

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

**Паспорт
курсовой работы**

по дисциплине «Теория формальных языков и компиляторов», 6 семестр

1. Методика оценки.

В рамках курсовой работы по дисциплине студенты должны разработать основные элементы транслятора с заданного варианта учебного языка программирования: лексический анализатор, синтаксический анализатор, семантический анализатор, генератор объектного кода или интерпретатор псевдокода, результаты разработки описать в расчетно-пояснительной записке.

При выполнении курсовой работы студенты должны использовать учебный пакет автоматизации проектирования трансляторов «Вебтранслаб».

Обязательные структурные части курсовой работы.

- 1.1. Определение лексики и синтаксиса учебного языка программирования с использованием регулярных выражений и формальных грамматик.
- 1.2. Алгоритмы и программы преобразования текста транслируемой программы в последовательность токенов и совокупность информационных таблиц.
- 1.3. Алгоритмы и программы преобразования последовательности токенов в постфиксную форму записи (ПФЗ) или абстрактное синтаксическое дерево (АСД).
- 1.4. Алгоритмы и программы преобразования ПФЗ или АСД в псевдокод.
- 1.5. Алгоритмы и программы генератора объектного кода или интерпретатора псевдокода.

Оцениваемые позиции: 1.1, 1.2, 1.3, 1.4, 1.5.

2. Критерии оценки.

- работа считается **не выполненной**, если выполнено не более двух оцениваемых пунктов, расчетно-пояснительная записка выполнена с отклонениями от нормативов программно-технической документации, при защите студент не может дать ответов на самые простые вопросы, оценка составляет 0-49 баллов.
- работа считается выполненной **на пороговом** уровне, если выполнены пункты 1.1 – 1.3, расчетно-пояснительная записка содержит соответствующие им разделы, при защите студент затруднился с ответами на некоторые вопросы, оценка составляет 50-72 баллов.
- работа считается выполненной **на базовом** уровне, если выполнены пункты 1.1 – 1.4, расчетно-пояснительная записка содержит соответствующие им разделы, при защите студент ответил на вопросы с незначительными отклонениями от правильных ответов, оценка составляет 73-85 баллов.
- работа считается выполненной **на продвинутом** уровне, если выполнены все пункты задания, все алгоритмы описаны в расчетно-пояснительной записке в соответствии с нормативами программно-технической документации, при защите студент аргументировано ответил на все заданные вопросы, оценка составляет 86-100 баллов.

3. Шкала оценки.

Курсовая работа оценивается самостоятельной оценкой согласно правилам балльно-рейтинговой системы, приведенным в рабочей программе дисциплины.

4. Примерный перечень тем курсового проекта (работы).

Цель работы: практическое применение теоретических основ проектирования

трансляторов с языков программирования; освоение средств автоматизации построения трансляторов; разработка элементов транслятора для учебного языка.

I. Содержание работы

Раздел 1.

Описать лексику, синтаксис и семантику заданного варианта языка. Написать несколько простых тестовых программ, содержащих все заданные элементы и управляющие конструкции языка. Вручную выполнить интерпретацию программы. Эти программы использовать впоследствии для проверки элементов разрабатываемого транслятора.

Раздел 2.

- 2.1. Разработать систему регулярных выражений, определяющую лексику заданного варианта языка. Используя пакет Вебтранслаб, построить автоматную реализацию лексического акцептора на выбранном инструментальном языке. Добиться работоспособности акцептора.
- 2.2. Разработать совокупность действий, т.е. расширение системы регулярных выражений, реализующую:
 - а. - поиск в таблицах идентификаторов / констант и пополнение таблиц;
 - б. - доформирование лексем,
 - с. и построить лексический анализатор. Убедиться в правильности его работы.

Раздел 3.

- 3.1. Разработать формальную грамматику класса LL(1), определяющую синтаксис заданного языка. Используя пакет Вебтранслаб, построить какую-либо автоматную реализацию нисходящего синтаксического акцептора, добиться его работоспособности.

Или:

- 3.2. Разработать формальную грамматику класса не выше, чем LALR(1). Используя пакет Вебтранслаб, построить автоматную реализацию восходящего синтаксического акцептора, добиться его работоспособности.
- 3.3. Разработать расширение построенного синтаксического акцептора для преобразования входной последовательности лексем в постфиксную форму записи или в абстрактное синтаксическое дерево.

Раздел 4.

Разработать семантический анализатор и преобразователь ПФЗ или АСД в псевдокод. Формат псевдокода определяется последней цифрой варианта задания.

Раздел 5.

Разработать генератор объектного кода или интерпретатор псевдокода.

Оформить (в электронном виде) расчетно-пояснительную записку следующего содержания:

- Данное задание;
- Оглавление;
- Введение;
- Описание заданного варианта языка;
- Тестовая программа (программы).
- По каждому из пунктов 1, 2.1, 2.2:
 - система регулярных выражений (п.1) или формальная грамматика (п.п. 2.1, 2.2);
 - фрагменты управляющих таблиц конечных автоматов, построенные Вебтранслаб'ом. Описать функционирование автоматов по этим фрагментам.

- примеры результатов работы автоматов с тестовыми примерами; объяснение и анализ этих результатов.
- По каждому из пунктов 3.1, 3.2, 3.3 и 4:
 - полное описание разработанного алгоритма, тексты разработанных программ (в случае их большого объема включать в записку фрагменты, содержащие наиболее важные части алгоритма; описание алгоритма можно не включать, если программы приводятся полностью и имеют детальные комментарии).
 - примеры результатов работы компонент транслятора с правильными и ошибочными входными программами; объяснение и анализ этих результатов.
- Заключение.

Объем записки **не должен** превышать 25-30 страниц.

II Определение учебного языка, варианты заданий.

Лексика, синтаксис и семантика учебного языка основаны на языках типа Java и C#. Любые конструкции, не оговоренные явно в задании, можно определять самостоятельно.

Ключевые слова (в задании выделены жирным шрифтом, например: **long**, **when**, **case**, ...) должны быть нечувствительны к регистру. Обозначения:

- [...] – необязательная часть
- ... – часть, повторяющаяся произвольное количество раз
- < > – описание конструкции:

<Б>|<Ц>|<пБ>|<пЦ>|<пБЦ> – одна буква | одна цифра | непустая последовательность букв | непустая последовательность цифр | возможно пустая посл-ть букв и/или цифр

<И> – Имя переменной / объекта; <К> – Константа;

<В> – произвольное Выражение; <ЛВ> – Логическое Выражение;

<ОБ> – Оператор или Блок; <О> – одиночный оператор; <ОП> – оператор присваивания;

<Код> – код операции; <Оп> – имя операнда; <Рез> – имя результата

Расшифровка цифр варианта задания:

II.1 Идентификаторы и константы

Вариант:	1	2	3	4
Идентификаторы	<пБ>\$<Ц>	\$<пБ><пЦ>	<Б>_<пБЦ>	<Б><пБЦ><Б>
Константы	целые, в том числе 16-ричные вещественные символьные	целые, в том числе двоичные вещественные символьные	целые, в том числе 4-ричные вещественные символьные	целые, в том числе 8-ричные вещественные символьные

II.2 Объявления примитивных типов (целое, вещественное, символьное):

Вариант:	1	2	3	4
	[u]long double symbol	integer float char	[un]signed number sign	[u]int real litera

II.3 Объявления объектов

Вариант:	1	2	3	4
Классов	object	entity	class	bundle

II.4 Оператор присваивания:

Вариант:	1	2	3	4
	<И> <= <В>;	put <В> to <И>;	<И> := <В> ;	set(<И>, <В>);

II.5 Условный оператор:

Вариант:	1	2	3	4
	when <ЛВ> then <ОБ> [else <ОБ>]	by <ЛВ> do <ОБ> [else <ОБ>]	? (<ЛВ>) :<ОБ> [: <ОБ>]	if (<ЛВ>) <ОБ> [else <ОБ>]

II.6 Оператор цикла:

Вариант:	1	2	3	4
	for <И> from <К> to <К> [step <К>] exec <ОБ>	while (<ЛВ>) do loop <ОБ>	for ([<ОП>];[<ЛВ>]; [<О>]) <ОБ>	do <ОБ> until <ЛВ>

II.7 Оператор переключателя

Вариант:	1	2	3	4
	select <В> case (<К>) <ОБ> [escape]... end	with <В> {? <К>:<ОБ> [off]...}	case <В> { when <К> then <ОБ> [break] ... }	switch <В> { by <К> do <ОБ> [break] ... }

II.8 Вид псевдокода

Вариант:	1	2	3	4
	тетрады	триады <Код><Оп><Оп>	пентады	Триады <Код><Оп><Рез>

5. Примерный перечень вопросов к защите курсового проекта (работы).

1. Как задать одну группу слов несколькими разными регулярными выражениями?
2. Как включать действия в системы регулярных выражений?
3. Какие операции используются в языке регулярных выражений?
4. Что такое мультиавтоматный лексический акцептор и для чего он может понадобиться?
5. Что такое контекстно-свободная грамматика?
6. Что такое дерево грамматического разбора и каковы его свойства?
7. Как используются регулярные выражения в грамматиках?
8. Как определить пригодность грамматики к выбранному способу синтаксического анализа – нисходящему или восходящему?
9. Что такое расширение синтаксического акцептора?
10. Как включать действия в формальную грамматику?
11. Что такое отношения предшествования и последования, как и для чего они используются построителем синтаксических анализаторов?
12. Что такое аннулируемые, бесплодные и недостижимые нетерминалы грамматики, как информация о таких символах используется при преобразовании грамматики в конечный автомат со стековой памятью?
13. Какими свойствами обладает постфиксная форма записи?
14. Что такое абстрактное синтаксическое дерево?
15. Как дерево грамматического разбора преобразуется в абстрактное синтаксическое дерево?
16. Что такое множество выбора правила грамматики?
17. Что такое рекурсивный спуск?
18. Какие операции выполняет восходящий синтаксический анализатор?
19. Какие виды псевдокода могут быть использованы в трансляторах?
20. В чем состоит отличие компилятора от интерпретатора?
21. Как в трансляторах реализуется контроль типов данных?
22. Что такое ассоциация наименования объекта программы?
23. Что такое активация функции и вызывающая последовательность?