

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Смирнов Сергей Николаевич
Должность: врио ректора
Дата подписания: 24.11.2023 15:59:12
Уникальный программный ключ:
69e375c64f7e975d4e8830e7b4fcc2ad1bf35f08

Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тверской государственный университет»

Утверждаю:
Руководитель ООП
С.М. Дудаков
2023 г.



Рабочая программа дисциплины (с аннотацией)
Языки программирования и методы трансляции

Направление подготовки
01.03.02 Прикладная математика и информатика

Профиль подготовки
Искусственный интеллект и анализ данных

для студентов 2 курса
Форма обучения – очная

Составитель(и):
• к.ф.-м.н. Карлов Б.Н.

I. Аннотация

1. Цель и задачи дисциплины:

Изучить основные группы языков программирования, овладеть методологией объектно-ориентированного программирования, изучить методы обработки данных на формальных языках.

2. Место дисциплины в структуре ООП

Дисциплина входит в раздел «Информатика и информационно-коммуникационные технологии» обязательной части блока 1.

Предварительные знания и навыки. Знание курсов «Алгоритмы и программы», «Дискретная математика», «Методы программирования», «Практикум на ЭВМ».

Дальнейшее использование. Полученные знания используются в последующем при изучении предметов: «Теория автоматов и формальных языков», «Архитектура ЭВМ», «Базы данных», «Программная инженерия», а также в трудовой деятельности выпускников. Знания, умения и навыки полученные при изучении дисциплины закрепляются практикумом на ЭВМ и в конце каждого семестра учебной практикой.

3. Объем дисциплины: 6 зач. ед., 216 акад. ч., в том числе:

контактная аудиторная работа лекций 62 ч., в том числе практическая подготовка 62 ч., практических занятий 31 ч., в том числе практическая подготовка 31 ч., **контактная внеаудиторная работа** контроль самостоятельной работы 20 ч., в том числе курсовая (расчетно-графическая) работа 20 ч.; **самостоятельная работа** 103 ч., в том числе контроль 74 ч.

4. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы:

Планируемые результаты освоения образовательной программы (формируемые компетенции)	Планируемые результаты обучения по дисциплине
ОПК-2, Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач	ОПК-2.1, Знает существующие математические методы и системы программирования ОПК-2.2, Использует существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач ОПК-2.3, Разрабатывает и реализует алгоритмы решения задач профессиональной деятельности на основе теоретических знаний

5. Форма промежуточной аттестации и семестр прохождения:

экзамен в 3–4 семестрах, РГР

6. Язык преподавания:

русский

II. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Для студентов очной формы обучения

Учебная программа – наименование разделов и тем	Всего (час.)	Контактная работа (час.)				Контроль сам. раб., в т.ч. РГР	Сам. раб., в т.ч. контроль (час.)
		Лекции		Практ. занятия / Лаб. работы			
		Всего	В т.ч. практ. подг.	Всего	В т.ч. практ. подг.		
1	2	3	4	5	6	7	8
Объектно-ориентированное программирование	46	14	14	7/0	7/0	5	20
Классы языков программирования	28	6	6	4/0	4/0	5	13
Лексический анализ текста	39	10	10	4/0	4/0	5	20
Синтаксический анализ	103	32	32	16/0	16/0	5	50
Итого	216	62	62	31/0	31/0	20	103

Учебная программа дисциплины

1. Объектно-ориентированное программирование

- Принципы объектно-ориентированного проектирования
- Классы в языке C++, инкапсуляция, скрытие данных
- Специальные элементы классов: константы, конструкторы, деструкторы
- Перегрузка операций
- Статические элементы классов
- Наследование в C++
- Полиморфизм и виртуальные функции
- Обработка исключительных ситуаций
- Динамическая идентификация типа
- Основы стандартной библиотеки шаблонов STL

2. Классы языков программирования

- Процедурные языки
- Объектно-ориентированные языки

- Языки функционального программирования
 - Языки, основанные на обратной польской записи
 - Элементы ассемблера
3. Лексический анализ текста
- Использование конечных автоматов для лексического анализа
 - Особенности чтения лексем, встречающихся в реальных языках
 - Примеры реализации лексического анализатора разными методами
4. Синтаксический анализ
- Формальные грамматики, как основа синтаксического анализа
 - Нисходящий и восходящий синтаксический анализ
 - Модель перенос-свертка, восходящий табличный анализатор
 - LR-грамматики, построение анализатора по LR-грамматике
 - Грамматики простого предшествования, построение анализатора по грамматике простого предшествования
 - Грамматики слабого предшествования
 - Автоматы с магазинной памятью
 - LL-грамматики, построение МП-автомата по LL-грамматике
 - Транслирующие грамматики, атрибуты, особенности построения атрибутных транслирующих грамматик для восходящего и нисходящего разборов
 - Пример построения синтаксического анализатора для подмножества языка С

III. Образовательные технологии

Учебная программа – наименование разделов и тем	Вид занятия	Образовательные технологии
Объектно-ориентированное программирование	лекции, практические занятия	изложение теоретического материала, решение задач, выполнение РГР
Классы языков программирования	лекции, практические занятия	изложение теоретического материала, решение задач, выполнение РГР
Лексический анализ текста	лекции, практические занятия	изложение теоретического материала, решение задач, выполнение РГР
Синтаксический анализ	лекции, практические занятия	изложение теоретического материала, решение задач, выполнение РГР

IV. Оценочные материалы для проведения текущей и промежуточной аттестации

Типовые контрольные задания и/или критерии для проверки индикатора ОПК-2.1

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
Знать принципы объектно-ориентированного про-	Примеры вопросов к экзамену: • Классы в языке С++, инкапсуляция, скрытие данных.	оценка 3 — знает основы объектно-ориентированного

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
граммирования	<ul style="list-style-type: none"> • Специальные элементы классов: константы, конструкторы, деструкторы. • Перегрузка операций. • Статические элементы классов. • Наследование в C++. Множественное наследование. Виртуальные базовые классы. • Полиморфизм и виртуальные функции. • Обработка исключительных ситуаций. • Система ввода/вывода языка C++. Манипуляторы. • Стандартная библиотека шаблонов. 	программирования, понятие класса, поля, метода, оценка 4 — кроме того знает перегрузку операций, наследование, систему ввода/вывода, оценка 5 — кроме того знает виртуальные функции, библиотеку STL.
Знать методы лексического анализа текста	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Конечные автоматы-распознаватели и конечные автоматы-преобразователи. • Распознавание констант языка MINI-BASIC. • Распознавание ключевых слов. Использование вектора альтернатив. • Реализация лексического анализатора на ЭВМ. 	оценка 3 — знает понятие конечно-го автомата, некоторые методы лексического анализа, оценка 4 — знает различные методы лексического анализа, оценка 5 — кроме того знает различные способы реализации лексического анализатора
Знать восходящий синтаксический анализ	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Алгоритм перенос-свертка. • LR-грамматики. Построение восходящего табличного анализатора по LR-грамматике. • LALR-грамматики. • Грамматики простого, слабого и операторного предшествования. Отношения предшествования Вирта-Вебера. • Построение анализатора типа «перенос-свертка» по грамматике предшествования. 	оценка 3 — знает алгоритм «перенос-свертка» с возвратами, оценка 4 — кроме того знает определения LR-грамматик, LALR-грамматик и грамматик предшествования, оценка 5 — кроме того знает алгоритмы построения восходящих анализаторов по перечисленным выше типам грамматик
Знать нисходящий синтаксический анализ	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Нисходящий синтаксический анализ. • Автоматы с магазинной памятью. • LL-грамматики. Построение МП-автомата по LL-грамматике. 	оценка 3 — знает алгоритм нисходящего синтаксического анализа с возвратами, оценка

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
		4 — кроме того знает понятия LL-грамматики и МП-автомата, оценка 5 — кроме того знает алгоритм построения МП-автомата по LL-грамматике
Знать понятие атрибутивных транслирующих грамматик	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Транслирующие грамматики. • Атрибутивные грамматики. Синтезируемые и наследуемые атрибуты. S-атрибутивные и L-атрибутивные грамматики. • Атрибутивные транслирующие грамматики. Использование атрибутивных транслирующих грамматик для восходящего и нисходящего разбора. 	оценка 3 — знает понятие атрибутивной грамматики и транслирующей грамматики, оценка 4 — знает понятие атрибутивной транслирующей грамматики, синтезируемых и наследуемых атрибутов, оценка 5 — знает возможности применения атрибутивных транслирующих грамматик для синтаксического анализа и трансляции языков

Типовые контрольные задания и/или критерии для проверки индикатора ОПК-2.2

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
Уметь применять на практике языки объектно-ориентированного программирования.	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Описать класс для представления натуральных чисел произвольной длины в двоичном виде (в виде массива нулей и единиц). Предусмотреть: конструктор по умолчанию, конструктор копирования, конструктор, принимающий целое число, деструктор, оператор присваивания, оператор потокового вывода <<, операторы сложения, инкремента, декремента, сравнения == и <. 	оценка 3 — умеет создавать класс по описанию типа данных, реализовывать методы и операции, оценка 4 — кроме того умеет интегрировать новый тип данных в стандартную систему ввода/вывода, осуществлять обработку ошибок с помо-

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
		щью исключительных ситуаций, оценка 5 — кроме того умеет использовать наследование и виртуальные функции, строить иерархию классов
Уметь писать программы с использованием обратной польской записи	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Написать программу с использованием обратной польской записи, которая вычисляет функцию $\text{arctg } x$ на промежутке $(-1; 1]$, используя ряд Тейлора, с заданной точностью ε (т.е. пока очередное слагаемое не станет меньше ε). • Напишите программу на стековом языке, которая находит количество степеней двойки на промежутке $[x, y)$. Разрешается использовать операции сравнения, +, -, *, /. 	оценка 3 — умеет строить по инфиксному выражению обратную польскую запись, оценка 4 — кроме того умеет писать простые программы на стековом языке, оценка 5 — кроме того умеет писать более сложные программы на стековом языке
Уметь разрабатывать лексический анализатор	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Построить лексический анализатор, описать автомат и выполняемые действия. Реализовать программно в виде одной функции с несколькими case. Язык состоит из лексем $\langle - \rangle \langle \langle \rangle \rangle$ -- и может содержать идентификаторы, состоящие из одной буквы и следующих за ней нескольких цифр, и комментарии в виде <i># комментарий до конца строки</i> • Постройте лексический анализатор, опишите автомат и выполняемые действия. Язык состоит из следующих лексем: <ul style="list-style-type: none"> • операции ++ +- ** -*; • идентификаторы, состоящие из русских букв; • комментарии вида <i>REM комментарий до конца строки</i>; • ключевые слова ASC ASM CALL CASE REM. Операции не обязаны быть отделены пробелами или символом конца строки от ключевых слов и идентификаторов. Операции могут идти подряд без разделителей. 	оценка 3 — умеет строить автомат-распознаватель по описанию языка, оценка 4 — кроме того умеет строить лексический анализатор и описывать процедуры, оценка 5 — кроме того умеет программно реализовывать построенный анализатор
Уметь строить атрибутные транслирующие грамматики для языков программирования и реализовывать	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Написать атрибутную транслирующую грамматику для следующего выражения: $\text{diff}(E, E1, E2, \dots, EN)$ равного разности наибольшего и наименьшего из значений выражений в скобках. 	оценка 3 — умеет строить атрибутные транслирующие грамматики для простейших языков, оценка 4 —

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
транслятор на их основе		умеет строить атрибутивные транслярующие грамматики для языков программирования высокого уровня, оценка 5 — умеет реализовывать транслятор на ЭВМ
Владеть инструментарием gcc/g++, lex, yacc	<p>Примеры задач для расчетно-графических работ:</p> <ul style="list-style-type: none"> • Разработать интерфейс класса для представления предметной области следующего языка. Реализовать этот класс. Описать язык программирования, основанный на обратной польской записи. Реализовать лексический анализатор и интерпретатор этого языка. Реализовать лексический анализатор языка. Построить грамматику языка. Реализовать синтаксический анализатор языка с помощью LL(1)-грамматики. Построить транслятор языка в язык с обратной польской записью. Предметная область языка состоит из натуральных чисел и строк, реализованных в виде массивов символов типа char. Требуется корректная обработка строк любых алфавитов. Для класса реализовать функции и операции: конкатенацию, выделение подстроки, удаление подстроки, доступ к символам строки, определение длины, сравнения. Реализовать итератор для доступа к элементам. Ввод-вывод реализовать в обычном виде. В языке программирования предусмотреть те же операции, но разработать собственный синтаксис для них. Для натуральных чисел реализовать арифметические операции +, -, *, /, % и сравнения. Язык содержит следующие конструкции: пустой оператор «;», объявление переменных «declare V TYPE, ... ;» до любых других операторов, оператор присваивания «set V, V, ..., V to E ;», цикл «while T do P end ;», цикл с параметром «for each V in E do P end ;», условный оператор «if T then P [else P] end ;», оператор ввода «V <- V <- ... <- cin ;», оператор вывода «E -> E -> ... -> cout ;», метка и переход к метке «L:» и «jump L ;», множественное ветвление «switch(E) when C, C, ...: P when C, C, ...: P ... [otherwise P] end ;» генерация исключений «throw ;», комментарии «<<-- комментарий -->>» может занимать несколько строк. Здесь заглавными буквами обозначены: L — имя метки (идентификатор), P — программа, V — имя переменной (идентификатор), E — выражение, T — тест, TYPE — название типа данных, C — константа. Маленькими буквами записаны ключевые слова, в квадратных скобках записаны ча- 	оценка 3 — владеет инструментарием gcc/g++, оценка 4 — кроме того владеет инструментарием lex, оценка 5 — кроме того владеет инструментарием yacc

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	сти операторов, которые могут отсутствовать. Операторы разделяются одним или несколькими пробельными символами.	
Типовые контрольные задания и/или критерии для проверки индикатора ОПК-2.3		
Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
Знать классы языков программирования	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Процедурные языки. • Объектно-ориентированные языки. • Языки функционального программирования. • Языки, основанные на обратной польской записи. 	оценка 3 — знает основные типы языков, понятие обратной польской записи, оценка 4 — знает, как перевести программу с языка одного типа на язык другого типа, оценка 5 — знает доказательство правильности такого перевода
Уметь строить LR-грамматики и грамматики предшествования, строить по этим грамматикам анализаторы	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Язык L состоит из всевозможных слов вида $w_1w_2\dots w_k$, где каждое слово w_i имеет вид $a^{n_i}bcb^{n_i}$. Здесь n_i — произвольные целые неотрицательные числа, k — произвольное целое положительное число. Постройте LR(1)-грамматику для языка L. По LR(1)-грамматике постройте LR(1)-анализатор. Используя построенный анализатор, определите, какие из слов $x = abcbbcb$ и $y = aabac$ принадлежат языку L. • Постройте грамматику простого предшествования для языка $L = \{ w_1w_2\dots w_k \mid w_i = ba^{n_i}c^{2n_i} \text{ для некоторых } n_i > 0 \}.$ Найдите для неё отношения предшествования. 	оценка 3 — умеет строить LR-грамматику по описанию языка, оценка 4 — кроме того умеет строить восходящий табличный анализатор по LR-грамматике, оценка 5 — кроме того умеет строить грамматику предшествования по описанию языка
Уметь строить LL-грамматики, строить по этим грамматикам нисходящие анализаторы	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Написать LL(1)-грамматику для следующего языка. По LL(1)-грамматике построить МП-автомат, распознающий этот язык, используя метод из теоремы. Язык состоит из всевозможных слов вида $w_1w_2\dots w_k$, где каждое слово w_i имеет вид $a^{n_i}b^{m_i}d(c+a)^{n_i}d$. Здесь k, m_i, n_i — произвольные целые числа, $m_i > 0$. • Постройте управляющую таблицу для LL(2)-грамматики с правилами $S \rightarrow A1B1 \mid 1B \quad A \rightarrow 01 \mid 0B \quad B \rightarrow 2 \mid 2A.$ 	оценка 3 — умеет строить LL-грамматику по описанию языка, оценка 4 — кроме того умеет строить нисходящий анализатор по LL(1)-грамматике, оценка 5 — кроме того умеет строить нисходящий анализа-

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
		тор по LL(k)-грамматике при $k > 1$

V. Учебно-методическое и информационное обеспечение дисциплины

1. Рекомендованная литература

Основная литература:

1. Малявко, А. А. Формальные языки и компиляторы: учебное пособие для вузов / А. А. Малявко. — Москва: Издательство Юрайт, 2021. — 429 с. — (Высшее образование). — ISBN 978-5-534-04288-7. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/472099>
2. Свердлов, С. З. Языки программирования и методы трансляции: учебное пособие для вузов / С. З. Свердлов. — 2-е изд., стер. — Санкт-Петербург: Лань, 2021. — 564 с. — ISBN 978-5-8114-8195-8. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/173116>
3. Молдованова, О. В. Языки программирования и методы трансляции: учебное пособие / О. В. Молдованова. — Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2012. — 134 с. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/54809.html>

Дополнительная литература:

1. Теория и реализация языков программирования: учебное пособие / В. А. Серебряков, М. П. Галочкин, Д. Р. Гончар, М. Г. Фуругян. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 372 с. — ISBN 978-5-4497-0944-8. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/102068.html>
2. Гагарина, Л. Г. Введение в теорию алгоритмических языков и компиляторов: учеб. пособие / Л.Г. Гагарина. Е.В. Кокорева: под ред. Л.Г. Гагариной. — М.: ИД «ФОРУМ». 2018. — 176 с: ил. — (Высшее образование). - ISBN 978-5-8199-0404-6. - Текст: электронный. - URL: <https://znanium.com/catalog/product/929631>

2. Программное обеспечение

Наименование помещений	Программное обеспечение
Ауд. 201а (компьютерная лаборатория ПМиК) (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Перечень программного обеспечения (со свободными лицензиями): Linux Kubuntu, KDE, TeXLive, TeXStudio, LibreOffice, GIMP, Gwenview, ImageMagick, Okular, Skanlite, Google Chrome, KDE Connect, Konversation, KRDC, KTorrent, Thunderbird, Elisa, VLC media player, PulseAudio, KAppTemplate, KDevelop, pgAdmin4, PostgreSQL, Qt, QtCreator, R, RStudio, Visual Studio Code, Perl, Python, Ruby, clang, clang++, gcc, g++, nasm, flex, bison, Maxima, Octave, Dolphin, HTop, Konsole, KSystemLog, Xterm, Ark, Kate, KCalc, Krusader, Spectacle, Vim

3. Современные профессиональные базы данных и информационные справочные системы

- [1] ЭБС «ZNIANIUM.COM» <http://www.znanium.com>
- [2] ЭБС «Университетская библиотека онлайн» <https://biblioclub.ru>
- [3] ЭБС IPRbooks <http://www.iprbookshop.ru>
- [4] ЭБС «Лань» <http://e.lanbook.com>
- [5] ЭБС «Юрайт» <https://urait.ru>
- [6] ЭБС ТвГУ <http://megapro.tversu.ru/megapro/Web>
- [7] Научная электронная библиотека eLIBRARY.RU (подписка на журналы) https://elibrary.ru/projects/subscription/rus_titles_open.asp
- [8] Репозиторий ТвГУ <http://eprints.tversu.ru>

4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- [1] Automata Theory Tutorial, https://www.tutorialspoint.com/automata_theory/index
- [2] Compiler Design Tutorial, https://www.tutorialspoint.com/compiler_design/index.

VI. Методические материалы для обучающихся по освоению дисциплины

Важной составляющей данного раздела РПД являются требования к рейтинг-контролю с указанием баллов, распределенных между модулями и видами работы обучающихся.

Максимальная сумма баллов по учебной дисциплине, заканчивающейся экзаменом, по итогам семестра составляет 60 баллов (30 баллов - 1-й модуль и 30 баллов - 2-й модуль).

Обучающемуся, набравшему 40–54 балла, при подведении итогов семестра (на последнем занятии по дисциплине) в рейтинговой ведомости учета успеваемости и зачетной книжке может быть выставлена оценка «удовлетворительно».

Обучающемуся, набравшему 55–57 баллов, при подведении итогов семестра (на последнем занятии по дисциплине) в графе рейтинговой ведомости учета

успеваемости «Премияльные баллы» может быть добавлено 15 баллов и выставлена экзаменационная оценка «хорошо».

Обучающемуся, набравшему 58–60 баллов, при подведении итогов семестра (на последнем занятии по дисциплине) в графе рейтинговой ведомости учета успеваемости «Премияльные баллы» может быть добавлено 27 баллов и выставлена экзаменационная оценка «отлично». В каких-либо иных случаях добавление премиальных баллов не допускается.

Обучающийся, набравший до 39 баллов включительно, сдает экзамен.

Распределение баллов по модулям устанавливается преподавателем и может корректироваться.

Примеры задач для подготовки к контрольным работам

1. Описать класс для представления строк произвольной длины в виде массива символов. Предусмотреть: конструктор по умолчанию, конструктор копирования, конструктор, принимающий целое число, деструктор, оператор присваивания, операторы потокового ввода >> и вывода <<, операторы конкатенации (+), проверки на пустоту (!), обращения (), доступа к отдельным символам ([]), сравнения == и <, метод поиска подстроки. Выход за границы массива обрабатывать с помощью исключительных ситуаций. Создать итератор для доступа к символам.
2. Постройте конечный автомат для распознавания ключевых слов языка C.
3. Постройте КС-грамматику для языка $L = \{ 0^{i_1} 1^{i_1} 0^{i_2} 1^{i_2} \dots 0^{i_n} 1^{i_n} : n > 0; i_j > 0 \}$.
4. Постройте КС-грамматику для объявления класса на языке C++ (класс содержит только объявления полей, методов и операторов).
5. Постройте LR(1)-грамматику для языка $L = \{ a^i b^j : 0 < i < j \}$. Постройте для нее восходящий табличный анализатор. Используя анализатор, определите, какие из следующих цепочек выводимы в грамматике, и постройте для них правые выводы: *aabbb*, *aaabbbbb*, *aaab*, *aaa*.
6. Для языка из предыдущей задачи постройте LALR(1)-грамматику. Примените анализатор для тех же цепочек.
7. Постройте грамматику простого предшествования для языка $L = \{ a^i b^j : i > j > 0 \}$. Найдите для нее отношения предшествования Вирта-Вебера. Постройте алгоритм типа «перенос-свертка». Определите с его помощью, какие из следующих цепочек выводимы в грамматике, и найдите для них правые выводы: *aaaabb*, *aaaa*, *aaabbb*, *aaab*.
8. Постройте МП-автомат для языка $L = \{ w \in \{0; 1\}^+ : |w|_0 = |w|_1 \}$.
9. Постройте LL(1)-грамматику для объявлений целочисленных переменных и массивов на языке C (в качестве индексов могут использоваться только константы). Постройте по этой грамматике МП-автомат. Определите, какие из следующих цепочек выводимы в грамматике и найдите для них левые выводы: *int v;*, *int v[c][c][c];*, *int v[c][c];*, *int v v;*

10. Преобразуйте грамматику из предыдущей задачи в LL(2)-грамматику без пустых правил. Постройте по ней LL(2)-анализатор. Примените этот анализатор для цепочек из предыдущей задачи.
11. Постройте атрибутивную транслирующую грамматику для выражений вида ADD(E1, E2, ..., EN). Значение такого выражения равно сумме значений выражений в скобках. Результатом трансляции должна быть программа на стековом языке.

Требования к рейтинг контролю (3 семестр)

Контрольная работа 1. Темы: объектно-ориентированное программирование.

Пример задания:

Описать шаблонный класс для представления стеков произвольных элементов в виде односвязных линейных списков. Предусмотреть: конструктор по умолчанию, конструктор копирования, деструктор, оператор присваивания, функции потокового ввода/вывода (операторы <<, >>), функции для помещения элемента в стек, взятия из стека, очистки стека, проверки на пустоту (оператор !), операторы сравнения (операторы ==, !=).

За решение задачи выставляется максимум 15 баллов.

Контрольная работа 2. Темы: лексический анализ. Пример задания:

Постройте лексический анализатор, опишите автомат и выполняемые действия, напишите программу на C++, реализующую лексический анализатор с помощью таблицы указателей на функции. Язык состоит из лексем ! != !! == .= .. и может содержать идентификаторы, состоящие из букв и цифр, и комментарии в виде # комментариев до конца строки. Лексемы разделены пробелами. За решение задачи выставляется максимум 15 баллов.

Расчетно-графическая работа 1. Темы: объектно-ориентированное программирование, лексический анализ, синтаксический анализ, трансляция. Пример задания: 1 семестр. Разработать интерфейс класса для представления предметной области языка LXXIX15O. Реализовать этот класс (15 баллов). При разработке класса запрещено использовать стандартные библиотеки (кроме библиотек ввода/вывода).

Описать язык программирования, основанный на обратной польской записи. Реализовать лексический анализатор (10 баллов) и интерпретатор этого языка (5 баллов).

2 семестр. Реализовать лексический анализатор языка LXXIX15O (5 баллов). Построить грамматику языка LXXIX15O (5 баллов). Реализовать синтаксический анализатор языка LXXIX15O с помощью грамматики простого предшествования (15 баллов). Построить транслятор языка LXXIX15O в язык с обратной польской записью (5 баллов).

Предметная область языка состоит из натуральных чисел и многочленов с вещественными коэффициентами, реализованных в виде списков коэффициентов. Для

класса реализовать функции и операции: арифметические действия, нахождение значения, нахождение производной, доступ к элементам, сравнения, нахождение степени. Реализовать итератор для доступа к элементам. Ввод-вывод реализовать в виде $\{\pm a_0(0)\pm a_1(1)\pm \dots\}$. В языке программирования предусмотреть те же операции, но разработать собственный синтаксис для них. Для натуральных чисел реализовать арифметические операции $+$, $-$, $*$, $/$, $\%$ и сравнения.

Язык LXXIX15O содержит следующие конструкции:

пустой оператор* ;

объявление переменных* $V : \text{TYPE}$; до любых других операторов

оператор присваивания* $\text{set } V, V, \dots, V \text{ to } E$;

цикл while $\text{while } T \text{ do } P \text{ od}$;

цикл с параметром $\text{for } V \text{ from } E \text{ to } E \text{ [by } E \text{] do } P \text{ od}$;

условный оператор* $\text{if } (T) P \text{ [else } p \text{] fi}$;

оператор ввода* $V \leftarrow V \leftarrow \dots \leftarrow \text{cin}$;

оператор вывода* $E \rightarrow E \rightarrow \dots \rightarrow \text{cout}$;

метка и переход к метке* $:L$ и $\text{goto } L$;

множественное ветвление

$\text{select } E \text{ in case } C: P \text{ case } C: P \dots \text{ [otherwise } P \text{] ni}$; выполняется только программа с соответствующей меткой

генерация исключений exception ;

комментарий* $\&$ комментарий $\&$ — комментарий не может содержать $\&$

составной оператор* отсутствует

Звездочкой помечены конструкции обязательные для реализации.

Здесь заглавными буквами обозначены: L — имя метки (идентификатор), P — программа, O — оператор, V — имя переменной (идентификатор), E — выражение, T — тест, TYPE — название типа данных, C — константа, CN — натуральное число. Маленькими буквами записаны ключевые слова. В квадратных скобках записаны части операторов, которые могут отсутствовать.

Операторы разделяются одним или несколькими пробельными символами.

Вопросы к экзамену

1. Объектно-ориентированное программирование.

- 1) Некоторые отличия языка C++ от C: ссылки, перегрузка функций, операторы `new` и `delete`, ключевое слово `auto`, пространства имён, подставляемые функции.
- 2) Классы в языке C++, инкапсуляция, скрытие данных. Поля и методы класса. Открытые и закрытые члены класса. Дружественные функции и классы.
- 3) Статические поля и методы классов.
- 4) Конструкторы и деструкторы. Специальные типы конструкторов (по умолчанию, копирования, перемещения). Делегирующие конструкторы. Удалённые члены класса.
- 5) Перегрузка операций. Копирующее и перемещающее присваивание.
- 6) Наследование в C++. Базовые и производные классы. Защищённые члены класса. Множественное наследование. Виртуальные базовые классы.

- 7) Полиморфизм и виртуальные функции. Таблица виртуальных функций. Чисто виртуальные функции и абстрактные классы.
 - 8) Обработка исключительных ситуаций. Раскрытие стека. Перехват классов. Спецификатор `noexcept`. Аварийное завершение программы.
 - 9) Система ввода/вывода языка C++. Классы потоков. Настройка ввода/вывода. Перегрузка операторов `<<` и `>>`. Манипуляторы. Работа с текстовыми и двоичными файлами.
 - 10) Динамическая идентификация типа. Операторы `typeid` и `decltype`. Создание функций приведения типа.
 - 11) Шаблоны в C++. Шаблоны функции и классы. Шаблоны с переменным числом аргументов.
 - 12) Основы стандартной библиотеки шаблонов STL. Создание итератора. Использование лямбда-функций. Интеллектуальные указатели.
2. Лексический анализ текста.
- 1) Конечные автоматы: распознаватели и преобразователи.
 - 2) Особенности чтения лексем, встречающихся в реальных языках. Обработка идентификаторов и констант.
 - 3) Обработка ключевых слов с помощью вектора альтернатив.
 - 4) Реализация лексического анализатора разными методами: множественное ветвление, таблица указателей.

Общая сумма В сумме за все задачи выставляется не более 60 баллов.

За ответ на экзамене выставляется максимум 40 баллов.

Требования к рейтинг контролю (4 семестр)

Контрольная работа 1. Темы: восходящий синтаксический анализ. Пример задания:

Язык L состоит из всевозможных слов вида $w_1dw_1^{-1}w_2dw_2^{-1}\dots w_kdw_k^{-1}$, где каждое слово w_i имеет вид ac^{2n_i} . Здесь n_i — произвольные целые неотрицательные числа, k — произвольное целое положительное число. Постройте LR(1)-грамматику для языка L . По LR(1)-грамматике постройте LR(1)-анализатор. Используя построенный анализатор, определите, какие из слов $x = accdccaada$ и $y = acdcca$ принадлежат языку L .

За решение задачи выставляется максимум 15 баллов.

Контрольная работа 2. Темы: нисходящий синтаксический анализ. Пример задания:

1. Объявление переменных имеет вид $\text{int } v_1(E_{1,1})\dots(E_{1,k_1}), \dots, v_n(E_{n,1})\dots(E_{n,k_n})$, где v_i — переменные, $E_{i,j}$ — выражения, $n > 0$, $k_{ij} \geq 0$. Выражения состоят из констант, скобок, инфиксной операции $*$ и одноместной префиксной операции `fact`. Постройте LL(1)-грамматику, порождающую объявления переменных. Постройте управляющую таблицу для этой грамматики.

2. Постройте управляющую таблицу для LL(2)-грамматики с правилами

$$S \rightarrow 1B0A \mid 1A0 \quad A \rightarrow 1 \mid 0B \quad B \rightarrow 2A \mid 2B.$$

За решение задачи 1 выставляется максимум 10 баллов. За решение задачи 2 выставляется максимум 5 баллов.

Вопросы к экзамену

1. Контекстно-свободные грамматики.

- 1) Контекстно-свободная грамматика. Вывод в КС-грамматике. Язык, порождаемый КС-грамматикой.
- 2) Левый вывод, правый вывод. Дерево вывода. Однозначная грамматика, однозначный язык.

2. Восходящий разбор.

- 1) Множества $FIRST_k^G$, $FOLLOW_k^G$ и EFF_k^G .
- 2) Алгоритмы типа «перенос-свёртка»
- 3) LR(k)-грамматика.
- 4) LR(k)-ситуация, активный префикс. Допустимая для активного префикса LR(k)-ситуация.
- 5) Функции V_k^G и GOTO. Каноническая система множеств LR(k)-ситуаций. LR(k)-таблица, каноническая система LR(k)-таблиц.
- 6) Канонический LR(k)-анализатор. Алгоритм для его построения.
- 7) LALR(k)-грамматика.
- 8) Отношения предшествования Вирта-Вебера. Обратимая грамматика. Грамматика предшествования. Грамматика простого предшествования.

3. Нисходящий разбор.

- 1) Автомат с магазинной памятью. Отношение перехода за один шаг для МП-автомата. Язык, распознаваемый МП-автоматом.
- 2) LL(k)-грамматика.
- 3) Построение управляющей таблицы для LL(1)-грамматики.
- 4) LL(k)-таблица. Построение управляющей таблицы для LL(k)-грамматики.
- 5) Устранение пустых правил в LL(k)-грамматике.

4. Трансляция.

- 1) Транслирующая грамматика и задаваемый ей перевод.
- 2) Атрибутная транслирующая грамматика.
- 3) Синтезируемые и наследуемые атрибуты.
- 4) S-атрибутные и L-атрибутные грамматики.
- 5) Трансляция при восходящем и нисходящем разборе.

Общая сумма В сумме за все задачи выставляет не более 60 баллов.

За ответ на экзамене выставляется максимум 40 баллов.

VII. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для аудиторной работы

Наименование помещений	Материально-техническое оснащение помещений
Ауд. 304 (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели, экран, комплект аудиотехники (радиосистема, стационарный микрофон с настольным держателем, усилитель, микшер, акустическая система), проектор, ноутбук.
Ауд. 310 (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели, меловая доска.
Ауд. 206 (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели, экран, проектор.

Для самостоятельной работы

Наименование помещений	Материально-техническое оснащение помещений
Ауд. 201а (компьютерная лаборатория ПМиК) (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели, доска маркерная, компьютер, сервер (системный блок), концентратор сетевой.

VIII. Сведения об обновлении рабочей программы дисциплины

№ п/п	Обновленный раздел рабочей программы дисциплины	Описание внесённых изменений	Дата и протокол заседания кафедры, утвердившего изменения
1	11. 2) Программное обеспечение	Внесены изменения в список ПО	От 24.08.2023 года, протокол № 1 ученого совета факультета
2	V. 1) Рекомендуемая литература	Обновление ссылок на литературу	От 24.08.2023 года, протокол № 1 ученого совета факультета