

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ:

Директор института прикладной
математики и компьютерных наук

А. В. Замятин



« 14 » _____ 2023г.

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

Методы компиляции

по направлению подготовки / специальности

02.03.03 Математическое обеспечение и администрирование информационных систем

Направленность (профиль) подготовки :


DevOps-инженерия в администрировании инфраструктуры ИТ-разработки

ОС составил(и):

ассистент кафедры компьютерной безопасности  В.А. Провкин

Рецензент:

канд. техн. наук, профессор,

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 №2

Председатель УМК ИПМКН,  С.П. Сущенко
д-р техн. наук, профессор

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

ОС разрабатываются в соответствии с рабочей программой (РП).

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
<p>●ПК-2. Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности</p>	<p>И●ПК-2.3 Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности. И●ПК-2.2 Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий.</p>	<p>●Р-2.3.1: знать теорию формальных грамматик, основные этапы и методы компиляции. ●Р-2.2.1: умеет использовать фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий.</p>	<p>Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности. Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий. Студент знает и понимает основы теории</p>	<p>Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности, использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий, но допускает неочности. Студент в целом знает основы теории формальных грамматик и</p>	<p>Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности, использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий, но допускает ошибки. Студент знает только некоторые</p>	<p>Не разрабатывает алгоритмы и программы при решении задач профессиональной деятельности, не использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий. Студент не знает и не понимает основы теории формальных грамматик и языков.</p>

<p>●ПК-5 – Способен установить и сопровождать программное обеспечение информационных систем и баз данных, в том числе отечественного производства.</p>	<p>И●ПК-5.3 Выполняет работы по настройке, администрированию и проверке работоспособности программного и аппаратного обеспечения при решении задач профессиональной деятельности</p>	<p>●Р-5.3.1: уметь использовать полученные знания при разработке блоков компилятора с языков высокого уровня и макроязыков.</p>	<p>формальных грамматик и языков, понимает основные этапы трансляции.</p> <p>Выполняет работы по настройке, администрированию и проверке работоспособности программного и аппаратного обеспечения при решении задач профессиональной деятельности при решении задач профессиональной деятельности</p> <p>Студент умеет использовать полученные знания при разработке блоков компилятора.</p>	<p>языков, однако его знания систематизированы не полностью.</p> <p>Выполняет работы по настройке, администрированию и проверке работоспособности программного и аппаратного обеспечения при решении задач профессиональной деятельности с неочностями</p> <p>Студенту требуется некоторая помощь для использования знаний при разработке блоков компилятора.</p>	<p>отдельные положения теории формальных грамматик и языков.</p> <p>Выполняет работы по настройке, администрированию и проверке работоспособности программного и аппаратного обеспечения при решении задач профессиональной деятельности с ошибками</p> <p>Студент испытывает серьезные затруднения при использовании знаний для разработки блоков компилятора.</p>	<p>Не выполняет работы по настройке, администрированию и проверке работоспособности программного и аппаратного обеспечения при решении задач профессиональной деятельности</p> <p>Студент не умеет использовать полученные знания при разработке блоков 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.	Тема 1. Элементы теории формальных грамматик	ОП-2.3.1, ОП-2.2.1, ОП-5.3.1	Лабораторные задания, вопросы
2.	Тема 2. Лексический анализ		Лабораторные задания, вопросы
3.	Тема 3. Методы детерминированный синтаксического анализа		Лабораторные задания, вопросы
4.	Тема 4. Атрибутивный анализ		Лабораторные задания, вопросы
5.	Тема 5. Оптимизация программ		Лабораторные задания, вопросы
6.	Тема 6. Генерация кода		Лабораторные задания, вопросы

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 линеаризации.

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 $L_t(U)$ и $R_t(U)$, 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.

Примеры задач:

1. Показать работу LR(1)-анализатора на примере разбора строки $!(a+b)*(b+c)!$

2. Показать работу метода рекурсивного спуска на примере разбора строки $!a+b*(b+a)!$

3. Показать работу метода простого предшествования на примере разбора строки $!a*b+b*a!$

4. Показать работу l-предсказывающего нисходящего анализатора на примере разбора строки $!a*(b*(a+c)+b)!$

5. Показать работу метода Кока-Янгера-Касами на примере разбора строки $!(a+b)*(a+c)!$

6. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:

```
a:=x; b:=5; b:=b+1;
```

```
c:=b-3; d:=b+c*(b+1);
```

```
b:= z ;
```

```
z:=(x+d)*(b-3);
```

7. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:

```
a:=0; c:=a-3;
```

```
d:=a+c-1;
```

```
b:=(a-3)*c+3*d;
```

```
a:=x; r:=(d+1)/(a-3);
```

8. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:

```
d:=2; c:=d-2;
```

```
b:=d*d*c; x:=d*(b-3)*(d-2);
```

```
d:=y; z:=(d-x)/(d-2);
```

9. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:

```
b:=3; c:=5;
```

```
a:=(b-c)*(c+b); d:=a+c;
```

```
a:=x; b:=(c+b)/(a+c);
```

10. Выполнить оптимизацию (свёртка констант и удаление лишних операций) для следующей программы:

```
a:=10; c:=a*a; d:=a*(c-3);
```

$$b:=d*d/2; a:=x;$$

$$d:=(a*a)/(c-3);$$

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.