МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Кубанский государственный университет»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра вычислительных технологий

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.08 «МЕТОДЫ РАЗРАБОТКИ ТРАНСЛЯТОРОВ»

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии
(код и наименование направления подготовки/специальности)

Направленность (профиль) " Вычислительные технологии"
(наименование направленности (профиля) специализации)

Программа подготовки академическая
(академическая /прикладная)

Форма обучения очная
(очная, очно-заочная, заочная)

Квалификация (степень) выпускника бакалавр
(бакалавр, магистр, специалист)

Краснодар 2018

Рабочая программа Б1.В.08 «Методы разработки трансляторов» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии»

Программу составил(и):
А.И.Миков, заведующий кафедрой вычислительных технологий, д.ф.-м.н., профессор

подпись

Рабочая программа дисциплины Б1.В.08 «Методы разработки трансляторов» утверждена на заседании кафедры вычислительных технологий
протокол № 7 «03 » апреля 2018 г.

Заведующий кафедрой (разработчика) Миков А.И. 

фамилия, инициалы подпись

Рабочая программа обсуждена на заседании кафедры вычислительных технологий
протокол № 7 «03 » апреля 2018 г.

Заведующий кафедрой (выпускающей) Миков А.И. 

фамилия, инициалы подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики
протокол № 1 от «20» апреля 2018 г

Председатель УМК факультета  К.В. Малыхин

Рецензенты:

Гаркуша О.В., доцент кафедры информационных технологий ФБГОУ ВО «Кубанский государственный университет», кандидат физико-математических наук.

Зайков В.П. Ректор НЧОУ ВО «Кубанский институт информзащиты» д.экон. наук, к.т.н., доцент.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1 Цель освоения дисциплины

Целью преподавания и изучения дисциплины «Методы разработки трансляторов» является овладение студентами методами разработки компиляторов и интерпретаторов, алгоритмов анализа текста и синтеза кода, оптимизации кода.

1.2. Задачи дисциплины

Основные задачи освоения дисциплины.

Студент должен **знать** основные понятия, методы и средства описания языков программирования, алгоритмы анализа; **уметь** применять методы, алгоритмы и программные средства для создания трансляторов; **владеть** инструментальными средствами разработки трансляторов.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Курс «Методы разработки трансляторов» относится к вариативной части блока Б1. Для изучения дисциплины необходимо знание методов программирования, дискретной математики, архитектуры вычислительных систем теории формальных языков и грамматик. Знания, получаемые при изучении курса, используются при изучении программистских дисциплин цикла учебного плана бакалавра.

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы.

Процесс освоения дисциплины направлен на формирование следующих компетенций:

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	**уметь**	**владеть**
1.	ПК-8	Способностью применять на практике международные и профессиональные стандарты информационных технологий, современные парадигмы и методологии, инструментальные и вычислительные средства.	международные и профессиональные стандарты представления данных	анализировать синтаксическую и семантическую организацию современных языков программирования	инструментальными и вычислительными средствами разработки трансляторов

2. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 4 зач.ед. (144 часа), их распределение по видам работ представлено в таблице *(для студентов ОФО)*

Вид учебной работы	Всего часов	Семестры (часы)	
		6	
Контактная работа в том числе:			
Аудиторные занятия (всего):	84	84	
В том числе:			
Занятия лекционного типа	34	34	
Занятия семинарского типа (семинары, практ. занятия)			
Лабораторные занятия	50	50	
Иная контрольная работа			
Контроль самостоятельной работы	2	2	
Промежуточная аттестация (ИКР)	0,3	0,3	
Самостоятельная работа, в том числе			
В том числе:			
Курсовая работа			
Проработка учебного (теоретического) материала	10	10	
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)	12	12	
Реферат			
Подготовка к текущему контролю			
Контроль:			
Подготовка к экзамену:	35,7	35,7	
Общая трудоемкость час	144	144	
в т.ч. контактная работа	86,3	86,3	
зач. ед.	4	4	

2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы дисциплины, изучаемые в _6__ семестре *(очная форма)*

№	Наименование разделов	Количество часов					
		Всего	Аудиторная работа				Внеаудиторная работа
			Л	КСР	ИКР	ЛР	СРС
1	2	3	4	5		6	7
1.	Грамматики языков	29,1	10	1	0,1	10	8
2.	Этапы анализа исходного текста	38,1	12		0,1	20	6
3.	Этапы синтеза кода	41,1	12	1	0,1	20	8
	Итого:	108,3	34	2	0,3	50	22
	экзамен	35,7					
	Итого по дисциплине:	144					

Примечание: Л – лекции, КСР – контрольные и самостоятельные работы, ЛР – лабораторные занятия, СРС – самостоятельная работа студента, Д-доклад, РГЗ – расчетно-графическое задание.

2.3 Содержание разделов дисциплины:

2.3.1 Занятия лекционного типа

раздела	Наименование раздела	Содержание раздела	Форма текущего контроля	Разработано с участием представи-телей работода-телей
1	2	3	4	5
1	Грамматики языков	Формальные языки. Префиксное (суффиксное) свойство языка. Операции над формальными языками, гомоморфизм. Грамматики, примеры. Классификация Хомского. Языки, порождаемые грамматиками. Примеры грамматик классов 2, 1, 0. Эквивалентные грамматики. Понятие вывода в грамматике. Выводимые цепочки, терминальные цепочки. Распознаватели языков. Конфигурации. Классы распознавателей. Распознаватели автоматных языков. Недетерминированный конечный автомат. Пример. Детерминированный КА. Построение детерминированного КА по недетерминированному. КС-языки. Примеры формальных КС-грамматик. Деревья вывода (разбора) Крона дерева. Сечение дерева, крона сечения. Левый и правый вывод. Алгоритм проверки непустоты языка. Нормальная форма Хомского. Рекурсивные грамматики. Нормальная форма Грейбах. Автомат с магазинной памятью. Конфигурация МП-автомата. Пример МП-автомата. Детерминированный МП-автомат. Общие методы синтаксического анализа КС-языков. Нисходящий разбор с возвратами. Восходящий разбор. Сложность общих методов. Табличные методы анализа. Алгоритм Кока-Янгера-Касами. Построение таблицы разбора. Построение разбора	ЛР	

		по таблице. Пример. Однопроходный синтаксический анализ без возвратов. Синтаксический анализ LL(*k*)-языков. LL(*k*)-грамматика. Множества FIRST и FOLLOW. LL(1)-языки. *k*-предсказывающий алгоритм разбора. Управляющая таблица. Построение разбора по таблице. Пример. Построение управляющей таблицы для *k*-предсказывающего алгоритма разбора. Пример.		
2	Этапы анализа исходного текста	Основы трансляции. Компиляторы и интерпретаторы. Методы получения объектного кода (абсолютный и переместимый код, преобразование в ассемблерный код, в байт-код, компиляция «на лету»). Схема компилятора. Основные фазы компиляции. Просмотры. Пример фрагмента программы и его преобразований на разных фазах. Лексический анализ. Заглядывание вперед при лексическом анализе. Таблицы трансляторов. Структуры таблиц. Способы использования идентификаторов. Описатель процедуры. Коды типов. Дескрипторы типов различных типов данных. Методы построения функций (процедур) синтаксического анализа для LL(1)-языков. Вспомогательные функции. Анализ конструкций ⟨программа⟩, ⟨блок⟩, ⟨оператор while⟩. Примеры на C или Паскале. Методы построения функций (процедур) синтаксического анализа для правил, содержащих { }. Анализ конструкций ⟨составной оператор⟩, ⟨переменная⟩, ⟨выражение⟩. Примеры на C или Паскале. Нейтрализация синтаксических ошибок в процессе трансляции. Семантический анализ программы при трансляции. Виды семантических ошибок. Контроль типов. Идентификация идентификаторов. Атрибутное дерево разбора.	ЛР, РГЗ	
3	Этапы синтеза кода	Промежуточные формы представления программы. Обратная польская запись. Перевод в ОПЗ. Исполнение ОПЗ. Трехадресный код как промежуточная форма программы. Тетрады, триады.	ЛР, РГЗ	

		Синтаксически управляемый перевод. Схема. Процесс перевода. Пример. Разнообразие систем команд ЭВМ. Перевод и генерация кода для арифметических выражений. Генерация кода для 80x86. Генерация кода для операторов FOR, WHILE, REPEAT. Промежуточный язык. ANDF. Платформа Java. Общая инфраструктура языков. Составные части спецификации CLI. Общая система типов в CLI. Виртуальная система выполнения. Генерация кода в .NET. Common Language Runtime, .NET Framework Class Library, JIT-компиляция, виртуальная машина, CIL. Оптимизирующие трансляторы. Задачи оптимизации. Виды оптимизации, зависимость между оптимизациями. Стадии оптимизации. Покадровая оптимизация. Примеры оптимизационных преобразований. Удаление мертвого кода. Упрощение выражений. Свертка констант. Экономия общих подвыражений. Оптимизация. Объединение циклов. Раскрутка циклов. Чистка циклов. Вложенные циклы. Понижение силы операций.		

2.3.2. Занятия семинарского типа

Учебным планом не предусмотрены

2.3.3 Лабораторные работы

№ работы	№ раздела дисциплины	Наименование лабораторных работ
1	1	Генерация слов на основе грамматики
2	1	Разработка распознавателей автоматных грамматик
3	1	Анализ свойств контекстно-свободных грамматик
4	1	Контекстно-зависимые грамматики
5	1	Итоговое обсуждение по разделу
6	2	Таблицы идентификаторов
7	2	Обратная польская запись
8	2	Синтаксический анализ

9	2	Табличные методы синтаксического анализа
10	2	Восходящий анализ
11	2	LR-анализ
12	2	Синтаксически управляемый анализ
13	2	Контекстный анализ программы
14	2	Семантический анализ программы
15	2	Промежуточная форма программы
16	3	Построение таблиц транслятора
17	3	Генерация кода для линейных участков
18	3	Генерация кода для ветвлений
19	3	Локальная оптимизация при трансляции
20	3	Глобальная оптимизация при трансляции
21	3	Отладка трансляторов
22	3	Тестирование трансляторов
23	3	Оформление программной документации
24	3	Конкурс трансляторов
25	3	Обсуждение результатов

ПЛАНЫ ОТДЕЛЬНЫХ ЛАБОРАТОРНЫХ РАБОТ

Занятие 1. Генерация слов на основе грамматики.

Разработка программы на Паскале, реализующей генерацию слов на основе грамматики автоматного типа.

Вход: текстовый файл – строка, содержащая правила грамматики, разделенные точкой с запятой.

Выход: слово из терминальных символов, принадлежащее языку, описываемому введенной грамматикой.

Условия: алфавит A – малые латинские буквы; множество N – заглавные латинские буквы; S – начальный символ грамматики. При наличии нескольких правил с одинаковыми левыми частями при генерации очередного символа правило выбирается случайным равновероятным образом.

Занятие 2. Разработка распознавателей автоматных грамматик.

1) Разработка программы на Паскале, реализующей автомат, принимающий слова, заданные автоматной грамматикой или регулярным выражением, например, 1{01 | 101}0 или 01{101}11.

2) Разработка программы, вычисляющей тип грамматики (3, 2, 1, 0) по текстовой строке, содержащей правила.

Занятие 3. Анализ свойств КС-грамматик

Разработка программы на Паскале, анализирующей свойства КС-грамматики: является ли она леворекурсивной, праворекурсивной, непустой, задана ли в нормальной форме Хомского, Грейбах.

Вход: текстовый файл – строка, содержащая правила грамматики, разделенные точкой с запятой.

Выход: текстовые сообщения о свойствах грамматики.

Занятие 6. Таблицы идентификаторов

Разработка программы на Паскале, осуществляющей создание и заполнение таблицы идентификаторов.

Вход: текстовый файл некоторой «программы», содержащий идентификаторы.

Выход: хэш-таблица или дерево.

Условия: алфавит *A* содержит все символы клавиатуры, кроме специальных; идентификатор – последовательность букв и цифр, начинающаяся с буквы; в тексте «программы» может быть любое количество идущих подряд пробелов; «программа» заканчивается символами **end** и точка.

Занятие 7. Обратная польская запись

Разработка программы на Паскале, переводящей выражения из стандартной математической формы в обратную польскую запись.

Вход: текстовый файл – алгебраическое выражение.

Выход: текстовый файл – обратная польская запись выражения.

Условия: выражение содержит только однобуквенные переменные и не содержит числовых констант; кроме букв могут содержаться символы **+ – * / ^ ( ).**

Занятие 8. Синтаксический анализ

Разработка программы на Паскале, реализующей синтаксический анализ для небольшого LL(1) языка.

Вход: 1) текстовый файл – грамматика LL(1) языка;

2) слово *w*.

Выход: сообщение о том, принадлежит ли слово *w* языку *L*.

Условия: грамматика языка *L* – формальная (подобно заданиям занятий 1 и 2); слово *w* состоит только из малых латинских букв.

2.3.3. Расчетно-графические задания

По дисциплине студентом выполняется одно комплексное индивидуальное расчетно-графическое задание. Темы заданий для каждого студента различны.

Задача РГЗ состоит в проверке умений студента и проверки эффективности его самостоятельной работы.

Темы заданий ежегодно обновляются. Общая тематика соответствует тематике лабораторных работ.

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	**Раздел 1.** Конечные автоматы, их описание и свойства.	Основная литература [1,2] Дополнительная литература [1-4,8-11]
2	**Раздел 2.** LR-анализ.	Основная литература [3] Дополнительная литература [2-5,9]
3	**Раздел 3.** Методы оптимизации при трансляции.	Основная литература [4] Дополнительная литература [6-8]

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа,

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа,

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Семестр	Вид занятия (Л, ПР, ЛР)	Используемые интерактивные образовательные технологии	Количество часов
6	Л	Компьютерные презентации и обсуждение	34
	ЛР	Разбор конкретных ситуаций (задач), тренинги по решению задач, компьютерные симуляции (программирование алгоритмов)	50
Итого:			84

4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1 Фонд оценочных средств для проведения текущего контроля

Фонд оценочных средств дисциплины состоит из средств текущего контроля выполнения заданий, лабораторных работ, средств для итоговой аттестации (экзамена в 6 семестре).

Оценка успеваемости осуществляется по результатам:

- выполнения лабораторных работ;

- оценки, выставляемой при сдаче индивидуального расчетно-графического задания;

- ответа на экзамене (для выявления знания и понимания теоретического материала дисциплины).

Образец РГЗ – задания на разработку транслятора

Целью работы является реализация интерпретатора или компилятора с простого языка программирования. Реализуемый язык должен содержать скалярные переменные и константы по крайней мере двух типов (один из них целый), одномерные массивы, а также следующие виды операторов: присваивания и формулы, условные операторы, циклы с условием, вложенные конструкции, описания и вызовы процедур с подстановкой параметров (по значению и ссылке), в том числе параметров-массивов с подстановкой по ссылке. При этом должны корректно транслироваться и исполняться рекурсивные процедуры.

Типовое задание:

1. Название работы:
Разработка языка и компилятора для (название языка *L*).
2. Характеристика языка *L*:
Язык программирования императивного или декларативного типа.
Основная структура данных языка (определение структуры данных *D*).
Для структуры *D* в программе могут вводиться переменные и константы. Константы задаются выделенными идентификаторами. Переменные задаются идентификаторами, не совпадающими с обозначениями констант. Для переменных предусматривается конструкция языка ⟨описание переменных⟩.
Для структур *D* предусматривается оператор присваивания, оператор ввода и оператор вывода. Кроме перечисленных, в языке должен быть оператор ветвления и оператор цикла. Для организации вычислений предусматривается вспомогательный тип данных – целый.
Над структурами могут выполняться операции сравнения (= или ≠), специфические операции сравнения, а также операции преобразования значений. Набор операций (не менее 4) предусматривается минимальный, но такой, чтобы с их помощью можно было построить любое значение типа *D* или решать определенные в названии задачи работы.
3. Описание языка *L*:
Основная часть правил языка должна быть описана в форме БНФ или синтаксических диаграмм. Остальные правила – в свободной форме.
4. Компилятор языка *L*:
Для языка *L* должен быть спроектирован и реализован компилятор *C*. Компилятор должен принимать на входе текст программы на языке *L* и генерировать соответствующий этому тексту код *Code* на языке ассемблера Intel.

Минимальные требования к компилятору.

Входная программа может иметь длину до 1 тысячи лексем и содержать до 10 идентификаторов и констант. При наличии в программе ошибок компилятор должен выдавать сообщения об ошибках (не менее 5 различных сообщений о синтаксических ошибках и 2 сообщений о семантических ошибках).

Нейтрализации ошибок может не производиться, но на неправильной программе компилятор не должен «зацикливаться».

Оптимизация кода не производится.

Язык реализации компилятора – Паскаль или Си.

5. Демонстрация работы компилятора (итоговая сдача индивидуального задания):

Преподаватель задает до 3 тестовых примеров – программ на языке *L*. Компилятор *C* должен откомпилировать каждый из примеров, а программа *P* – выполнить их так, чтобы в файле вывода были сформированы правильные результаты.

6. Содержание итогового отчета:

1) постановка задачи;
2) описание языка;
3) исходный текст компилятора;
4) тестовые примеры и результаты тестирования программы;
5) список использованной литературы.

Перечень вопросов, которые выносятся на экзамен в 6 семестре

1. Математическое моделирование языков. Формальные языки. Префиксное (суффиксное) свойство языка. Операции над формальными языками, гомоморфизм.
2. Грамматики, примеры. Классификация Хомского. Языки, порождаемые грамматиками. Примеры грамматик классов 2, 1, 0. Эквивалентные грамматики.
3. Понятие вывода в грамматике. Выводимые цепочки, терминальные цепочки. Распознаватели языков. Конфигурации. Классы распознавателей.
4. Распознаватели автоматных языков. Недетерминированный конечный автомат. Пример. Детерминированный КА. Построение детерминированного КА по недетерминированному.
5. КС-языки. Примеры формальных КС-грамматик. Деревья вывода (разбора) Крона дерева. Сечение дерева, крона сечения.
6. Левый и правый вывод. Алгоритм проверки непустоты языка. Нормальная форма Хомского.
7. Рекурсивные грамматики. Нормальная форма Грейбах. Автомат с магазинной памятью. Конфигурация МП-автомата. Пример МП-автомата. Детерминированный МП-автомат.
8. Общие методы синтаксического анализа КС-языков. Нисходящий разбор с возвратами. Восходящий разбор. Сложность общих методов.
9. Табличные методы анализа. Алгоритм Кока-Янгера-Касами. Построение таблицы разбора. Построение разбора по таблице. Пример.
10. Однопроходный синтаксический анализ без возвратов. Синтаксический анализ LL(k)-языков. LL(k)-грамматика. Множества FIRST и FOLLOW. LL(1)-языки.
11. k-предсказывающий алгоритм разбора. Управляющая таблица. Построение разбора по таблице. Пример.
12. Построение управляющей таблицы для k-предсказывающего алгоритма разбора. Пример.
13. Основы трансляции. Компиляторы и интерпретаторы. Методы получения объектного кода (абсолютный и переместимый код, преобразование в ассемблерный код, в байт-код, компиляция «на лету»). Схема компилятора. Основные фазы компиляции. Просмотры. Пример фрагмента программы и его преобразований на разных фазах.
14. Лексический анализ. Заглядывание вперед при лексическом анализе. Таблицы трансляторов. Структуры таблиц. Способы использования идентификаторов. Описатель процедуры. Коды типов. Дескрипторы типов различных типов данных.

15. Методы построения функций (процедур) синтаксического анализа для LL(1)-языков. Вспомогательные функции. Анализ конструкций ⟨программа⟩, ⟨блок⟩, ⟨оператор while⟩. Примеры на C или Паскале.
16. Методы построения функций (процедур) синтаксического анализа для правил, содержащих { }. Анализ конструкций ⟨составной оператор⟩, ⟨переменная⟩, ⟨выражение⟩. Примеры на C или Паскале.
17. Нейтрализация синтаксических ошибок в процессе трансляции.
18. Семантический анализ программы при трансляции. Виды семантических ошибок. Контроль типов. Идентификация идентификаторов. Атрибутное дерево разбора. Промежуточные формы представления программы. Обратная польская запись. Перевод в ОПЗ. Исполнение ОПЗ.
19. Трехадресный код как промежуточная форма программы. Тетрады, триады.
20. Синтаксически управляемый перевод. Схема. Процесс перевода. Пример.
21. Разнообразие систем команд ЭВМ. Перевод и генерация кода для арифметических выражений. Генерация кода для 80x86. Генерация кода для операторов FOR, WHILE, REPEAT.
22. Промежуточный язык. ANDF. Платформа Java. Общая инфраструктура языков. Составные части спецификации CLI. Общая система типов в CLI. Виртуальная система выполнения.
23. Генерация кода в .NET. Common Language Runtime, .NET Framework Class Library, JIT-компиляция, виртуальная машина, CIL.
24. Оптимизирующие трансляторы. Задачи оптимизации. Виды оптимизации, зависимость между оптимизациями. Стадии оптимизации. Покадровая оптимизация.
25. Примеры оптимизационных преобразований. Удаление мертвого кода. Упрощение выражений. Свертка констант. Экономия общих подвыражений.
26. Оптимизация. Объединение циклов. Раскрутка циклов. Чистка циклов. Вложенные циклы. Понижение силы операций.
27. Сети Петри. Определение. Графическое изображение. Маркировка. Выполнение. Пример: Функционирование простой системы обработки данных.
28. Сети Петри. Состояние. Функция следующего состояния. Последовательность запусков переходов. Примеры сети Петри: задача о взаимном исключении, моделирование fork и join.
29. Сети Петри. Непосредственно достижимая маркировка. Множество достижимости. Пример. Задача об обедающих мудрецах.
30. Понятие (неформальное) последовательного процесса. События. Префиксы. Рекурсия. Выбор. Примеры. Законы для оператора выбора и рекурсии процессов.
31. Протокол поведения процесса. Примеры. Операция конкатенации и ее законы. Операция сужения для протоколов и ее законы, операция «голова и хвост», ее законы.
32. Операция для протоколов «длина» и ее законы. Операция над протоколами «порядок» и ее законы. Монотонные функции.
33. Множество протоколов процесса. Примеры. Законы. Операция «после» над протоколами и ее законы. Примеры.
34. Понятие спецификации процесса. Примеры. Соответствие спецификации. Отношение «удовлетворяет». Законы.
35. Параллельные процессы. Взаимодействие. Примеры. Законы. Протоколы параллельных процессов. Законы.
36. Математическое определение детерминированного процесса. Определения D0-D7.
37. Теория неподвижной точки. Отношение порядка на множестве процессов (D1). Законы. Понятие цепи в частичном упорядочении. Полный частичный порядок. Законы.
38. Непрерывная функция из одного полного частичного порядка в другой. Композиция непрерывных функций. Законы для непрерывных функций. Теорема о неподвижной точке и ее доказательство.

Критерии оценивания

Экзаменационный билет состоит из 3 разделов (вопросов):
1. Вопрос по языкам и трансляции (33 балла).
2. Вопрос по процессам (33 балла).
3. Практическое задание (34 балла).

Перевод баллов в стандартные оценки:
«отлично» - от 100 до 75 баллов;
«хорошо» - от 74 до 50 баллов;
«удовлетворительно» - от 49 до 25 баллов;
«неудовлетворительно» - менее 25 баллов

Практическое задание (индивидуальное задание) принимается совместно экзаменатором и преподавателем, ведущим лабораторные занятия, в период зачетной недели. По этому пункту выставляется от 0 до 34 баллов. Полученный «зачет» по курсу означает, что раздел 3 оценивается как минимум в 10 баллов.

«отлично» на экзамене за 3 лучших транслятора, удовлетворяющих всем требованиям задания. Если компилятор не вошел в тройку лучших, но удовлетворяет всем требованиям, то на экзамене студент отвечает только на вопросы по темам 7 и 8; вопросы по остальным темам засчитываются как отличные.
Если компилятор удовлетворяет не всем требованиям, то экзамен сдается в полном объеме.

Оценка индивидуального задания:

Полностью выполненное задание – 34 балла;
генерируется код, но не всегда верно исполняется – 34 – 30 баллов;
генерируется правильный код, но не исполняется – 29 -25 баллов;
генерируется не всегда верный код – 24 – 20 баллов;
код не генерируется, семантический анализ проводится – 19 – 15 баллов;
код не генерируется, синтаксический анализ проводится – 14 – 10 баллов;
синтаксический анализ не всегда верно проводится – менее 10 баллов.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.
– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;
– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;
– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.
Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:
Для лиц с нарушениями зрения:
– в печатной форме увеличенным шрифтом,
– в форме электронного документа.
Для лиц с нарушениями слуха:
– в печатной форме,
– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:
– в печатной форме,
– в форме электронного документа.
Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

5.1 Основная литература:

Основная литература:

1. Миков А.И. Распределенные компьютерные системы и алгоритмы. Учебное пособие. – Краснодар. Изд-во КубГУ, 2009. (37 экз. в библиотеке КубГУ).
2. Дроздов, С.Н. **Структуры** и **алгоритмы** обработки **данных** : учебное пособие / С.Н. Дроздов ; Министерство образования и науки РФ, Южный федеральный университет, Инженерно-технологическая академия. - Таганрог : Издательство Южного федерального университета, 2016. - 228 с. : схем., ил. - Библиогр. в кн. - ISBN 978-5-9275-2242-2 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=493032
3. Малявко, А.А. Формальные языки и **компилятор**ы : учебное пособие / А.А. Малявко. - Новосибирск : НГТУ, 2014. - 431 с. : табл., схем. - (Учебники НГТУ). - Библиогр. в кн. - ISBN 978-5-7782-2318-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=436055

Дополнительная литература:

1. Ануфриенко, А.В. Введение в оптимизацию приложений с использованием **компиляторов** Intel : лекции / А.В. Ануфриенко, Р.И. Идрисов. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 230 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428836
2. Алексеев, В.Е. Структуры данных. Модели вычислений / В.Е. Алексеев, В.А. Таланов. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 248 с. : схем., ил. - (Основы информационныхтехнологий). - Библиогр. в кн. - ISBN 5-9556-0066-3 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428782

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах
1. ЭБС Издательства «Лань» http://e.lanbook.com ,
2. ЭБС «Университетская библиотека онлайн» www.biblioclub.ru ,
3. ЭБС «Юрайт» http://www.biblio-online.ru ,
4. ЭБС «ZNANIUM.COM» www.znanium.com,
5. ЭБС «BOOK.ru» https://www.book.ru.

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля).

7. Методические указания для обучающихся по освоению дисциплины

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал, лабораторных работ, контрольной работы, зачета и экзамена.

Важнейшим этапом курса является самостоятельная работа по дисциплине с использованием указанных литературных источников и методических указаний автора курса.

Виды и формы СР, сроки выполнения, формы контроля приведены выше в данном документе.

Для лучшего освоения дисциплины при защите ЛР студент должен ответить на несколько вопросов из лекционной части курса.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

8.1 Перечень информационных технологий.

– Проверка домашних заданий и консультирование посредством электронной почты.

– Использование электронных презентаций при проведении лекций и практических занятий.

8.2 Перечень необходимого программного обеспечения

1. MS Office.
2. Free Pascal.
3. Компилятор языка Си.
4. Компилятор языка C#.
5. Assembler.

8.3 Перечень информационных справочных систем:

1. Электронная библиотечная система eLIBRARY.RU (http://www.elibrary.ru)/

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

№	Вид работ	Материально-техническое обеспечение дисциплины
1.	Лекционные занятия	Лекционная аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) и соответствующим программным обеспечением (ПО) PowerPoint. ауд. 129, 131, А305.
2.	Лабораторные занятия	Лаборатория, укомплектованная специализированными техническими средствами обучения – компьютерный класс, с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета. (лаб. 102-106.).
3.	Групповые (индивидуальные) консультации	Аудитория, (кабинет) – компьютерный класс
4.	Текущий контроль, промежуточная	Аудитория, приспособленная для письменного ответа при промежуточной аттестации.

	аттестация	
5.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.