

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

Кафедра вычислительной техники
Кафедра автоматизированных систем управления

“УТВЕРЖДАЮ”
ДЕКАН АВТФ

к.т.н. Рева И. Л.

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Теория формальных языков и компиляторов

Образовательная программа: 09.03.01 Информатика и вычислительная техника, профиль:
Программное обеспечение компьютерных систем и сетей

Курс: 3, семестр : 6

Факультет автоматики и вычислительной техники,

		Семестр
№	Вид деятельности	6
1	Всего зачетных единиц (кредитов)	4
2	Всего часов	144
3	Всего занятий в контактной форме, час.	84
4	Лекции, час.	36
5	Практические занятия, час.	0
6	Лабораторные занятия, час.	36
7	из них в активной и интерактивной форме, час.	18
8	Аттестация, час.	2
9	Консультации, час.	10
10	Самостоятельная работа, час.	60
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	КР
12	Вид аттестации	Э

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 09.03.01 Информатика и вычислительная техника

ФГОС введен в действие приказом №5 от 12.01.2016 г. , дата утверждения: 09.02.2016 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 09.03.01 Информатика и вычислительная техника

Рабочая программа обсуждена на заседании кафедры

ВТ, протокол заседания кафедры №6 от 20.06.2017
АСУ, протокол заседания кафедры №7 от 20.06.2017

Утверждена на совете факультета автоматизации и вычислительной техники, протокол № 6 от 21.06.2017

Программу разработал:

доцент, к.т.н. Малявко А. А.

Заведующий кафедрой:

доцент, к.т.н. Якименко А. А.
профессор, д.т.н. Гриф М. Г.

Ответственный за образовательную программу:

доцент Романов Е. Л.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ОПК.5 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; в части следующих результатов обучения:	
з5. знать правовые основы информационной безопасности и принципы защиты авторского права на программные продукты	
у10. уметь использовать специализированные программные средства при решении профессиональных задач	
у11. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	
у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	
у8. владеть персональным компьютером как средством управления информацией	
Компетенция ФГОС: ПК.3 способность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности; в части следующих результатов обучения:	
з4. знать современные компиляторы, отладчики и оптимизаторы программного кода	
Компетенция НГТУ: ПК.10.В/ПТ готовность к разработке компонентов аппаратно-программных комплексов и баз данных с использованием современных инструментальных средств и технологий программирования; в части следующих результатов обучения:	
у8. уметь сравнивать и выбирать суперкомпьютерные средства и технологии под прикладную задачу	

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
<i>Теория формальных языков и компиляторов</i>	
ПК.3.34 знать современные компиляторы, отладчики и оптимизаторы программного кода	
1. знать современные компиляторы, отладчики и оптимизаторы программного кода	Лекции
ОПК.5.35 знать правовые основы информационной безопасности и принципы защиты авторского права на программные продукты	
2. об общей совокупности проблем, возникающих при создании и реализации языков программирования, ориентированных на решение задач из конкретных прикладных областей человеческой деятельности	Лекции; Самостоятельная работа
3. о достигнутом на современном этапе уровне результатов теоретических исследований в области автоматизированного перевода с одного языка на другой	Лекции; Самостоятельная работа
4. об уровне практической реализации теоретических результатов для автоматизации проектирования трансляторов	Лекции; Самостоятельная работа
5. об актуальных вопросах теории формальных языков и трансляции, интенсивно развиваемых в настоящее время	Лекции; Самостоятельная работа
6. объект курса (трансляторы с языков программирования, системы автоматизации проектирования трансляторов), предмет курса (формальные системы определения лексики, синтаксиса и семантики языков программирования, методы анализа свойств таких систем, методы и технологии их преобразования в автоматы, решающие задачи соответствующих этапов анализа), задачи курса (свойства и характеристики языков программирования, их реализации, методы трансляции и автоматизация проектирования трансляторов), место теории языков программирования и методов трансляции в общей совокупности изучаемых дисциплин	Лекции; Самостоятельная работа

7.формальный аппарат для описания алгоритмических языков: системы регулярных выражений, контекстно-свободные грамматики, принципы использования конечных автоматов без памяти и со стековой памятью для построения анализаторов	Лекции; Лабораторные работы
8.свойства формальных систем определения языков программирования, их классификацию и методы эквивалентных преобразований внутри классов	Лекции; Лабораторные работы
9.методы преобразования формальных описаний лексики и синтаксиса языка в управляющие таблицы детерминированных оптимальных конечных автоматов	Лекции; Лабораторные работы
10.алгоритмы лексического, синтаксического и семантического анализа, реализуемые соответствующими конечными автоматами без памяти и с магазинной памятью	Лекции; Лабораторные работы
11.промежуточные формы представления транслируемой программы, методы ее оптимизации	Лекции; Лабораторные работы
12.методы генерации объектного кода для конкретной целевой машины.	Лекции; Лабораторные работы
ОПК.5.y10 уметь использовать специализированные программные средства при решении профессиональных задач	
13.использовать знания основ информатики, программирования на языках высокого уровня, теории конечных автоматов, дискретной математики, математической логики и теории алгоритмов при проектировании языка программирования и разработке транслятора для него	Лекции; Лабораторные работы; Самостоятельная работа
14.разрабатывать формальные определения лексики и синтаксиса языков программирования, определять свойства разрабатываемых формальных систем, оценивать их характеристики	Лабораторные работы; Самостоятельная работа
15.разрабатывать и использовать сложные структуры данных (таблицы, деревья, списки) и алгоритмы работы с этими структурами	Лекции; Лабораторные работы; Самостоятельная работа
ОПК.5.y11 уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	
16.представлять результаты решения отдельных задач, описание расчетно-графического задания по курсу, в удобной для восприятия форме	Лабораторные работы; Самостоятельная работа
ОПК.5.y5 уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	
17.выполнять эквивалентные преобразования регулярных выражений и формальных грамматик	Лабораторные работы; Самостоятельная работа
18.разрабатывать расширения формальных систем, определять функциональность семантического анализа и генерации объектного кода	Лекции; Лабораторные работы; Самостоятельная работа
19.осуществлять самооценку и самоконтроль при самостоятельном выполнении работ по проектированию и разработке элементов системного программного обеспечения	Лабораторные работы; Самостоятельная работа
ОПК.5.y8 владеть персональным компьютером как средством управления информацией	
20.использования средств автоматизации проектирования трансляторов для построения лексических, синтаксических и семантических анализаторов	Лекции; Самостоятельная работа
21.проектирования формального языка на разных уровнях определения его свойств и характеристик	Лекции; Лабораторные работы
ПК.10.В/ПТ.y8 уметь сравнивать и выбирать суперкомпьютерные средства и технологии под прикладную задачу	
22.уметь сравнивать и выбирать суперкомпьютерные средства и технологии под прикладную задачу	Лабораторные работы

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 6				
Дидактическая единица: Введение				

1. Введение	0	0,5	1, 2, 5	Освоение основных понятий формальных языков и языков программирования, их связей с задачами автоматического анализа и эквивалентного преобразования программ с одного языка на другой. Изучение систем правил и метаязыков для определения лексики, синтаксиса и семантики формальных языков, этапов процесса трансляции и промежуточных форм представления программы в трансляторах.
Дидактическая единица: Лексический анализ				
2. Регулярные выражения	0	1,5	6, 7	Изучение метаязыков для определения лексики формальных языков, в частности - регулярных выражений. Правила построения регулярных выражений. Связь регулярных выражений с понятием недетерминированного конечного автомата без памяти. Алгоритм преобразования регулярного выражения в граф состояний и переходов недетерминированного конечного автомата без памяти. Типы недетерминированности и методы их устранения.
3. Лексический акцепт	0	3	10, 7	Изучение групп методов разработки лексических акцепторов - процедурной и автоматной, алгоритма преобразования недетерминированного конечного автомата в детерминированный лексический акцептор. изучение методов оптимизации управляющей таблицы лексического акцептора и методики построения лексического акцептора для нескольких групп слов

4. Лексический анализ.	0	2	13, 8	Изучение понятий лексемы и токена как промежуточного представления слов формального языка в трансляторе. Формирование лексем и методы организации информационных таблиц транслятора. Простые последовательные, упорядоченные и рандомизированные методы поиска и дополнения, их временные оценки. Структура и основные алгоритмы лексического анализа-тора. Обработка ошибок при лексическом анализе.
Дидактическая единица: Синтаксический анализ				
5. Формальные грамматики	0	2	7, 8	Изучение основных понятий и классификации формальных грамматик. Алфавиты терминальных и нетерминальных символов. Цепочки символов, порождающие правила, понятие непосредственного вывода. Понятие правильного предложения как цепочки терминальных символов, которая может быть получена путём некоторой последовательности непосредственных выводов из начального нетерминального символа грамматики. Понятие пустой цепочки символов. Регулярные, контекстно-свободные, контекстно-зависимые и общие грамматики. Соотношение между языками и порождающими их грамматики.

6. Свойства грамматик.	0	2	7, 8, 9	Изучение свойств грамматик и их символов. Свойства нетерминальных символов грамматики. Аннулируемые, недостижимые и бесплодные нетерминалы, алгоритмы их определения. Отношения между символами грамматики. Понятие символа-предшественника. Алгоритм определения множеств предшественников для символа грамматики. Понятие множества предшественников цепочки символов, алгоритм его определения. Понятие символа-последователя. Алгоритм определения множеств непосредственных последователей для символов грамматики. Алгоритм определения множеств символов, замыкающих цепочки, выводимые из символов грамматики. Алгоритм определения отношения "символ - последователь" для всех символов грамматики. Эквивалентные преобразования грамматик.
7. Дерево грамматического разбора	0	2	6, 7, 9	Понятие дерева грамматического разбора и его связь с задачами синтаксического анализа. Свойства дерева разбора: левосторонность, правосторонность, однозначность. Нисходящие и восходящие группы методов синтаксического анализа, их основные идеи. Понятие конечного автомата со стековой памятью в качестве основы синтаксического акцептора.

8. Нисходящие методы синтаксического анализа.	0	6	10, 7, 9 Изучение нисходящих методов синтаксического анализа. Основной алгоритм нисходящего синтаксического анализа, критерии выбора грамматики для реализации этого алгоритма. Понятие S- и Q-грамматики и множества выбора для правила грамматики. Обобщение понятия множества выбора для замыкающих подцепочек символов правой части правила. Понятие и определение LL(1)-грамматики. Алгоритм определения принадлежности грамматики к классу LL(1). Задача преобразования LL(1)-грамматики в управляющую таблицу нисходящего синтаксического акцептора. Структура и состав управляющей таблицы, алгоритмы определения множеств выбора, адресов перехода и управляющих флажков. Алгоритм преобразования LL(1)-грамматики в управляющую таблицу. Алгоритм функционирования нисходящего синтаксического акцептора. Метод преобразования LL(1)-грамматики в программу, реализующую синтаксический анализ рекурсивным спуском. Понятие LL(k)-языков и LL(k)-грамматик. Эквивалентные преобразования грамматик с целью улучшения LL-свойств.
--	----------	----------	--

9. Восходящие методы синтаксического анализа.	0	6	10, 7, 9 Восходящий акцептор в виде конечного автомата со стековой памятью и единственным рабочим состоянием. Операции сдвига и свёртки, состояние стека автомата до и после их выполнения. Возможность уменьшения количества шагов работы автомата путём введения нескольких рабочих состояний и стека номеров состояний. Алгоритмы выполнения операций сдвига и свёртки для такого автомата. Возможность ликвидации стека символов. Задача определения множества состояний восходящего акцептора. Понятие конфигурации, связь между подмножествами конфигураций и состоянием автомата. Алгоритм построения подмножеств эквивалентных конфигураций. Разметка грамматики. Структура управляющей таблицы восходящего акцептора. Первый этап заполнения управляющей таблицы: занесение знаков операции сдвига. Формирование знаков операции свёртки, понятие конфликтов типа "сдвиг/свёртка" и "свёртка/свёртка". Понятие LR(k)-грамматик. Случай LR(0) - отсутствие конфликтов при построении управляющей таблицы. Разрешение конфликтов методом использования полных множ
--	----------	----------	---

10. Постфиксная форма записи.	0	3	11, 15, 18	Понятие постфиксной формы записи, её связь с деревом грамматического разбора, задача преобразования последовательности лексем в постфиксную запись. Понятие грамматики действий (синтаксически управляемой или СУ-схемы), как расширения формальной грамматики, позволяющей определять операции синтаксического анализатора по формированию постфиксной записи. Задача генерации уникальных символов для преобразования в постфиксную запись структурированных управляющих конструкций языка. Обработка ошибок при синтаксическом анализе, методы нейтрализации ошибок
-------------------------------	---	---	------------	--

Дидактическая единица: Семантический анализ

11. Задачи семантического анализа.	0	2	10, 11, 18	Основные понятия семантики языка программирования. Атрибутные грамматики. Наследуемые и синтезируемые атрибуты. Использование атрибутных грамматик для решения задач семантического анализа.
------------------------------------	---	---	------------	--

12. Методы семантического анализа.	0	2	10, 11, 18, 20	Операции, операнды и результаты. Виды переменных, наименования и значения, типы значений, классы памяти, области действия и зоны видимости, их связь с понятием блока. Статическое и динамическое связывание. Неименованные промежуточные результаты вычислений. Задача преобразования постфиксной записи в последовательность тетрад. Состав и структура таблицы идентификаторов, операции над таблицей при обработке входа в блок и выхода из блока. Задача проверки правильности употребления наименований объектов (операций, операндов и результата) в последовательности тетрад. Обработка семантических ошибок.
------------------------------------	---	---	----------------	--

Дидактическая единица: Оптимизация, генерация кода и интерпретация

13. Оптимизация программы.	0	2	11, 15, 21	Цели и задачи оптимизации транслируемой программы. Оптимизация по занимаемой памяти и по времени выполнения. Методы оптимизации. Выявление зависимостей по обрабатываемым данным между операциями. Изменение порядка вычислений, вынос инвариантных операций из циклов.
14. Генерация объектного кода.	0	1	12, 20	Генерация псевдокода и связанные с этим задачи распределения памяти транслируемой программы. Статические и динамические объекты, формирование относительных адресов объектов разных классов памяти. Методы вычисления исполнительных адресов для доступа к собственным и к внешним объектам. Методы управления памятью при рекурсивном вызове функций. Передача параметров.
15. Интерпретация.	0	0,5	11, 15, 20	Отличия интерпретаторов от компиляторов. Внутреннее представление программы для интерпретации. Хранение обрабатываемых данных.
Дидактическая единица: Заключение				
16. Заключение.	0	0,5	3, 4	Место трансляторов в системном программном обеспечении. Направления их дальнейшего развития.

Таблица 3.2

Темы лабораторных работ	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 6				
Дидактическая единица: Лексический анализ				
1. Регулярные выражения	2	4	14, 8, 9	Освоение учебного пакета автоматизации проектирования трансляторов. Разработка систем регулярных выражений для заданного варианта формального языка. Изучение и освоение методов преобразования систем регулярных выражений в конечные автоматы без памяти. Исследование поведения конечных автоматов без памяти в качестве лексических акцепторов.
2. Лексический акцепт	2	4	13, 17, 7	

3. Лексический анализ	2	4	17, 9	Изучение и освоение технологии расширения систем регулярных выражений действиями для преобразования токенов (лексем) и поиска/пополнения информационных таблиц.
Дидактическая единица: Синтаксический анализ				
4. Формальные грамматики	2	4	14, 17, 9	Освоение технологии разработки формальных грамматик для определения правил синтаксиса формального языка.
5. Нисходящие методы синтаксического анализа	3	4	10, 13, 14	Изучение методов преобразования LL(1)-грамматик в нисходящие синтаксические анализаторы.
6. Восходящие методы синтаксического анализа	3	4	10, 13, 14	Изучение методов построения LR(0), SLR(1), LALR(1) и LR(1) конечных автоматов со стековой памятью для восходящего синтаксического анализа.
7. Постфиксная форма записи	2	4	11, 14, 16, 17, 19	Разработка совокупности действий для преобразования заданной программы на учебном языке в постфиксную форму записи.
Дидактическая единица: Семантический анализ				
8. Семантический анализ	2	4	10, 18, 19, 21	Изучение и разработка алгоритмов преобразования постфиксной формы записи в последовательность операторов псевдокода и оптимизации этого псевдокода.
Дидактическая единица: Оптимизация, генерация кода и интерпретация				
9. Генерация кода и интерпретация	0	4	12, 15, 21, 22	Разрабатывается простейший генератор объектного кода или интерпретатор псевдокода

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 6				
1	Курсовая работа	13, 14, 15, 16, 18, 19, 20, 6	24	4
Разработка транслятора для заданного варианта учебного языка программирования: Системное программное обеспечение ЭВМ. Трансляторы : методические указания к лабораторным работам и курсовому проектированию для АВТФ всех форм обучения / Новосиб. гос. техн. ун-т ; [сост. А. А. Малявко]. - Новосибирск, 2006. - 76, [3] с.				
2	Изучение основной и дополнительной литературы по темам занятий	17, 2, 3, 4, 5	22	6

: Малявко А. А. Теория формальных языков и компиляторов [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 230100 Информатика и вычислительная техника] / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000196666. - Загл. с экрана.

3	Самостоятельное изучение теоретического материала	15, 17, 2, 3, 4, 5	14	0
---	---	--------------------	----	---

Самостоятельное изучение теоретического материала по основной и дополнительной литературе, поиск в Интернете: Малявко А. А. Системное программное обеспечение. Методы трансляции [Электронный ресурс] : электронный учебно-методический комплекс / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156340. - Загл. с экрана. Малявко А. А. Теория формальных языков и компиляторов [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 230100 Информатика и вычислительная техника] / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000196666. - Загл. с экрана. Системное программное обеспечение ЭВМ. Трансляторы : методические указания к лабораторным работам и курсовому проектированию для АВТФ всех форм обучения / Новосиб. гос. техн. ун-т ; [сост. А. А. Малявко]. - Новосибирск, 2006. - 76, [3] с.

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	Личный типовой сайт: http://vt.cs.nstu.ru/~malyavko
Консультирование	e-mail:malyavko@vt.cs.nstu.ru
Контроль	e-mail:malyavko@vt.cs.nstu.ru
Размещение учебных материалов	ЭБС:Системное программное обеспечение. Языки программирования и методы трансляции : электронный учебно-методический комплекс

Таблица 5.2

Активные и интерактивные формы проведения занятий

№	Наименование активных форм
1	Метод проектов:Разработка элементов транслятора
Краткое описание применения: Все лабораторные работы выполняются студентом с учетом задания на курсовую работу.	
Подробная информация об использовании технологии приводится в "Системное программное обеспечение ЭВМ. Трансляторы : методические указания к лабораторным работам и курсовому проектированию для АВТФ всех форм обучения / Новосиб. гос. техн. ун-т ; [сост. А. А. Малявко]. - Новосибирск, 2006. - 76, [3] с."	

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 6		

Лабораторная №1: Регулярные выражения	2	4
Лабораторная №2: Лексический акцепт	4	8
Лабораторная №3: Лексический анализ	4	8
Лабораторная №4: Формальные грамматики	4	8
Лабораторная №5: Нисходящие методы синтаксического анализа	4	8
Лабораторная №6: Восходящие методы синтаксического анализа	4	8
Лабораторная №7: Постфиксная форма записи	4	8
Лабораторная №8: Семантический анализ	4	8
Курсовая работа №0: Курсовая работа	0	
Курсовая работа №1: Лексический анализ	4	8 (в состав баллов за КР)
Курсовая работа №2: Синтаксический анализ	16	32 (в состав баллов за КР)
Курсовая работа №3: Семантический анализ	12	24 (в состав баллов за КР)
Курсовая работа №4: Генерация ОК / Интерпретация	8	16 (в состав баллов за КР)
Курсовая работа №5: Защита	10	20 (в состав баллов за КР)
Экзамен:	0	40

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля	
		Защита КР/КР	Экзамен
ОПК.5	з5. знать правовые основы информационной безопасности и принципы защиты авторского права на программные продукты		+
	у10. уметь использовать специализированные программные средства при решении профессиональных задач	+	+
	у11. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	+	+
	у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	+	+
	у8. владеть персональным компьютером как средством управления информацией	+	+
ПК.3	з4. знать современные компиляторы, отладчики и оптимизаторы программного кода		+
	ПК.10.В/ПТ у8. уметь сравнивать и выбирать суперкомпьютерные средства и технологии под прикладную задачу		+

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Малявко А. А. Формальные языки и компиляторы : [учебное пособие для вузов по направлению подготовки 230100 "Информатика и вычислительная техника"] / А. А. Малявко. - Новосибирск, 2014. - 429, [1] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000184529
2. Малявко А. А. Формальные языки и компиляторы : [учебник] / А. А. Малявко. - Новосибирск, 2013. - 429, [1] с. : табл., ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000181694
3. Малявко А. А. Системное программное обеспечение. Формальные языки и методы трансляции. [В 3 ч.]. Ч. 1 : [учебное пособие] / А. А. Малявко ; Новосиб. гос. техн. ун-т, Фак. автоматики и вычисл. техники. - Новосибирск, 2010. - 99, [2] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000143812
4. Малявко А. А. Системное программное обеспечение. Формальные языки и методы трансляции. [В 3 ч.]. Ч. 2 : [учебное пособие] / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, 2011. - 158, [1] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000155235
5. Малявко А. А. Системное программное обеспечение. Формальные языки и методы трансляции. [В 3 ч.]. Ч. 3 : [учебное пособие] / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, 2012. - 118, [1] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000170641

Дополнительная литература

1. Малявко А. А. Теория формальных языков и компиляторов [Электронный ресурс] : электронный учебно-методический комплекс [для студентов направления 230100 Информатика и вычислительная техника] / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2014]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000196666. - Загл. с экрана.
2. Малявко А. А. Системное программное обеспечение. Методы трансляции [Электронный ресурс] : электронный учебно-методический комплекс / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000156340. - Загл. с экрана.
3. Молчанов А. Ю. Системное программное обеспечение : учебник для вузов / А. Ю. Молчанов. - СПб., 2003. - 395 с. : ил.. - На тит. л.: Издательская программа "300 лучших учебников для высшей школы в честь 300-летия Санкт-Петербурга".
4. Карпов Ю. Г. Теория и технология программирования. Основы построения трансляторов : учебное пособие для вузов по направлениям подготовки бакалавров и магистров 553000 - "Системный анализ и управление" и 552800 - "Информатика и вычислительная техника" / Ю. Г. Карпов. - СПб., 2005. - 270 с. : ил., табл.
5. Ахо А. В. Компиляторы : Принципы, технологии, инструменты / А. Ахо, Р. Сети, Д. Ульман. - М., 2001. - 766 с.
6. Мозговой М. В. Классика программирования : алгоритмы, языки, автоматы, компиляторы : практический подход / Мозговой М. В. - СПб., 2006. - 320 с. : ил.
7. Хантер Р. Проектирование и конструирование компиляторов / Р. Хантер; пер. с англ. С. М. Круговой под ред. В. М. Савинкова. - М., 1984. - 231, [1] с. : ил.
8. Льюис Ф. Теоретические основы проектирования компиляторов / Ф. Льюис, Д. Розенкранц, Р. Стирнз ; пер. с англ. В. А. Исаева [и др.] ; под ред. В. Н. Агафонова. - М., 1979. - 654 с.
9. Теория и реализация языков программирования : учебное пособие по курсу теории и реализации языков программирования / В. А. Серебряков [и др.]. - М., 2003. - 294 с. : ил., табл.

10. Грис Д. Конструирование компиляторов для цифровых вычислительных машин / Д. Грис ; пер. с англ. Е. Б. Докшицкой, Л. А. Зелениной, Л. Б. Морозовой, В. С. Штаркмана, под ред. Ю. М. Баяковского, Вс. С. Штаркмана. - М., 1975. - 544 с. : табл., схемы

11. Свердлов С. З. Языки программирования и методы трансляции : учебное пособие для вузов / С. З. Свердлов. - СПб., 2007. - 637 с. + 1 CD-ROM. - Издательская программа 300 лучших учебников для высшей школы.

Интернет-ресурсы

1. Вебтранслаб [Электронный ресурс] : учебный пакет автоматизации проектирования трансляторов, реализованный по архитектуре «клиент-сервер». - Режим доступа: <http://217.71.138.14:8083/wtl>. - Загл. с экрана.

2. ЭБС НГТУ : <http://elibrary.nstu.ru/>

3. ЭБС «Издательство Лань» : <https://e.lanbook.com/>

4. ЭБС IPRbooks : <http://www.iprbookshop.ru/>

5. ЭБС "Znaniy.com" : <http://znaniy.com/>

6. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Системное программное обеспечение ЭВМ. Трансляторы : методические указания к лабораторным работам и курсовому проектированию для АВТФ всех форм обучения / Новосиб. гос. техн. ун-т ; [сост. А. А. Малявко]. - Новосибирск, 2006. - 76, [3] с.

8.2 Специализированное программное обеспечение

1 Учебный пакет автоматизации проектирования трансляторов "Вебтранслаб"

9. Материально-техническое обеспечение

Компьютерный класс

№	Наименование	Назначение
1	Компьютерный класс (Компьютеры объединены в локальную сеть с выходом в Internet)	Выполнение всех лабораторных работ

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

Кафедра автоматизированных систем управления
Кафедра вычислительной техники

“УТВЕРЖДАЮ”
ДЕКАН АВТФ
к.т.н., доцент И.Л. Рева
“ ” _____ Г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ УЧЕБНОЙ ДИСЦИПЛИНЫ

Теория формальных языков и компиляторов

Образовательная программа: 09.03.01 Информатика и вычислительная техника, профиль:
Программное обеспечение компьютерных систем и сетей

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Теория формальных языков и компиляторов приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.5 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	з5. знать правовые основы информационной безопасности и принципы защиты авторского права на программные продукты	Введение Восходящие методы синтаксического анализа Восходящие методы синтаксического анализа. Генерация объектного кода. Дерево грамматического разбора Задачи семантического анализа. Заключение. Интерпретация. Лексический акцепт Лексический анализ Лексический анализ. Методы семантического анализа. Нисходящие методы синтаксического анализа Нисходящие методы синтаксического анализа. Оптимизация программы. Постфиксная форма записи Постфиксная форма записи. Регулярные выражения Свойства грамматик. Семантический анализ Формальные грамматики		Экзамен, вопросы 1-8
ОПК.5	у5. уметь применять основные методы, способы и средства получения, хранения и переработки информации с помощью компьютеров и компьютерных средств	Задачи семантического анализа. Лексический акцепт Лексический анализ Методы семантического анализа. Постфиксная форма записи Постфиксная форма записи. Семантический анализ Формальные грамматики	Курсовая работа, разделы 1, 2	Экзамен, вопросы 9-21
ОПК.5	у8. владеть персональным компьютером как средством управления информацией	Генерация объектного кода. Интерпретация. Методы семантического анализа. Оптимизация программы. Семантический анализ	Курсовая работа, разделы 4, 5	Экзамен, вопросы 22-30
ОПК.5	у10. уметь использовать специализированные программные средства при решении профессиональных задач	Восходящие методы синтаксического анализа Генерация кода и интерпретация Интерпретация. Лексический акцепт Лексический анализ. Нисходящие методы синтаксического анализа Оптимизация программы. Постфиксная форма записи	Курсовая работа, разделы 2,3	Экзамен, вопросы 31-38

		Постфиксная форма записи. Регулярные выражения Формальные грамматики		
ОПК.5	у11. уметь использовать элементарные навыки алгоритмизации и программирования на одном из языков высокого уровня как средство программного моделирования изучаемых объектов и процессов	Постфиксная форма записи	Курсовая работа, разделы 3, 4	Экзамен, вопросы 39-44
ПК.10.В/ПТ готовность к разработке компонентов аппаратно-программных комплексов и баз данных с использованием современных инструментальных средств и технологий программирования	у8. уметь сравнивать и выбирать суперкомпьютерные средства и технологии под прикладную задачу	Генерация кода и интерпретация		Экзамен, вопросы 45-52
ПК.3/НИ готовность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности	з2. знать современные компиляторы, отладчики и оптимизаторы программного кода	Введение		Экзамен, вопросы 53-60

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по **дисциплине** проводится в 6 семестре - в форме экзамена, который направлен на оценку сформированности компетенций ОПК.5, ПК.10.В/ПТ, ПК.3/НИ.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 6 семестре обязательным этапом текущей аттестации является курсовая работа. Требования к выполнению курсовой работы, состав и правила оценки сформулированы в паспорте курсовой работы.

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе учебной дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.5, ПК.10.В/ПТ, ПК.3/НИ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

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

Паспорт экзамена

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

1. Методика оценки

Экзамен проводится в устной форме, по билетам. Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-29, второй вопрос из диапазона вопросов 30-60 (список вопросов приведен ниже). В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма экзаменационного билета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к экзамену по дисциплине «Теория формальных языков и компиляторов»

1. Преобразование системы регулярных выражений в недетерминированный конечный автомат.
2. Разрешение конфликтов методом использования полных множеств последователей. Понятие SLR(1)-грамматик.

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись)

(дата)

2. Критерии оценки

- Ответ на экзаменационный билет считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен обосновать причинно-следственные связи процессов трансляции, оценка составляет 0-49 *баллов*.
- Ответ на экзаменационный билет засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи процессов трансляции, оценка составляет 50-72 *баллов*.
- Ответ на экзаменационный билет засчитывается на **базовом** уровне, если студент при ответе на вопросы четко формулирует основные понятия и соотношения, дает характеристику процессов трансляции, проводит анализ причин и может представить качественные характеристики процессов, оценка составляет 73-85 *баллов*.
- Ответ на экзаменационный билет (тест) билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный анализ подходов,

проводит комплексный анализ процессов трансляции, выявляет возникающие проблемы, предлагает механизмы их решения, способен промоделировать поведение автоматов-анализаторов, приводит конкретные примеры, оценка составляет 86-100 баллов.

3. Шкала оценки

В общей оценке по дисциплине экзаменационные баллы учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Вопросы к экзамену по дисциплине «Теория формальных языков и компиляторов»

- 1 Формальные языки, основные понятия.
- 2 Этапы процесса трансляции, промежуточные формы представления программы
- 3 Лексический анализ. Основные функции. Понятия слова и лексем, их сходство и различие
- 4 Процедурная модель лексического анализатора. Особенности алгоритма обработки слов
- 5 Автоматная модель лексического анализатора: стадии разработки и функционирования
- 6 Конечные автоматы без памяти, основные понятия, детерминированность и недетерминированность КА
- 7 Эквивалентность конечных автоматов, понятие истории работы КА, виды эквивалентности КА
- 8 Оптимальность конечных автоматов без памяти
- 9 Способы определения лексики формальных языков. Первичные регулярные выражения
- 10 Регулярные выражения и системы регулярных выражений
- 11 Преобразование системы регулярных выражений в недетерминированный конечный автомат
- 12 Задача преобразования недетерминированного автомата в детерминированный оптимальный конечный автомат, структура алгоритма ее решения
- 13 Устранение пересечения множеств символов разметки столбцов
- 14 Ликвидация всех недетерминированностей одним алгоритмом
- 15 Удаление тупиковых и недостижимых состояний
- 16 Оптимизация управляющей таблицы лексического акцептора
- 17 Преобразование не полностью определенного КА в полностью определенный
- 18 Взаимодействие лексического анализатора с синтаксическим и семантическим анализаторами. Структура информационных таблиц транслятора
- 19 Последовательная и упорядоченная организация информационных таблиц, алгоритмы поиска и дополнения, временные оценки
- 20 Рандомизированная организация информационных таблиц, алгоритмы поиска и дополнения, временные оценки
- 21 Синтаксический анализ, основные понятия и функции
- 22 Формальные грамматики, основные понятия
- 23 Понятие цепочки, предложения, непосредственного вывода и вывода. Соотношение между грамматиками и языками
- 24 Деревья грамматического разбора, их свойства и связь с задачами синтаксического анализа
- 25 Классификация формальных грамматик

- 26 Свойства формальных грамматик: рекурсивность левая, правая и общая, однозначность
- 27 Аннулируемые, недостижимые и бесплодные нетерминалы, алгоритмы их определения
- 28 Отношения между символами. Понятие символа-предшественника. Алгоритм определения множества предшественников символа и цепочки символов
- 29 Понятие символа-последователя. Алгоритм определения множества непосредственных последователей для символов грамматики
- 30 Алгоритм определения множеств символов, которые могут быть последними в цепочках, выводимых из нетерминалов. Вычисление множеств последователей
- 31 Нисходящие методы синтаксического анализа, основные идеи
- 32 Понятие множеств выбора правил грамматики с одним нетерминалом в правой части. Определение LL(1)-грамматик
- 33 Процедурная реализация рекурсивного спуска для LL(1)-грамматик
- 34 Автоматная реализация рекурсивного спуска. Состав клетки управляющей таблицы автомата с несколькими состояниями
- 35 Преобразование LL(1)-грамматики в управляющую таблицу автомата с несколькими состояниями
- 36 Преобразование LL(1)-грамматики в автомат с одним состоянием, управляемый входным символом и содержимым стека
- 37 Восходящие методы синтаксического анализа, основные идеи
- 38 Конфликты свертка/свертка и перенос/свертка. Развитие основной идеи восходящего анализа, восходящий автомат с несколькими состояниями
- 39 Операции сдвига и свертки восходящего синтаксического автомата с несколькими состояниями
- 40 Понятие конфигурации и его связь с понятием состояния восходящего акцептора и его операциями
- 41 Построение таблицы конфигураций по грамматике
- 42 Преобразование таблицы конфигураций в управляющую таблицу восходящего акцептора. Выявление конфликтов. Понятие LR(0)-грамматик
- 43 Разрешение конфликтов методом использования полных множеств последователей. Понятие SLR(1)-грамматик
- 44 Понятие ожидаемого правого контекста и расширенной конфигурации. Правила определения правого контекста для расширенных конфигураций
- 45 Метод использования правого контекста для предотвращения конфликтов при построении управляющей таблицы. LR(1)-грамматики и автоматы
- 46 LALR(1)-грамматики и автоматы
- 47 Эквивалентные преобразования грамматик
- 48 Нейтрализация ошибок при синтаксическом анализе.
- 49 Понятие постфиксной записи, преобразование выражений в ПФЗ.
- 50 Преобразование управляющих операторов в постфиксную запись.
- 51 Семантический анализ, основные задачи и понятия. Операции и данные, адреса и значения, l-value и r-value
- 52 Тетрады и триады, алгоритм преобразования ПФЗ в последовательность тетрад
- 53 Оптимизация псевдокода. Основные методы
- 54 Оптимизация псевдокода. Методы глубокой оптимизации
- 55 Генерация кода, основные понятия
- 56 Генерация кода. Распределение памяти
- 57 Генерация кода. Формирование исполнительных адресов
- 58 Интерпретация псевдокода
- 59 Зависимость времени работы программы от метода ее трансляции (генерация кода или интерпретация)

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?