

Энгельсский технологический институт (филиал) федерального  
государственного бюджетного образовательного учреждения  
высшего образования  
«Саратовский государственный технический университет  
имени Гагарина Ю.А.»

Кафедра «Естественные и математические науки»

**Оценочные материалы по дисциплине**

Б.1.1.26 «Теория языков программирования и методы трансляции»

направления подготовки  
09.03.04 «Программная инженерия»

профиль «Управление разработкой программных проектов»

# 1. Перечень компетенций и уровни их сформированности по дисциплинам (модулям), практикам в процессе освоения ОПОП ВО

В процессе освоения образовательной программы у обучающегося в ходе изучения дисциплины «Теория языков программирования и методы трансляции» должны сформироваться компетенции: ОПК-2.

## Критерии определения сформированности компетенций на различных уровнях их формирования

Индекс компетенции	Содержание компетенции
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности

Код и наименование индикатора достижения компетенции	Виды занятий для формирования компетенции	Оценочные средства для оценки уровня сформированности компетенции
<b>ИД- 4</b> опк 2 Умеет использовать современные информационные технологии и программные средства, в том числе отечественного производства, для задач трансляции с языков высокого уровня	Лекции, практические занятия, самостоятельная работа	Вопросы для устного опроса Задания для практических работ. Вопросы к зачету Контрольная работа для студентов заочной формы обучения

## Уровни освоения компетенции

Уровень освоения компетенции	Критерии оценивания
Продвинутый (отлично)	<p><b>Знать:</b> теоретические основы методов проектирования и способы описания языков программирования, основные положения теории формальных грамматик и языков, методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования.</p> <p><b>Уметь:</b> самостоятельно выполнять формальное описание синтаксиса и семантики, несложных процедурно-ориентированных и проблемно-ориентированных языков программирования, разрабатывать алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик, пользоваться стандартными терминами и определениями, пользоваться литературой для самостоятельного решения научно - исследовательских задач, связанных с разработкой языков и реализацией систем программирования.</p> <p><b>Владеть:</b></p>

	представлением о перспективных направлениях работ и методических подходах в области формальных методов описания и введения стандартов, используемых для описания языков программирования.
Повышенный (хорошо)	<p><b>Знать:</b> на достаточном хорошем уровне теоретические основы методов проектирования и способы описания языков программирования, основные положения теории формальных грамматик и языков, методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования.</p> <p><b>Уметь:</b> на хорошем уровне выполнять формальное описание синтаксиса и семантики, несложных процедурно-ориентированных и проблемно-ориентированных языков программирования, разрабатывать средней сложности алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик, уместно пользоваться стандартными терминами и определениями.</p> <p><b>Владеть:</b> не в полной мере представлением о перспективных направлениях работ и методических подходах в области формальных методов описания и введения стандартов, используемых для описания языков программирования.</p>
Пороговый (базовый) (удовлетворительно)	<p><b>Знать:</b> с трудом объясняет теоретические основы методов проектирования и способы описания языков программирования, основные положения теории формальных грамматик и языков, методы синтаксического анализа и перевода для класса формальных языков, используемых для описания основных конструкций языков программирования, стандарты, используемые для языков программирования.</p> <p><b>Уметь:</b> на низком уровне, допуская ошибки выполнять формальное описание синтаксиса и семантики, несложных процедурно-ориентированных и проблемно-ориентированных языков программирования, разрабатывать простые алгоритмы, реализующие методы синтаксического анализа и перевода для наиболее часто используемых классов формальных грамматик, пользоваться стандартными терминами и определениями.</p> <p><b>Владеть:</b> частично представлением о перспективных направлениях работ и методических подходах в области формальных методов описания и введения стандартов, используемых для описания языков программирования</p>

## 2. Методические, оценочные материалы и средства, определяющие процедуры оценивания сформированности компетенций (элементов компетенций) в процессе освоения ОПОП ВО

### 2.1 Оценочные средства для текущего контроля

#### Вопросы для устного опроса

##### Тема 1. Формальные языки и грамматики

1. Какие операции можно выполнять над цепочками символов?
2. Какие существуют методы задания языков?
3. Что такое грамматика языка?
4. Как выглядит описание грамматики в форме Бэкуса-Наура? Какие ещё формы описания

грамматик существуют?

5. На основе какого принципа классифицируются грамматики в классификации Н. Хомского?
6. Какие типы грамматик выделяют по классификации Н. Хомского?
7. Какие типы языков выделяют по классификации Н. Хомского? Как классификация языков соотносится с классификацией грамматик?
8. Что такое сентенциальная форма грамматики?
9. Что такое левосторонний и правосторонний выводы?
10. Для чего необходимо выполнять приведение грамматик?

## **Тема 2. Распознаватели. Преобразование грамматик.**

1. Какие грамматики относятся к регулярным? Назовите два класса регулярных грамматик.
2. Можно ли граф переходов конечного автомата использовать для однозначного определения данного автомата?
3. Всегда ли недетерминированный КА может быть преобразован к детерминированному виду?
4. Всякая ли регулярная грамматика является однозначной?
5. Если язык задан КА, то можно ли для него построить регулярное выражение?
6. Если язык задан КА, то может ли он быть задан КС-грамматикой?

## **Тема 3. Алгоритмы синтаксического разбора.**

1. На каком алгоритме основана работа распознавателя для LL(k)-грамматик?
2. На каком алгоритме основана работа распознавателя для LR(k)-грамматик?
3. В чём отличие метода нисходящего синтаксического анализа от метода восходящего анализа?
4. В каких случаях алгоритм для LALR(1)-грамматик не применим?
5. Как определяются отношения предшествования? Как они используются при выполнении синтаксического анализа?
6. Опишите работу алгоритма синтаксического анализа для грамматик операторного предшествования.
7. Какая грамматика называется основной?
8. Как организуется совместная работа Flex и Bison?
9. Расскажите о структуре программы на языке Bison.
10. Как описывается грамматика на языке Bison? Приведите пример.

## **Тема 4. Трансляторы. Таблицы идентификаторов.**

1. Перечислите основные этапы и фазы компиляции.
2. Верно ли, что любой компилятор является транслятором? А наоборот?
3. Что такое интерпретатор? В чём его отличие от транслятора и компилятора?
4. От чего зависит количество проходов, необходимых компилятору для построения результирующей объектной программы на основе исходной программы?
5. Что такое система программирования? Перечислите основные структурные элементы таких систем.
6. Какая информация может храниться в таблице идентификаторов?
7. Исходя из каких характеристик оценивается эффективность того или иного метода организации таблицы идентификаторов?
8. Какие существуют методы организации таблиц идентификаторов?
9. Что такое коллизия? Почему она происходит при использовании хэшфункций для организации таблиц идентификаторов?
10. В чём заключаются преимущества и недостатки метода цепочек по сравнению с методом рехэширования?

## **Тема 5. Лексический анализ.**

1. Чем различаются таблица лексем и таблица идентификаторов? В какую из этих таблиц лексический анализатор должен помещать ключевые слова, разделители и знаки операций?
2. Какую роль выполняет лексический анализ в процессе компиляции?

3. Как связаны лексический и синтаксический анализ?
4. Какие проблемы необходимо решить при построении лексического анализатора на основе конечного автомата?
5. Назовите этапы трансляции программы.
6. Что такое лексема?
7. В чём состоят задачи лексического анализа?
8. Дайте определение метаязыка.
9. Исходные данные для сканера

### Тема 6. Синтаксический анализ.

1. Какие способы адресации используются в системе команд?
2. Какие специфические подзадачи включает генерация кода ?
3. Какую роль выполняет синтаксический анализ в процессе компиляции?
4. Какие проблемы возникают при построении синтаксического анализатора и как они могут быть решены?
5. Какие типы грамматик существуют? Что такое КС-грамматики? Расскажите об их использовании в компиляторе.
6. Какие типы распознавателей для КС-грамматик существуют? Расскажите о недостатках и преимуществах различных типов распознавателей.
7. Поясните правила построения дерева вывода грамматики.
8. Что такое грамматики простого предшествования?
9. Как вычисляются отношения предшествования для грамматик простого предшествования?
10. Что такое грамматика операторного предшествования?
11. Как вычисляются отношения для грамматик операторного предшествования?
12. Что такое сдвиг, свёртка? Для чего необходим алгоритм «сдвиг-свёртка»?
13. Расскажите, как работает алгоритм «сдвиг-свёртка» в общем случае (с возвратами).
14. Как работает алгоритм «сдвиг-свёртка» без возвратов (объясните на своем примере)?

## Практические задания

### Тема 1. Формальные языки, грамматики и их свойства.

1. Дана грамматика. Построить вывод заданной цепочки.

a)  $S \rightarrow T \mid T+S \mid T-S$

$T \rightarrow F \mid F*T$

$F \rightarrow a \mid b$

Цепочка  $a-b*a+b$

b)  $S \rightarrow aSBC \mid abC$

$CB \rightarrow BC$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

Цепочка  $aaabbbccc$

2. Построить все сентенциальные формы для грамматики с правилами:

$S \rightarrow A+B \mid B+A$

$A \rightarrow a$

$B \rightarrow b$

3. Какой язык порождается грамматикой с правилами:

- а)  $S \rightarrow aaCFD$       б)  $S \rightarrow aQb \mid accb$       в)  $S \rightarrow A\perp \mid B\perp$       г)  $S \rightarrow 1B$   
 $AD \rightarrow D$        $Q \rightarrow cSc$        $A \rightarrow a \mid Ba$        $B \rightarrow B0 \mid 1$   
 $F \rightarrow AFB \mid AB$        $B \rightarrow b \mid Bb \mid Ab$   
 $Cb \rightarrow bC$   
 $AB \rightarrow bBA$   
 $CB \rightarrow C$   
 $Ab \rightarrow bA$   
 $bCD \rightarrow \varepsilon$

4. Построить грамматику, порождающую язык:

- а)  $L = \{a^n b^m c^k \mid n, m, k \geq 0\}$   
б)  $L = \{(ab)^n (cb)^m \mid n, m \geq 0\}$   
в)  $L = \{0^n (10)^m \mid n, m \geq 0\}$   
г)  $L = \{(ac)^n \mid n \geq 0, a \in \{b, d\}, c \in \{+, -\}\}$   
д)  $L = \{\perp(010)^n \perp \mid n > 0\}$   
е)  $L = \{a_1 a_2 \dots a_n a_1 \mid a_i \in \{0, 1\}\}$   
ж)  $L = \{a_1 a_2 \dots a_n a_1 a_2 \dots a_n \mid a_i \in \{c, d\}\}$   
з)  $L = \{\omega c \omega \mid \omega \in \{a, b\}^+\}$

5. К какому типу по Хомскому относится грамматика с правилами:

- а)  $S \rightarrow a \mid Ba$       б)  $S \rightarrow Ab$   
 $B \rightarrow Bb \mid b$        $A \rightarrow Aa \mid ba$   
в)  $S \rightarrow 0A1 \mid 01$       г)  $S \rightarrow AB$   
 $0A \rightarrow 00A1$        $AB \rightarrow BA$   
 $A \rightarrow 01$        $A \rightarrow a$   
 $B \rightarrow b$

6.

Пусть  $A = \{ab, c\}$  и  $B = \{c, ca\}$  - два формальных языка над алфавитом  $\{a, b, c\}$ .  
Найти следующие формальные языки:

1.  $A \cup B$ ;
2.  $A \setminus B$ ;
3.  $A^2$ ;
4.  $A^2 \setminus B^2$ ;
5.  $AB$ .

7.

Для языка  $L = \{x \in \{a, b\}^* \mid |x|_a - \text{четное}, |x|_b - \text{нечетное}\}$  постройте

1. Детерминированный конечный автомат;
2. По нему - регулярное выражение;
3. По этому выражению - грамматику;
4. По полученной грамматике перейдите по GN-теореме к N- автомату.

## Тема 2. Распознаватели. Преобразование грамматик

1. Принадлежит ли цепочка  $x = ((\ ))(\ ))$  языку, порождаемому грамматикой с правилами:

$$S \rightarrow SA|A$$

$$A \rightarrow (S) \underline{|} ()$$

2. Принадлежит ли цепочка  $x = 00011011$  языку, порождаемому грамматикой с правилами:

$$S \rightarrow SS|A$$

$$A \rightarrow 0A1|S|01$$

3. Принадлежит ли цепочка  $x = 0111000$  языку, порождаемому грамматикой с правилами:

$$S \rightarrow A0B|B1A$$

$$A \rightarrow BB|0$$

$$B \rightarrow AA|1$$

4. Верно ли соотношение  $a^*cb^* \in L(G)$  для следующей грамматики  $G$ ?

$$S \rightarrow \underline{Bab|aDa}; A \rightarrow \underline{Dc|cA}; B \rightarrow \underline{Sb|b};$$

$$D \rightarrow \underline{AB|aD}.$$

5. Верно ли соотношение  $ab^*c^* \in L(G)$  для следующей грамматики  $G$ ?

$$S \rightarrow SAS|A; A \rightarrow Ac|Da|b; B \rightarrow \underline{DaD};$$

$$D \rightarrow ABD|AB.$$

6. Пусть грамматика  $G$  определяется правилами

$$S \rightarrow AB; AB \rightarrow \underline{CBb}; CB \rightarrow ABb;$$

$$A \rightarrow \underline{a}; \underline{aB} \rightarrow a:$$

Какому классу (по Хомскому) она принадлежит? Порождается ли  $L(G)$  грамматикой более узкого класса?

7. Пусть грамматика  $G$  определяется правилами

$$S \rightarrow \underline{aAbB}; \underline{AbB} \rightarrow \underline{aAbB}; \underline{bBb} \rightarrow \underline{bb}; A \rightarrow \varepsilon.$$

Какому классу (по Хомскому) она принадлежит? Порождается ли  $L(G)$  грамматикой более узкого класса?

8. Какому классу по Хомскому принадлежит:

а) Грамматика с правилами:

$$S \rightarrow \underline{AS|\varepsilon}; A \rightarrow \underline{a|b}:$$

б) Язык, порождаемый этой грамматикой?

9. Построить праволинейные грамматики для языков, состоящих из:

а) идентификаторов произвольной длины, начинающихся с буквы;

б) идентификаторов, содержащих от 1 до 6 символов и начинающихся с букв I, J, K, L, M, N;

в) вещественных констант;

г) всех цепочек из нулей и единиц, имеющих:

- четное число нулей и четное число единиц;

- либо нечетное число нулей и нечетное число единиц.

10. Построить КС-грамматики для следующих языков:

а)  $\{0^n 1^n : n \geq 1\}$

б)  $\{ww^R : w \in \{a, b\}^*\}$

в) Все цепочки из нулей и единиц с одинаковым числом тех и других

г)  $\{\{a, b\}^* \setminus \{a^m b^n a^m b^n : m, n \geq 1\}\}$

д)  $\{\{a, b\}^* \setminus \{a^{2m} b^{3n} a^{2m} b^n : m, n \geq 1\}\}$

е)  $\{\{a, b\}^* \setminus \{a^m b^n a^m : m, n \geq 1\}\}$

ж)  $\{\{a, b\}^* \setminus \{ww : w \in \{a, b\}^*\}\}$

з)  $\{\{a, b\}^* \setminus \{a^n b^n a^n : n \geq 1\}\}$

11. Пусть G - грамматика с правилами:

$$\begin{array}{lll} S \rightarrow CD & C \rightarrow aCA | bCB | \varepsilon & AD \rightarrow aD \\ BD \rightarrow bD & Aa \rightarrow aA & Ab \rightarrow bA \\ Va \rightarrow aV & Bb \rightarrow bB & D \rightarrow \varepsilon \end{array}$$

Показать, что  $L(G) = \{xx|x \in \{a, b\}^*\}$ .

12. Построить грамматику, порождающую данный язык:

$$\{a^n c b^n a^n c b^n | n > 0\}:$$

### Тема 3. Алгоритмы синтаксического разбора

1. Построить множества FIRST и FOLLOW для каждого нетерминала грамматики

а)  $S \rightarrow aAB | B$

$A \rightarrow aA | a$

$B \rightarrow BS | A | b;$

б)  $S \rightarrow aAB | BA$

$A \rightarrow BBB | a$

$B \rightarrow AS | b;$

в)  $S \rightarrow S + T$

$S \rightarrow T$

$T \rightarrow a$

$T \rightarrow S[S];$

г)  $S \rightarrow ABC$

$A \rightarrow BB | \varepsilon$

$B \rightarrow CC | a$

$C \rightarrow AA | b;$

д)  $S \rightarrow aB | bA$

$A \rightarrow aS | bAA | a$

$B \rightarrow bS | aBB | b;$

е)  $S \rightarrow Ba | Ab$

$A \rightarrow Sa | AAb | a$

$B \rightarrow Sb | BBa | b;$

ж)  $S \rightarrow (SbS)$

$S \rightarrow (T)$

$S \rightarrow a$

$T \rightarrow TS$

$T \rightarrow S;$

$S \rightarrow B;$

з)  $B \rightarrow \text{begin } D; S \text{ end}$

$B \rightarrow \varepsilon$

$D \rightarrow D; d$

$D \rightarrow d$

$S \rightarrow S; B$

и)  $A \rightarrow aACd | b$

$C \rightarrow c | \varepsilon.$

2. Для грамматики написать эквивалентную LL(1)-грамматику



- а)  $S \rightarrow \underline{aS|a};$   
 б)  $S \rightarrow \underline{ba|A}$                        $A \rightarrow \underline{a|Aab|Ab};$   
 в)  $S \rightarrow \underline{aaS|abA}$                        $A \rightarrow \underline{\varepsilon|Aa|Ab};$   
 г)  $S \rightarrow \underline{baaA|babA}$                        $A \rightarrow \underline{\varepsilon|Aa|Ab};$   
 д)  $S \rightarrow \underline{abaA|abbA}$                        $A \rightarrow \underline{\varepsilon|Aa|Ab};$   
 е)  $S \rightarrow \underline{ab|baA}$                        $A \rightarrow \underline{\varepsilon|Aab|Ab}.$

3. Для следующих грамматик определить, являются ли они LL(k) грамматиками и найти точное значение k. Для LL(1)-грамматик построить детерминированный левый анализатор:

- а)  $S \rightarrow \underline{aAS|b}$                        $A \rightarrow \underline{a|bSA};$   
 б)  $S \rightarrow A|B$                        $A \rightarrow aAb|0$                        $B \rightarrow aBbb|1;$   
 в)  $S \rightarrow \underline{\varepsilon|abA}$                        $A \rightarrow \underline{Saa|b};$   
 г)  $S \rightarrow \underline{aS|a};$   
 д)  $S \rightarrow \underline{aAaa|bAba}$                        $A \rightarrow b|\varepsilon;$   
 е)  $S \rightarrow \underline{Sa|b};$   
 ж)  $S \rightarrow TE';$                        $E' \rightarrow +TE'|\varepsilon$                        $T \rightarrow FT'$   
 $T' \rightarrow \underline{*FT'|\varepsilon}$                        $F \rightarrow (S)|a.$

4. Преобразовать грамматику к LL(1)-виду и построить для нее LL(1)-таблицу

- а)  $S \rightarrow Ab$                        $A \rightarrow \underline{aA|a};$   
 б)  $S \rightarrow \underline{aB}$                        $B \rightarrow \underline{aBB|b};$

#### Тема 4. Трансляторы. Таблицы идентификаторов.

Написать программу, которая получает на входе набор идентификаторов, организует таблицу по заданному методу и позволяет осуществить многократный поиск идентификатора в этой таблице. Список идентификаторов задан в виде текстового файла. Длина идентификаторов ограничена 32 символами.

#### Варианты заданий

№ варианта	Первый метод организации таблицы	Второй метод организации таблицы
1	Простое рехэширование	Метод цепочек
2	Простое рехэширование	Простой список
3	Простое рехэширование	Упорядоченный список
4	Простое рехэширование	Бинарное дерево
5	Рехэширование с помощью псевдослучайных чисел	Простое рехэширование
6	Рехэширование с помощью псевдослучайных чисел	Метод цепочек
7	Рехэширование с помощью псевдослучайных чисел	Простой список
8	Рехэширование с помощью псевдослучайных чисел	Упорядоченный список

9	Рехэширование с помощью псевдослучайных чисел	Рехэширование с помощью произведения
10	Рехэширование с помощью псевдослучайных чисел	Бинарное дерево

## Тема 5. Лексический анализ.

1. Заменить каждое из следующих выражений эквивалентным, в котором не используются знак "+":

- 1)  $(a + b)^*$ ;
- 2)  $(a + bb + ba)^*$ ;
- 3)  $(a + (bb + ab)^*)^*$ ;

2. Найти регулярные выражения, обозначающие языки, все слова которых - элементы множества  $\{0, 1\}^*$ :

- 1) оканчивающиеся на 011, 101, 110;
- 2) начинающиеся с 110, 101 или 011;
- 3) у которых каждый третий символ есть 0 или каждым второй - 1;
- 4) не содержащие ни одной из подстрок 011 и 101;
- 5) содержащие каждую из подстрок 011 и 101;
- 6) начинающиеся с 011 и оканчивающиеся на 110 или 101;
- 7) начинающиеся с 011 или 110 и оканчивающиеся на 101;
- 8) начинающиеся с 011 и содержащие вхождения подстроки 110;
- 9)  $\{01^n \mid n > 1\}$ ;
- 10)  $\{01^n 0 \mid n > 0\}$ ;
- 11)  $\{0^m 1^n \mid n, m > 1\}$ ;

3. Определить язык, состоящий из всех идентификаторов, с помощью:

- а) регулярного выражения;
- б) левосторонней грамматики;
- в) конечного автомата;
- г) правосторонней грамматики.

4. Построить правостороннюю грамматику, порождающую язык L всех слов в алфавите  $\{0, 1\}$ , содержащих четное число единиц и нечетное число нулей. Будет ли она однозначной?

5. Построить регулярное выражение для языка  $L^R$ , где L - язык всех слов в алфавите  $\{0, 1\}$ , содержащих четное число единиц и нечетное число нулей

## Тема 6. Синтаксический анализ

1. Пусть G - грамматика с правилами:

$$S \rightarrow \underline{SbS} \mid \underline{ScS} \mid a$$

Найти 2 различных дерева вывода для цепочки abаса.

2. Дана однозначная КС-грамматика  $G = (N, T, P, S)$  и цепочка  $w \in L(G)$ . Количество элементов во множествах N, T, P равно  $n_1, n_2, n_3$  соответственно, а  $|w| = l$ . Найти нижнюю и верхнюю границу для числа деревьев разбора w в G.

3. Построить МП-автоматы, определяющие языки

- а)  $\{ww^R : w \in \{a, b\}^*\}$ ;  
 б) язык всех цепочек из нулей и единиц с одинаковым числом тех и других  
 в)  $\{\{a, b\}^* \setminus \{a^m b^n a^m b^n\} : m, n \geq 1\}$ ;  
 г)  $\{\{a, b\}^* \setminus \{a^m b^n a^m\} : m, n \geq 1\}$ ;  
 д)  $\{\{a, b\}^* \setminus \{ww\} : w \in \{a, b\}^*\}$ ;

4. Построить автомат с магазинной памятью, допускающий язык:

- а)  $(\{a^n b^n c^m | n, m \geq 1\}) \cup (\{a^m b^n c^n | n, m \geq 1\})$ ;  
 б)  $\{a^n c^k b^n | k, n \geq 1\}$ ;  
 в)  $\{a^m b^n c^p | m + n + p \equiv 0 \pmod{2}, m, n, p \geq 0\}$ ;  
 г)  $\{a^p b^q c^r | p + q \geq r; p, q, r \geq 0\}$ ;  
 д)  $\{x | x \in \{a, b\}^*, |x|_a = |x|_b\}$ ;  
 е)  $\{x | x \in \{a, b\}^*, |x|_a \geq |x|_b\}$ ;

5. Построить детерминированные МП-автоматы, определяющие языки:

- а)  $\{wcw^R : w \in \{a, b\}^*\}$ ;  
 б)  $\{0^n 1^n : n \geq 1\}$   
 в)  $\{xcx^R ycy^R | x, y \in \{a, b\}^*\}$ .

6. Доказать, что для любой КС-грамматики  $G'$  существует эквивалентная ей КС грамматика  $G$ , имеющая лишь правила вида

$$A \rightarrow BC; A \rightarrow a, \text{ где } A, B, C \in V_N; a \in V_T.$$

## Тема 8. Генерация промежуточного и целевого кода

1. Для следующих операторов присваивания сгенерировать оптимальный код методом сопоставления образцов:

- а)  $a = b[i] + j$ ;                      б)  $a = b[i+5]$ ;                      в)  $a = b[i] + [2]$ ;  
 г)  $a = b[i+2+j]$ ;                      д)  $a = b[2+c[1]]$ ;                      е)  $a = b[i+j]$ ;  
 ж)  $a = b[i+2] + 3$ ;                      з)  $a = j + b[i+3]$ ;                      и)  $a = b[i+j+1]$ ;

2. Для следующих логических выражений сгенерировать код на командах перехода и изобразить атрибутивное дерево

- а)  $A \text{ and not } (B \text{ or } C) \text{ or } (D \text{ and } E)$ ;  
 б)  $A \text{ and } B \text{ and } C \text{ or not } (D \text{ or } E)$ ;  
 в)  $A \text{ and } (B \text{ or not } (C \text{ and } D) \text{ and } E)$ ;  
 г)  $\text{not } (A \text{ and } B \text{ or } C \text{ or } D) \text{ and } E$ ;  
 д)  $A \text{ and } B \text{ or } C \text{ or } D \text{ and not } E$ .

3. Для следующих арифметических выражений с помощью алгоритма Сети-Ульмана сгенерировать программу и изобразить атрибутивное дерево:

- а)  $A * B + C * (D + E) * F$ ;                      б)  $A * (B + C) * (D + E) * F$ ;  
 в)  $A + B + C * D + E * F$ ;                      г)  $A + B * C * D * E + F$ ;  
 д)  $A + B * (C * D + E * F)$ .

## Оценочные средства для промежуточного контроля

### Вопросы к зачету

- 1 Определение компилятора. Анализ, синтез, контекст компилятора. Фазы компиляции.
- 2 Обнаружение ошибок пример простого компилятора.
- 3 Контекстно-свободные грамматики. Их свойства.
- 4 Задачи лексического анализа. Токены, лексемы, шаблоны. Атрибуты токенов.
- 5 Буферизация ввода при лексическом анализе. Ограничители.
- 6 Операции над языками. Объединение, конкатенация, замыкания(Клини, позитивное).
- 7 Строки и языки. Регулярные выражения.
- 8 Регулярные определения. Нерегулярные множества.
- 9 Распознавание токенов. Диаграммы переходов.
- 10 Конечные автоматы. Детерминированные и недетерминированные автоматы.
- 11 Синтаксический анализ. Дерево синтаксического разбора. Неоднозначность деревьев разбора
- 12 Преобразование НКА в контекстно-свободную грамматику.
- 13 Левая рекурсия, левая факторизация. Их устранение.
- 14 СА методом рекурсивного спуска.
- 15 Нерекурсивный предиктивный анализ.
- 16 Среды времени исполнения. Связывание имен.
- 17 Классификация памяти времени исполнения. Записи активации.
- 18 Стратегии выделения памяти.
- 19 Последовательности вызовов и возвратов. Распределение памяти в куче.
- 20 Генерация промежуточного кода. Трехадресный код. Типы трехадресных инструкций.
- 21 Реализация трехадресных инструкций. Четверки, тройки, косвенные тройки.
- 22 Генерация кода. Целевые программы. Выбор инструкций. Распределение регистров.
- 23 Базовые блоки. Преобразования в базовых блоках. Графы потоков.
- 24 Вычисление последующих использований временных переменных. Простой генератор кода.
- 25 Дескрипторы регистров и адресов. Функция getreg.

Оценивание результатов обучения в форме уровня сформированности элементов компетенций проводится путем контроля во время промежуточной аттестации в форме зачета:

а) оценка «зачтено» – компетенция(и) или ее часть(и) сформированы на базовом уровне;

б) оценка «не зачтено» – компетенция(и) или ее часть(и) не сформированы.

Критерии, на основе которых выставляются оценки при проведении текущего контроля и промежуточной аттестации приведены в табл. 1.

Оценки «Не зачтено» ставятся также в случаях, если обучающийся не приступал к выполнению задания, а также при обнаружении следующих нарушений:

- списывание;
- плагиат;
- фальсификация данных и результатов работы.

Таблица 1 – Критерии выставления оценок при проведении текущего контроля и промежуточной аттестации

Шкала оценки	Оценка	Критерий выставления оценки
Двухбалльная шкала	Зачтено	Обучающийся ответил на теоретические вопросы. Показал знания в рамках учебного материала. Выполнил практические задания. Показал удовлетворительные умения и владения навыками применения полученных знаний и умений при решении задач в рамках учебного материала
	Не зачтено	Обучающиеся при ответе на теоретические вопросы и при выполнении практических заданий продемонстрировал недостаточный уровень знаний и умений при решении задач в рамках учебного материала. При ответах на дополнительные вопросы было допущено множество неправильных ответов

### 2.3. Итоговая диагностическая работа по дисциплине

#### ЗАДАНИЯ ДЛЯ ДИАГНОСТИЧЕСКОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «Теория языков программирования и методы трансляции»

Номер задания	Правильный ответ *	Содержание вопроса	Компетенция	Код и наименование индикатора достижения компетенции
1.	Счетное множество допустимых символов языка.	Что такое алфавит?	ОПК-2	<b>ИД- 4</b> опк 2 Умеет использовать современные информационные технологии и программные средства, в том числе отечественного производства, для задач трансляции с языков высокого уровня
2.	Лексема	Структурная единица языка («слово» заданного языка), состоящая из элементарных символов языка	ОПК-2	<b>ИД-4</b> опк-2
3.	а	Некоторое множество цепочек в алфавите называют а. язык б. порядок с. слово	ОПК-2	<b>ИД-4</b> опк-2
4.	с	Алфавит содержит перечисление 1. нетерминальных символов 2. классов лексем 3. атрибутов а. Верно только 1 б. Верно только 2 и 3 с. Верно все	ОПК-2	<b>ИД-4</b> опк-2
5.	Подстановка самого левого нетерминала	Левосторонним называется вывод, в котором в любой сентенциальной форме на каждом шаге производится...	ОПК-2	<b>ИД-4</b> опк-2

6.	Языки высокого уровня	Какие языки являются основным средством разработки программ?	ОПК-2	<b>ИД-4</b> ОПК-2
7.	b	К "универсальным" языкам относятся 1. Модула-2 2. Ада 3. Фортран a. Верно только 1 b. Верно только 1 и 2 c. Верно все	ОПК-2	<b>ИД-4</b> ОПК-2
8.	b	Пустой язык и язык, содержащий только пустую цепочку a. абсолютно идентичны друг другу b. два разных языка c. похожи	ОПК-2	<b>ИД-4</b> ОПК-2
9.	Поддерево, состоящее из некоторой дуги данного дерева, ее начальной и конечной вершин, а также всех вершин и дуг, лежащих на всех путях, выходящих из конечной вершины этой дуги.	Что такое ветвь?	ОПК-2	<b>ИД-4</b> ОПК-2
10.	Построение для программы на входном языке эквивалентной машинной программы	Что является задачей генератора кода?	ОПК-2	<b>ИД-4</b> ОПК-2
11.	Заголовки процедур, заголовки функций	Что содержит раздел библиотеки?	ОПК-2	<b>ИД-4</b> ОПК-2
12.	В виде простого массива строк фиксированной длины	В виде чего может быть организована таблица символов, если длина идентификатора ограничена?	ОПК-2	<b>ИД-4</b> ОПК-2
13.	a	По количеству, языков a) больше, чем конечных представлений b) столько же, сколько конечных представлений c) меньше, чем конечных представлений	ОПК-2	<b>ИД-4</b> ОПК-2
14.	Совокупность множества вершин и множества линейно упорядоченных	Что такое упорядоченный граф?	ОПК-2	<b>ИД-4</b> ОПК-2

	списков дуг			
15.	с	Синтаксическое дерево программы является а) самой сложной формой промежуточного представления б) неприменяемой формой промежуточного представления с) простейшей формой промежуточного представления	ОПК-2	<b>ИД-4</b> ОПК-2
16.	Язык, который допускается детерминированным автоматом с магазинной памятью	Какой язык называется детерминированным контекстно-свободным языком?	ОПК-2	<b>ИД-4</b> ОПК-2
17.	Переменные	Что отображает все многообразие механизмов доступа в языке?	ОПК-2	<b>ИД-4</b> ОПК-2
18.	а, б	Основными формализмами, лежащим в основе реализации лексических анализаторов, являются (возможны несколько вариантов ответа) а) конечные автоматы б) регулярные выражения с) бесконечные автоматы	ОПК-2	<b>ИД-4</b> ОПК-2
19.	Да	Возможно ли переворачивание цепочки?	ОПК-2	<b>ИД-4</b> ОПК-2
20.	Леворекурсивная грамматика	Как называется грамматика, в которой имеется такой нетерминал, что для некоторой цепочки существует левосторонний вывод нетерминала?	ОПК-2	<b>ИД-4</b> ОПК-2

### **Критерии и шкалы оценивания уровня сформированности компетенции**

Таблица – Критерии выставления оценок при итоговой диагностической работе по дисциплине по дисциплине «Структуры и алгоритмы обработки данных»

Шкала оценки	Оценка	Критерий выставления оценки
100-процентная шкала	Отлично	85-100 %% правильных ответов
	Хорошо	71-84 %% правильных ответов
	Удовлетворительно	50-70 %% правильных ответов
	Неудовлетворительно	менее 50 % правильных ответов