамматики. Расширенная БНФ. Примеры.
32. Метод Кока-Янгера-Касами: алгоритм построения разбора.
33. Метод Кока-Янгера-Касами: алгоритм формирования левого вывода.
34. Алгоритм преобразования контекстно-свободной грамматики к грамматике в нормальной форме Хомского.
35. Определение идентификации.
36. Определяющее и использующее вхождение лексем.
37. Простейший способ реализации идентификации.
38. Ошибки, обнаруживаемые при идентификации.
39. Определение атрибутной индукции. Пример.
40. Эквивалентные и корректные преобразования. Оптимизирующие преобразования.
41. Оптимизация на линейных участках: свёртка констант.
42. Оптимизация на линейных участках: удаление лишних операций.

4. Методические материалы, определяющие процедуры оценивания образовательных результатов обучения

4.1. Методические материалы для оценки текущего контроля успеваемости по дисциплине.

Допуск к зачёту с оценкой осуществляется только при условии выполнения всех лабораторных работ.

4.2. Методические материалы для проведения промежуточной аттестации по дисциплине.

Каждый билет состоит из двух вопросов. Первый вопрос посвящён разделам 1,2 и 4, 5 данного курса (вопросы 1-14, 34-42). Второй вопрос посвящён разделу 3 (15-33) и обязательно содержит пример, на котором студент должен продемонстрировать работу метода из второго вопроса.