

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный экономический университет (РИНХ)»

УТВЕРЖДАЮ
Директор Таганрогского института
имени А. П. Чехова (филиала)
РГЭУ (РИНХ)
_____ С. А. Петрушенко
«25» мая 2026 г.

Рабочая программа дисциплины
Теория алгоритмов

Направление подготовки
09.03.03 Прикладная информатика

Направленность (профиль) программы бакалавриата
09.03.03.02 Разработка программного обеспечения

Для набора 2026 года

Квалификация
Бакалавр

КАФЕДРА информатики**Распределение часов дисциплины по семестрам / курсам**

Курс	4		5		Итого	
Вид занятий	УП	РП	УП	РП		
Лекции	4	4			4	4
Лабораторные	4	4	2	2	6	6
Итого ауд.	8	8	2	2	10	10
Контактная работа	8	8	2	2	10	10
Сам. работа	64	64	30	30	94	94
Часы на контроль			4	4	4	4
Итого	72	72	36	36	108	108

ОСНОВАНИЕ

Учебный план утвержден учёным советом вуза от 03.03.2026, протокол № 9.

Программу составил(и): канд. техн. наук, Доц., Белоконова Светлана Сергеевна

Зав. кафедрой: Тюшнякова И. А.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	обеспечение студентов базовыми знаниями в области алгоритмической логики, принципов проектирования программного обеспечения компьютерной техники, формальной теории вычислимости, частично-рекурсивных функций, теории сложности, а также формирование навыков абстрактного мышления и решения практических задач, связанных с формализацией и алгоритмизацией процессов получения и переработки информации.
-----	--

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

ОПК-1:	Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности
ОПК-1.1:	Знает основы математики, физики, вычислительной техники и программирования
ОПК-1.2:	Умеет решать стандартные профессиональные задачи с применением естественнонаучных и обще-технических знаний, методов математического анализа и моделирования
ОПК-1.3:	Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности
ПКО-1:	Способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение
ПКО-1.1:	Умеет проводить формализацию и алгоритмизацию поставленных задач
ПКО-1.2:	Пишет программный код с использованием языков программирования, определения и манипулирования данными в соответствии с установленными требованиями
ПКО-1.3:	Проверяет и выполняет отладку программного кода

В результате освоения дисциплины обучающийся должен:

Знать:

- основные методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности (соотнесено с индикаторами ОПК-1.1);
- основы построения правильного логического вывода на основе схем формализации вычислительных процедур на естественном языке, аксиомы и формулы преобразования исчисления высказываний (соотнесено с индикаторами ОПК-1.1, ПКО-1.1);
- ключевые понятия и определения теории алгоритмов, формальной теории вычислимости, правила и этапы полного построения алгоритмов, способы их записи и реализации в различных алгоритмических системах (соотнесено с индикаторами ОПК-1.1, ПКО-1.1);
- способы и правила записи алгоритмов (соотнесено с индикаторами ОПК-1.1, ПКО-1.1);
- критерии выявления алгоритмически неразрешимых, легко и трудно разрешимых проблем, оценки мер сложности алгоритмов (соотнесено с индикаторами ОПК-1.1);
- основные возможности современного прикладного программного обеспечения для решения вычислительных задач (соотнесено с индикаторами ОПК-1.1).

Уметь:

- производить формализацию и постановку задачи построения алгоритма с заданными свойствами (соотнесено с индикаторами ОПК-1.2, ПКО-1.1);
- корректно и компактно отображать алгоритмы, используя различные формы записи: словесный, графический, блок-схемный, операторный (соотнесено с индикаторами ОПК-1.2, ПКО-1.2);
- получать частично-рекурсивные, примитивно-рекурсивные и общерекурсивные функции (соотнесено с индикаторами ОПК-1.2);
- решать задачи по составлению абстрактных машин, реализующих заданные вычислимые функции (соотнесено с индикаторами ОПК-1.2, ПКО-1.2);
- применять принцип логического программирования, элементы алгоритмической логики для решения практических задач (соотнесено с индикаторами ОПК-1.2, ПКО-1.2);
- производить временную и емкостную оценку сложности алгоритма, выявлять «узкие» места алгоритмов, а также производить сравнительную характеристику сложности различных алгоритмов (соотнесено с индикаторами ОПК-1.2);
- применять различные математические методы при формализации решения вычислительных задач, использовать современного прикладного программного обеспечения для моделирования процессов и явлений, их теоретического и экспериментального исследования (соотнесено с индикаторами ОПК-1.2, ПКО-1.1);

Владеть:

- разработки, внедрения и использования прикладного программного обеспечения для решения вычислительных задач (соотнесено с индикаторами ОПК-1.3, ПКО-1.2, ПКО-1.3);
- выявления алгоритмически разрешимых и неразрешимых задач и проблем (соотнесено с индикаторами ОПК-1.3, ПКО-1.2, ПКО-1.3);
- формализации вычислительных задач, применения принципа логического программирования и составления алгоритмов решения различных задач с оценкой их временной и емкостной сложности (соотнесено с индикаторами ОПК-1.3, ПКО-1.3);
- реализации различных видов алгоритмов в абстрактных алгоритмических системах и машинах (соотнесено с индикаторами ОПК-1.3, ПКО-1.2, ПКО-1.3).

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**Раздел 1. Основы теории алгоритмов**

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
1.1	Основные понятия и определения. Свойства и виды алгоритмов. Основные направления развития и приложения теории алгоритмов. Краткая характеристика способов записи алгоритмов: блок-схемный, словесный, операторные схемы Ляпунова. Правила записи алгоритмов. Принципы построения алгоритмов - поэтапной детализации, ограниченного числа структур, модульный принцип. Типовые и запрещенные конструкции. Этапы полного построения алгоритмов.	Лекционные занятия	4	2	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.2	Правила и способы записи алгоритмов. Исследование запрещенных структур. Блок-схемный способ алгоритмизации числовых алгоритмов. Блок-схемный способ алгоритмизации логических алгоритмов. Словесный способ алгоритмизации.	Лабораторные занятия	4	2	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.3	Правила и способы записи алгоритмов. Исследование запрещенных структур. Блок-схемный способ алгоритмизации числовых алгоритмов. Блок-схемный способ алгоритмизации логических алгоритмов. Словесный способ алгоритмизации. Построение операторных схем Ляпунова для числовых алгоритмов. Построение операторных схем Ляпунова для логических алгоритмов. Реализация смешанных алгоритмических процедур на примере игр. Подготовка к тестированию, лабораторным работам.	Самостоятельная работа	4	30	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1 ОПК-1.2 ОПК-1.3

Раздел 2. Алгоритмические системы

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
2.1	Структура и функции машины Тьюринга. Реализация алгоритмов на машине Тьюринга. Нормальные алгоритмы Маркова. Реализация числовых и логических функций в нормальных алгоритмах Маркова.	Лекционные занятия	4	2	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.2	Исследование структуры и функций машины Тьюринга. Реализация числовых алгоритмов на машине Тьюринга.	Лабораторные занятия	4	2	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.3	Частично- и общерекурсивные функции. Композиция частично-рекурсивных функций. Исследование примитивно-, частично- и общерекурсивных функций. Реализация логических алгоритмов на машине Тьюринга. Композиция нормальных алгоритмов Маркова.	Самостоятельная работа	4	34	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1

	Композиция машин Тьюринга. Составление нормальных алгоритмов Маркова для логических задач. Составление нормальных алгоритмов Маркова для числовых задач.				ОПК-1.2 ОПК-1.3
Раздел 3. Оценка сложности алгоритмов					
№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
3.1	Оценка сложности алгоритмов. Сравнение сложности логических и числовых алгоритмов.	Лабораторные занятия	5	2	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
3.2	Теория сложности. Оценка временной и емкостной сложности алгоритмов. Предел Бреммермана. Классификация алгоритмов по сложности реализации. Алгоритмы экспоненциальной и полиномиальной сложности. Основы теории NP-полноты. Неразрешимые проблемы.	Самостоятельная работа	5	30	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
3.3	Подготовка к промежуточной аттестации	Зачет	5	4	ОПК-1 ПКО-1 ПКО-1.1 ПКО-1.2 ПКО-1.3 ОПК-1.1 ОПК-1.2 ОПК-1.3

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Структура и содержание фонда оценочных средств для проведения текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Учебные, научные и методические издания

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
1	Самойленко А. П., Усенко О. А.	Математическая логика и теория алгоритмов в примерах и задачах: учеб. пособие по спец. 032100 "Математика" по курсу "Дискретный анализ" для студентов физико-мат. фак.	Таганрог: Изд-во Таганрог. гос. пед. ин-та, 2004	
2	Игошин	Математическая логика и теория алгоритмов: учеб. пособие для студентов вузов, обучающихся по специальности 032100 "Математика"	М.: Академия, 2004	
3	Самойленко, А. П., Усенко, О. А.	Математическая логика и теория алгоритмов в примерах и задачах: учеб. пособие по спец. 032100 "Математика" по курсу "Дискретный анализ" для студентов физико-мат. фак.	Таганрог: Изд-во Таганрог. гос. пед. ин-та, 2004	28 экз.
4	Игошин, Владимир Иванович	Математическая логика и теория алгоритмов: учеб. пособие для студентов вузов, обучающихся по специальности 032100 "Математика"	М.: Академия, 2004	25 экз.
5	Балюкевич Э. Л., Ковалева Л. Ф.	Математическая логика и теория алгоритмов: учебно-практическое пособие: учебное пособие	Москва: Евразийский открытый институт, 2009	http://biblioclub.ru/index.php?page=book&id=93166
6	Судоплатов С. В., Овчинникова Е. В.	Математическая логика и теория алгоритмов: учебник	Новосибирск: Новосибирский государственный технический университет, 2012	http://biblioclub.ru/index.php?page=book&id=135676

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
7	Брыкалова А. А.	Теория алгоритмов: лабораторный практикум: практикум	Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2016	http://biblioclub.ru/index.php?page=book&id=467401
8	Брыкалова А. А.	Теория алгоритмов: учебное пособие	Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2016	http://biblioclub.ru/index.php?page=book&id=467402
9	Алябьева, В. Г., Пастухова, Г. В.	Теория алгоритмов: учебное пособие для специальности 050201.65 – «математика с дополнительной специальностью «информатика», направление подготовки 050100 – «педагогическое образование»	Пермь: Пермский государственный гуманитарно-педагогический университет, 2013	http://www.iprbookshop.ru/32100.html
10	Безусова, Т. А.	Теория алгоритмов. Основные подходы к формализации алгоритма: учебное пособие	Соликамск: Соликамский государственный педагогический институт, 2011	http://www.iprbookshop.ru/47905.html
11	Опабекова, А. М., Умбетбаев, К. У., Беделов, К. А.	Теория алгоритмов: учебно-методический комплекс	Алматы: Нур-Принт, 2012	http://www.iprbookshop.ru/67154.html
12	Брыкалова, А. А.	Теория алгоритмов: лабораторный практикум	Ставрополь: Северо-Кавказский федеральный университет, 2016	http://www.iprbookshop.ru/69439.html
13	Брыкалова, А. А.	Теория алгоритмов: учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2016	http://www.iprbookshop.ru/69440.html
14	Поднебесова, Г. Б.	Теория алгоритмов: практикум	Челябинск: Южно-Уральский государственный гуманитарно-педагогический университет, 2017	http://www.iprbookshop.ru/83880.html
15	Мирзоев, М. С., Матросов, В. Л.	Теория алгоритмов: учебное пособие	Москва: Прометей, 2019	http://www.iprbookshop.ru/94547.html

5.1. Учебные, научные и методические издания

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
1	Макоха А. Н., Шапошников А. В., Бережной В. В.	Математическая логика и теория алгоритмов: учебное пособие	Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2017	http://biblioclub.ru/index.php?page=book&id=467015
2	Перемилова Т. О.	Математическая логика и теория алгоритмов: учебное пособие	Томск: ТУСУ, 2016	http://biblioclub.ru/index.php?page=book&id=480886
3	Макоха, А. Н., Шапошников, А. В., Бережной, В. В.	Математическая логика и теория алгоритмов: учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2017	http://www.iprbookshop.ru/69397.html
4	Мачикина, Е. П.	Математическая логика и теория алгоритмов: учебно-методическое пособие	Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2020	http://www.iprbookshop.ru/102154.html
5	Гамова, А. Н.	Математическая логика и теория алгоритмов: учебное пособие для студентов механико-математического факультета и факультета компьютерных наук и информационных технологий	Саратов: Издательство Саратовского университета, 2020	http://www.iprbookshop.ru/106266.html

5.2. Профессиональные базы данных и информационные справочные системы

Научная электронная библиотека <https://www.elibrary.ru/defaultx.asp>

Актуальные новости из области компьютерных технологий, информация о программном обеспечении, сетях, безопасности: <https://www.theregister.co.uk>

Справочник по функциям Microsoft Excel: <https://excel2.ru/functions>

5.3. Перечень программного обеспечения

Python
OpenOffice
Notepad++

5.4. Учебно-методические материалы для обучающихся с ограниченными возможностями здоровья

При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения:

- столы, стулья;
- персональный компьютер / ноутбук (переносной);
- проектор;
- экран / интерактивная доска.

Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными и/или свободно распространяемыми программными средствами и выходом в Интернет, и/или в специализированных лабораториях, предусмотренных образовательной программой.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

1.1 Показатели и критерии оценивания компетенций:

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ОПК-1: Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности			
З: основные возможности современного прикладного программного обеспечения для решения вычислительных задач; основы построения правильного логического вывода на основе схем формализации вычислительных процедур на естественном языке, аксиомы и формулы преобразования исчисления высказываний; ключевые понятия и определения теории алгоритмов, формальной теории вычислимости, правила и этапы полного построения алгоритмов, способы их записи и реализации в различных алгоритмических системах;	Осуществление поиска и сбора необходимой литературы, изучение лекционного материала, основной и дополнительной литературы, подготовка к контрольной работе	соответствие проблеме исследования; полнота и содержательность ответа; умение приводить примеры; умение отстаивать свою позицию; умение пользоваться дополнительной литературой при подготовке к занятиям; соответствие представленной информации материалам лекции и учебной литературы, сведениям из информационных ресурсов Интернет	КВ – контрольные вопросы (1-54), Т - тест (1-2)

У: корректно и компактно отображать алгоритмы, используя различные формы записи: словесный, графический, блок-схемный, операторный; получать частично-рекурсивные, примитивно-рекурсивные и общерекурсивные функции; применять принцип логического программирования, элементы алгоритмической логики для решения практических задач; производить временную и емкостную оценку сложности алгоритма, выявлять «узкие» места алгоритмов, а также производить сравнительную характеристику сложности различных алгоритмов; применять различные математические методы при формализации решения вычислительных задач, использовать современного прикладного программного обеспечения для моделирования процессов и явлений, их теоретического и экспериментального исследования;	Изучение современных информационно-коммуникационных технологий, прохождение тестов, выполнение контрольных заданий	достоверность решения заданий с помощью программных средств, правильность выполнения тестовых и контрольных заданий	ЛЗ - лабораторные задания (1.1-1.8, 2.1-2.9, 3.1), КЗ - контрольные задания, Т - тест (1-2)
В: навыками разработки, внедрения и использования прикладного программного	Использование современных информационных технологий	достоверность решения заданий с помощью программных средств, правильность выполнения лабораторных и	ЛЗ - лабораторные задания (1.1-1.8, 2.1-2.9, 3.1), КЗ-

обеспечения для решения вычислительных задач; выявления алгоритмически разрешимых и неразрешимых задач и проблем; формализации вычислительных задач, применения принципа логического программирования и составления алгоритмов решения различных задач с оценкой их временной и емкостной сложности;		контрольных заданий	контрольные задания
ПКО-1: Способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение			
З: основные методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности; способы и правила записи алгоритмов; критерии выявления алгоритмически неразрешимых, легко и трудно разрешимых проблем, оценки мер сложности алгоритмов;	Осуществление поиска и сбора необходимой литературы, изучение лекционного материала, основной и дополнительной литературы, подготовка к контрольной работе	соответствие проблеме исследования; полнота и содержательность ответа; умение приводить примеры; умение отстаивать свою позицию; умение пользоваться дополнительной литературой при подготовке к занятиям; соответствие представленной информации материалам лекции и учебной литературы, сведениям из информационных ресурсов Интернет	КВ – контрольные вопросы (1-54), Т - тест (1-2)
У: производить формализацию и постановку задачи построения алгоритма с заданными свойствами; корректно и компактно отображать алгоритмы, используя различные формы записи: словесный, графический, блок-	Изучение современных информационно-коммуникационных технологий, прохождение тестов, выполнение контрольных заданий	достоверность решения заданий с помощью программных средств, правильность выполнения тестовых и контрольных заданий	ЛЗ - лабораторные задания (1.1-1.8, 2.1-2.9, 3.1), КЗ - контрольные задания, Т - тест (1-2)

схемный, операторный; применять принцип логического программирования, элементы алгоритмической логики для решения практических задач; применять различные математические методы при формализации решения вычислительных задач, использовать современного прикладного программного обеспечения для моделирования процессов и явлений, их теоретического и экспериментального исследования;			
В: навыками реализации различных видов алгоритмов в абстрактных алгоритмических системах и машинах.	Использование современных информационных технологий	достоверность решения заданий с помощью программных средств, правильность выполнения лабораторных и контрольных заданий	ЛЗ - лабораторные задания (1.1-1.8, 2.1-2.9, 3.1), КЗ-контрольные задания

Итоговый результат формируется из суммы набранных баллов за выполнение зачетного задания (1 теоретический вопрос и 1 практико-ориентированное задание) и соответствует шкале:

- 50-100 баллов (зачтено);
- 0-49 баллов (не зачтено).

2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Вопросы к зачету

1. Области применения теории алгоритмов. Приложения теории алгоритмов в информатике.
2. Требования к составу микроопераций и логических условий алгоритма.
3. Дайте определение алгоритма, его особенности.
4. Основные свойства понятия алгоритма в формулировке Колмогорова.
5. Сформулируйте основные определения абстрактного алфавита (алфавит, слово алфавита, расширение алфавита, алфавитный оператор).

6. Понятие кодирования. Приведите пример кодирующего отображения. Каковы основные требования для обеспечения обратимости кодирования?
7. Перечислите и поясните основные этапы полного построения алгоритмов.
8. Какие известны способы записи алгоритмов, дайте их краткую характеристику. Перечислите основные требования к записи алгоритмов.
9. Графический (блок-схемный) способ записи алгоритмов. Его преимущества и недостатки. Перечислите основные обозначения и приведите пример.
10. Операторные алгоритмы Ван Хао. Теорема Минского.
11. Операторные алгоритмы А.А. Ляпунова. Его преимущества и недостатки, приведите пример.
12. Словесная форма записи алгоритмов. Его преимущества и недостатки, приведите пример.
13. Правила составления алгоритмов. В чем суть принципов поэтапной детализации, ограниченного числа структур, модульного принципа.
14. Приведите примеры алгоритмов с использованием типовых структур алгоритмов.
15. Что такое алгоритмизация? Дайте определение и приведите примеры логических и численных алгоритмов.
16. Свойства и виды алгоритмов и соотношение между ними.
17. Понятия колмогоровских комплексов и ансамблей.
18. Понятие исчисления. Эквивалентные исчисления. В чем отличие исчисления от алгоритма?
19. Разрешимые и перечислимые множества.
20. Проблема построения неразрешимого породимого множества. Приведите пример.
21. Проблема сводимости Поста.
22. Понятие вычислимой функции. График вычислимой функции. Приведите пример невычислимой функции.
23. Эффективные операции над вычислимыми функциями. Теорема о неподвижной точке.
24. Частично-рекурсивные и общерекурсивные функции.
25. Сформулируйте и поясните суть тезиса Черча, его особенности.
26. Операции, позволяющие получать сложные арифметические функции из элементарных частично-рекурсивных функций. Приведите примеры.
27. Конечные и бесконечные машины. Регистровые машины.
28. Структура и функции машины Тьюринга.
29. Тезис Тьюринга. Реализация алгоритмов на машинах Тьюринга: функциональная схема и диаграмма переходов.
30. Машина Тьюринга с двумя выходами, как решение проблемы разрешимости множеств.
31. Многоленточная машина Тьюринга. Структура, время и емкость работы.
32. Универсальная машина Тьюринга. Требования к кодированию программы.
33. Композиция машин Тьюринга. Основные операции. Правила составления программ для композиции машин.
34. Композиция машин Тьюринга: произведение, возведение в степень, итерация. Графическое представление композиции машин. Приведите пример.
35. Понятие вычисления на машине Тьюринга. Вычисление арифметических функций на машинах Тьюринга.
36. Нормальные алгоритмы А.А. Маркова, графическое представление. Принцип нормализации. Приведите пример.
37. Композиция нормальных алгоритмов Маркова: суперпозиция, объединение, разветвление и повторение (итерация).
38. Теорема Маркова об универсальном нормальном алгоритме.
39. Теоремы Маркова и Детловса, их значение.

40. В чем суть и значимость теоремы о параметризации? Проблема останова. Эффективные операции над вычислимыми функциями. Теорема Клини о неподвижной точке.

41. Сформулируйте обобщенное определение программы. Универсальные программы и универсальные функции.

42. Поясните понятие нумерованной структуры. Нумерация программного типа.

43. Грамматики. Языки, иерархия языков по Хомскому. Свойства языков и операции над языками. Методы анализа грамматики языков: метод Эйка, Флойда, Наура.

44. Компьютер фон Неймана.

45. В чем состоит общая проблема оценки сложности алгоритмов? Причины и пути анализа сложности алгоритмов.

46. Аксиоматический подход Блюма к оценке сложности алгоритмов.

47. Понятия единичной и массовой проблем. Требования к заданию алгоритмической массовой проблемы. Приведите пример.

48. Понятие и свойства меры сложности алгоритмов. Верхняя и нижняя оценки сложности алгоритмов.

49. Предел сложности Бреммермана.

50. Классификация задач в зависимости от сложности алгоритмов их решения.

51. Классификация сложности алгоритмов по типу сложности решаемых задач: P, NP, NP-полные, coNP.

52. Неразрешимые задачи и проблемы теории алгоритмов.

53. Основы теории NP-полноты: алгоритмы полиномиальной и экспоненциальной сложности, приведите примеры.

54. Сравнительная характеристика алгоритмов полиномиальной и экспоненциальной сложности.

Критерии оценки:

Оценка	Критерии
Отлично (84–100)	ответы на вопросы четкие, обоснованные и полные, проявлена готовность к дискуссии, студент демонстрирует высокий уровень владения знаниями, умениями и навыками соответствующих компетенций, что позволяет ему решать широкий круг типовых и нетиповых задач.
Хорошо (67–83)	ответы на вопросы преимущественно правильные, но недостаточно четкие, студент способен самостоятельно воспроизводить и применять соответствующие знания, умения и навыки для решения типовых задач дисциплины, может выполнять поиск и использование новой информации для выполнения новых профессиональных действий на основе полностью освоенных знаний, умений и навыков соответствующих компетенций
Удовлетворительно (50-66)	ответы на вопросы не полные, на некоторые ответ не получен, знания, умения, навыки сформированы на базовом уровне, студенты частично, с помощью извне (например, с использованием наводящих вопросов, ассоциативного ряда понятий и т.д.) могут воспроизводить и применять соответствующие знания, умения, навыки
Неудовлетворительно (0-49)	на большую часть вопросов ответы не были получены, либо они показали полную некомпетентность студента в материале дисциплины, студент не способен самостоятельно, без помощи извне, воспроизводить и применять соответствующие знания, умения, навыки или знания, умения и навыки у студента не выявлены

Примеры контрольных работ

Раздел «Основы теории алгоритмов»

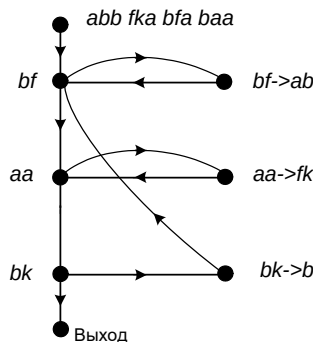
Вариант 0

1. Разработать алгоритм и написать программу для нахождения корней квадратного уравнения.
2. Разработать алгоритм и написать программу для вычисления чисел Фибоначчи.
3. Разработать алгоритм и написать программу для определения значения истинности логической функции $((Y \oplus Z) \rightarrow X) \& (Y \leftrightarrow \bar{Z}) \downarrow (\bar{X} \& \bar{Y}) \mid (Y \& X)$

Раздел «Алгоритмические системы»

Вариант 0

1. Доказать, что функция $f(x) = x + n$ является примитивно рекурсивной.
2. Доказать, что функция $f(x) = \begin{cases} x-1, & \text{если } x > 0, \\ 0, & \text{если } x = 0 \end{cases}$ является общерекурсивной.
3. Вычислить значение функции $f(x) = x - y$ с помощью μ -оператора минимизации.
4. Определите, какое слово будет получено на выходе нормального алгоритма Маркова, заданного следующим графом?



5. Пусть заданы нормальные алгоритмы $A = \{kf \rightarrow mf, fm \rightarrow mf\}$, $B = \{mf \rightarrow k\}$, $C = \{fm \rightarrow k, mf \rightarrow k, k \rightarrow e\}$, $D = \{kkf \rightarrow f\}$ и входное слово $p = kkfmfkffm$ в алфавите $X = \{k, m, f\}$. Составьте композицию алгоритмов: 1) суперпозицию A и B ; 2) объединение A и B ; 3) разветвление A , B и C ; 4) итерацию A и D .
6. Разработать алгоритм Маркова для перемножения двух чисел, записанных в последовательном коде.
7. Определите, какую функцию реализует машина Тьюринга, программа которой представлена функциональной схемой:

	a_0	*	
q_1		$a_0 \text{ Н } q_0$	$a_0 \text{ П } q_2$
q_2	Н q_3	* П q_2	П q_2
q_3	$a_0 \text{ П } q_1$	* Л q_3	Л q_3

8. Построить машину Тьюринга, реализующую функцию $\varphi(n) = n + 10$ в десятичном алфавите.
9. Построить машину Тьюринга (функциональную схему и граф-схему), реализующую функцию $\varphi(n) = 2n - 3$ в алфавите $A = \{a_0, |\}$.

Раздел «Оценка сложности алгоритмов»

Вариант 0

1. Оцените сложность алгоритма по заданной программе:

```
begin
...
s:= Pi/10 + 5;
for i:=1 to m do
```

```

for j:=1 to 2m do
  for k:=i to 3m do s:= k – s + a[i]*b[j];
...
end.

```

2. Докажите или опровергните утверждение, что полином $3n^5 + 10n^3 + n^2 + 25$ имеет сложность $O(n^5)$.

3. Докажите или опровергните утверждение, что процедура вычисления $n!$ имеет сложность $O(n^n)$.

Критерии оценивания. Максимальное количество баллов – 30:

- 21-30 баллов, если студент перечисляет все существенные характеристики обозначенного в вопросе предмета и возможные варианты дальнейшего развития решения проблемы, если это возможно;
- 12-20 баллов, если студент раскрыл только часть основных положений вопроса, продемонстрировал неточность в представлениях о предмете вопроса;
- 6-11 баллов, если студент обозначил общую траекторию ответа, но не смог конкретизировать основные компоненты;
- 0-5 балла, если студент не продемонстрировал знаний основных понятий, представлений об изучаемом предмете.

Лабораторные задания

1. Тематика лабораторных работ по разделам и темам

1. Основы теории алгоритмов 1.1. Правила и способы записи алгоритмов. Исследование запрещенных структур алгоритмов. 1.2. Блок-схемный способ алгоритмизации числовых алгоритмов. 1.3. Блок-схемный способ алгоритмизации логических алгоритмов. 1.4. Словесный способ алгоритмизации. 1.5. Построение операторных схем Ляпунова для числовых алгоритмов. 1.6. Построение операторных схем Ляпунова для логических алгоритмов. 1.7. Реализация смешанных алгоритмических процедур на примере игр.
2. Алгоритмические системы 2.1. Исследование примитивно-, частично- и общерекурсивных функций. 2.2. Реализация заданных функций через частично- и общерекурсивные функции. 2.3. Исследование структуры и функций машины Тьюринга. 2.4. Реализация логических алгоритмов на машине Тьюринга. 2.5. Реализация числовых алгоритмов на машине Тьюринга. 2.6. Композиция машин Тьюринга. 2.7. Составление нормальных алгоритмов Маркова для логических задач. 2.8. Составление нормальных алгоритмов Маркова для числовых задач. 2.9. Композиция нормальных алгоритмов Маркова.
3. Оценка сложности алгоритмов 3.1. Оценка сложности алгоритмов. Сравнение сложности логических и числовых алгоритмов.

2. Критерии оценки:

За выполнение всех лабораторных работ курса запланирован максимум в 40 баллов, если студент в ходе защиты показал наличие твердых знаний по материалу лабораторной работы, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по

применению знаний на практике. В случае частичного выполнения работ, баллы уменьшаются пропорционально количеству защищенных работ.

Тесты письменные и/или компьютерные*

по дисциплине Теория алгоритмов

ВАРИАНТ 1

1. Что не является целью и задачей теории алгоритмов:

- a) формализация понятия «алгоритм» и исследование формальных алгоритмических систем;
- b) формальное доказательство алгоритмической неразрешимости ряда задач;
- c) разработка численных методов решения математических задач;
- d) анализ сложности алгоритмов.

2. Интуитивное понятие алгоритма:

- a) точное предписание, определяющее последовательность действий исполнителя, направленных на решение поставленной задачи;
- b) программа для машины Тьюринга;
- c) программа для машины Поста;
- d) нормальный алгоритм Маркова.

3. Исполнитель алгоритма:

- a) конструктивный объект;
- b) блок-схема;
- c) субъект или устройство, способные правильно выполнить содержащийся в алгоритме перечень действий;
- d) совокупность команд, которые умеет выполнить данный исполнитель.

4. Свойство алгоритма записываться в виде только тех команд, которые находятся в системе команд исполнителя, называется ...

- a) дискретность;
- b) понятность;
- c) определенность;
- d) результативность.

5. Массовость – это свойство алгоритма ...

- a) записываться только директивами однозначно и одинаково интерпретируемыми разными исполнителями;
- b) что при точном исполнении всех предписаний процесс должен прекратиться за конечное число шагов с определенным ответом на поставленную задачу;
- c) записываться в виде упорядоченной совокупности отделенных друг от друга предписаний;
- d) обеспечивающее решение не одной задачи, а целого класса задач этого типа.

6. Данные для алгоритма:

- a) только числа и символы;
- b) только числа и программы;
- c) только числа;
- d) конструктивные объекты: числа, символы, программы.

7. Блок-схема это ...

- a) алгоритмический язык;
- b) таблица;
- c) графическое представление алгоритма;
- d) словесное описание алгоритма.

8. Способ организации вычислительного процесса, при котором процедура или функция в ходе выполнения составляющих ее операторов обращается сама к себе:

- a) итерация;
- b) рекурсия;
- c) цикл;
- d) ветвление.

9. Рекурсия в алгоритме будет косвенной, когда ...

- a) рекурсивный вызов данного алгоритма происходит из вспомогательного алгоритма, к которому в данном алгоритме имеется обращение;
- b) порядок следования команд определяется в зависимости от результатов проверки некоторых условий;
- c) команда обращения алгоритма к самому себе находится в самом алгоритме;
- d) один вызов алгоритма прямо следует за другим/

10. Алгоритм реализован через ... алгоритмическую конструкцию, если каждый шаг алгоритма выполняется один раз, причем после каждого i-го шага выполняется (i+1)-й шаг, если i-й шаг не конец алгоритма.

- a) ветвящуюся;
- b) последовательную;
- c) циклическую;
- d) рекурсивную.

11. Алгоритм реализован с использованием циклической алгоритмической конструкции, если ...

- a) некая, подряд идущая группа шагов алгоритма может выполняться несколько раз в зависимости от входных данных;
- b) каждый шаг алгоритма выполняется один раз, причем после каждого i-го шага выполняется (i+1)-й шаг, если i-й шаг не конец алгоритма;
- c) от входных данных зависит, какие шаги алгоритма будут выполняться;
- d) на каком-либо шаге он прямо или косвенно обращается сам к себе.

12. Трудоемкость алгоритма ...

- a) количество элементарных операций в вычислительном процессе алгоритма;
- b) время, затрачиваемое на выполнение элементарной операции алгоритма;
- c) выполнение алгоритма за конечное разумное время;
- d) эффективность алгоритма.

13. Трудоемкость алгоритма

```

s:=0;
for i:=1 to n do
  begin
    p:=1;
    for j:=1 to n do p:=p*j;
    s:=s+p;
  end;

```

равна:

- a) $1+7n+5n^2$;

- b) $1+5n+15n^3$;
- c) $3+4n$;
- d) $n \cdot \ln(n)$.

14. Итерация (проход) алгоритма – это ...

- a) команда обращения алгоритма к самому себе;
- b) операция выполнения алгоритма к входным данным. При этом исходными данными для каждой последующей итерации являются результаты применения предыдущих итераций;
- c) количество элементарных шагов в вычислительном процессе этого алгоритма;
- d) последовательность шагов алгоритма, пройденных при исполнении этого алгоритма.

15. Глубиной рекурсии называют ...

- a) адрес возврата процедуры;
- b) объем стека;
- c) количество рекурсивных вызовов;
- d) состояние процедуры вызова.

16. Оператор сдвига – это ...

- a) примитивно-рекурсивная функция, определяемая по правилу $\lambda(x) = x + 1$;
- b) арифметический оператор увеличения на единицу $i := i + 1$;
- c) арифметический оператор уменьшения на единицу $i := i - 1$;
- d) элементарная частично-рекурсивная функция, определяемая по правилу $\lambda(x) = x + 1$.

17. Суперпозиция функций заключается:

- a) в подстановке одних арифметических функций вместо аргументов других арифметических функций;
- b) в последовательном выполнении нескольких арифметических функций;
- c) в перемножении нескольких частично-рекурсивных функций;
- d) в последовательном рекурсивном вызове арифметических функций.

18. Функция $f(x,y)$, заданная схемой примитивной рекурсии:

$$\left. \begin{array}{l} f(0, x) = x \\ f(y + 1, x) = f(y, x) + 1 \end{array} \right\}$$

реализует функцию:

- a) $f(x,y) = y * x$;
- b) $f(x,y) = y + x$;
- c) $f(x,y) = y^x$;
- d) $f(x,y) = x^y$.

19. Примитивно рекурсивными функциями называются функции, которые могут быть построены из элементарных арифметических функций с помощью:

- a) операции примитивной рекурсии, примененной любое конечное число раз;
- b) операций суперпозиции, примитивной рекурсии и минимизации, примененных любое конечное число раз в произвольной последовательности;
- c) операций суперпозиции и примитивной рекурсии, примененных любое конечное число раз в произвольной последовательности;
- d) операций примитивной рекурсии и минимизации, примененных любое конечное число раз в произвольной последовательности.

20. Тезис Тьюринга утверждает, что:

- а) для любого алгоритма, понимаемого в интуитивном смысле, можно построить машину Тьюринга, функционирование которой эквивалентно этому алгоритму;
- б) для любого алгоритма можно построить суперпозицию машин Тьюринга, функционирование которых эквивалентно этому алгоритму;
- с) для любой арифметической функции, можно построить машину Тьюринга, вычисляющую эту функцию;
- д) для любой частично-рекурсивной функции, можно построить машину Тьюринга, вычисляющую эту функцию.

21. Совокупность, образованная последовательностью состояний всех ячеек информационной ленты и состоянием устройства управления машины Тьюринга, называется:

- а) итерацией машины Тьюринга;
- б) конфигурацией машины Тьюринга;
- с) тактом работы машины Тьюринга;
- д) состоянием машины Тьюринга.

22. Композитом, или произведением, машины Тьюринга T_1 на машину T_2 называется новая машина T , функционирование которой эквивалентно:

- а) рекурсивному вызову машины T_1 из машины T_2 ;
- б) последовательному выполнению машин T_1 и T_2 ;
- с) определению произведения результатов работы машин T_1 и T_2 ;
- д) циклическому выполнению машин T_1 и T_2 .

23. Алгоритмы, которые задаются графами, составленными исключительно из распознавателей вхождения слов и операторов подстановки, называются:

- а) обобщенными нормальными алгоритмами Маркова;
- б) универсальными машинами Тьюринга;
- с) алгоритмами в виде блок-схем;
- д) машинами Поста.

24. Какую операцию композиции нельзя применять к нормальным алгоритмам Маркова?

- а) повторение;
- б) разветвление;
- с) объединение;
- д) пересечение.

25. Нельзя найти алгоритма, преобразующего программы, который бы по каждой программе давал другую, не эквивалентную ей – это суть ...

- а) принципа нормализации;
- б) тезиса Черча;
- с) Клини о неподвижной точке;
- д) теоремы Тьюринга.

26. Алгоритмическая проблема – это...

- а) проблема создания программы для заданного алгоритма;
- б) проблема построения алгоритма с заданными свойствами;
- с) проблема построения машины Тьюринга для заданного алгоритма;
- д) проблема построения алгоритма для решения данной задачи.

27. Какая задача считается неразрешимой?

- a) задача, для которой не существует алгоритма ее решения на машине Тьюринга;
- b) задача, для которой не может быть написана программа ни на одном из существующих языков программирования;
- c) задача, для которой не существует ее решение с использованием нормального алгоритма Маркова;
- d) задача, для решения которой требуется большой объем памяти и (или) вычислительного времени.

28. Какой из алгоритмов, перечисленных ниже, не является логическим?

- a) алгоритм сортировки;
- b) алгоритм поиска пути в лабиринте;
- c) алгоритм определения корней квадратного уравнения;
- d) алгоритм поиска одинаковых чисел в массиве.

29. К сложностному классу NP относятся задачи:

- a) решаемые алгоритмами полиномиальной сложности;
- b) решение которых может быть проверено алгоритмами за экспоненциальное время;
- c) решаемые алгоритмами логарифмической сложности;
- d) решение которых может быть проверено алгоритмами за полиномиальное время.

30. Алгоритм, временная сложность которого не выражается некоторой полиномиальной функцией размера задачи n , называется ...

- a) экспоненциальным;
- b) полиномиальным;
- c) логарифмическим;
- d) линейным.

ВАРИАНТ 2

1. Точное предписание, которое задает вычислительный процесс, начинающийся с произвольного исходного данного и направленный на получение полностью определяемого этим исходным данным результата – это ...

- a) формула;
- b) интуитивное понятие алгоритма;
- c) точное понятие алгоритма;
- d) блок-схема.

2. Как называется графическое представление алгоритма?

- a) последовательность формул;
- b) блок-схема;
- c) таблица;
- d) словесное описание.

3. Свойство алгоритма записываться в виде упорядоченной совокупности отделенных друг от друга предписаний – ...

- a) понятность;
- b) определенность;
- c) дискретность;
- d) массовость.

4. Субъект или устройство, способные правильно интерпретировать описание алгоритма и выполнить содержащийся в нем перечень действий, – это ...

- a) алгоритмическая структура;
- b) исполнитель алгоритма;
- c) конструктивный объект;
- d) система команд исполнителя.

5. Алгоритмическая модель – это ...

- a) формальное понятие алгоритма;
- b) интуитивное понятие алгоритма;
- c) система команд исполнителя;
- d) формула.

6. Конструктивный объект – это ...

- a) только слова;
- b) только числа, к которым применяется алгоритм;
- c) только пары чисел;
- d) произвольные объекты, к которым может применяться алгоритм.

7. Как называется свойство алгоритма, что при точном исполнении всех предписаний процесс должен прекратиться за конечное число шагов с определенным ответом на поставленную задачу?

- a) понятность;
- b) детерминированность;
- c) дискретность;
- d) результативность.

8. Элементарность – это свойство алгоритма ...

- a) что при точном исполнении всех предписаний процесс должен прекратиться за конечное число шагов с определенным ответом на поставленную задачу;
- b) записываться в виде упорядоченной совокупности отделенных друг от друга предписаний;
- c) записываться только директивами однозначно и одинаково интерпретируемыми разными исполнителями;
- d) обеспечивающее решение не одной задачи, а целого класса задач этого типа.

9. Рекурсией называют ...

- a) сведение задачи к самой себе;
- b) метод последовательных приближений;
- c) разложение задачи в последовательность однородных подзадач;
- d) разложение задачи в последовательность разнородных подзадач.

10. Рекурсия в алгоритме будет прямой, когда ...

- a) рекурсивный вызов данного алгоритма происходит из вспомогательного алгоритма, к которому в данном алгоритме имеется обращение;
- b) порядок следования команд определяется в зависимости от результатов проверки некоторых условий;
- c) команда обращения алгоритма к самому себе находится в самом алгоритме;
- d) один вызов алгоритма прямо следует за другим.

11. Всякая функция, вычисляемая неким алгоритмом является рекурсивной – это формулировка ...

- a) тезиса Тьюринга;
- b) тезиса Черча;
- c) теоремы Клини;

d) теоремы о неподвижной точке.

12. Оператор проектирования – это ...

a) примитивно-рекурсивная функция, определяемая по правилу $\lambda(x) = x + 1$;

b) элементарная частично-рекурсивная функция, определяемая по правилу $O^n(x_1, x_2, \dots, x_n) = 0$;

c) элементарная частично-рекурсивная функция, определяемая по правилу $\lambda(x) = x + 1$;

d) элементарная частично-рекурсивная функция $I_m^n(x_1, x_2, \dots, x_n) = x_m$, $(1 \leq m \leq n)$, повторяющая значения своих аргументов.

13. Функция $f(x, y)$, заданная схемой примитивной рекурсии:

$$\left. \begin{aligned} f(0, x) &= x \\ f(y+1, x) &= f(y, x) + x \end{aligned} \right\}$$

реализует функцию:

a) $f(x, y) = y * x$;

b) $f(x, y) = y + x$;

c) $f(x, y) = y^x$;

d) $f(x, y) = x^y$.

14. Функция, которая является частично рекурсивной и всюду определенной, называется:

a) примитивно-рекурсивной;

b) общезначимой;

c) общерекурсивной;

d) вычислимой.

15. В состав машины Тьюринга не входит:

a) внешний и внутренний алфавиты;

b) вычислительное устройство;

c) управляющая головка;

d) информационная лента.

16. Итерация машины Тьюринга – это...

a) однократное выполнение программы данной машины Тьюринга;

b) отождествление одного из ее выходных состояний с начальным;

c) циклическое многократное выполнение программы данной машины Тьюринга;

d) это операция произведения данной машины Тьюринга n раз.

17. Полиномиальным называется алгоритм ...

a) вычисляющий полином;

b) аппроксимирующий функцию многочленом;

c) временная сложность которого выражается некоторой полиномиальной функцией размера задачи n ;

d) который может проверить предложенное решение за полиномиальное время.

18. К сложностному классу P относятся задачи:

a) полиномиально проверяемые;

b) с полиномиальной сложностью;

c) с логарифмической сложностью;

d) с экспоненциальной сложностью.

19. Алгоритм реализован через ветвящуюся алгоритмическую конструкцию, если ...

- a) некая, подряд идущая группа шагов алгоритма может выполняться несколько раз в зависимости от входных данных;
- b) от входных данных зависит, какие шаги алгоритма будут выполняться;
- c) каждый шаг алгоритма выполняется один раз, причем после каждого i -го шага выполняется $(i+1)$ -й шаг, если i -й шаг не конец алгоритма;
- d) на каком-либо шаге он прямо или косвенно обращается сам к себе.

20. Следствием теоремы о неподвижной точке является утверждение:

- a) всегда можно найти рекурсивный алгоритм для решения данной задачи;
- b) для любой вычислимой функции можно построить машину Тьюринга;
- c) любая частично рекурсивная функция может быть реализована в виде нормального алгоритма Маркова;
- d) существует программа, печатающая на любом входе свой собственный текст.

21. Основными идеями в формулировке алгоритма по Колмогорову являются:

- a) итеративность вычислительного процесса и локальность каждого отдельного шага;
- b) последовательность и детерминированность вычислительного процесса;
- c) возможность бесконечной работы алгоритма и элементарность каждого шага;
- d) необходимость минимального набора операций для реализации алгоритма и использование только элементарных типовых структур.

22. Исчисление включает в себя:

- a) породимое множество, многозначный алфавитный оператор для переработки входных слов и правила извлечения результата;
- b) элементарные частично рекурсивные функции, индуктивное правило получения новых частично рекурсивных и общерекурсивных функций;
- c) порождающие правила, задающие порождающий процесс из объектов рабочей среды, правила выделения основных состояний и правила извлечения результата;
- d) породимые и перечислимые множества, а также порождающие правила, задающие порождающий процесс из объектов рабочей среды.

23. Породимое множество – это множество, которое порождается

- a) каким-либо исчислением;
- b) каким-либо алгоритмом;
- c) какой-либо программой;
- d) какой-либо машиной Тьюринга.

24. Машина Тьюринга, заданная функциональной схемой (на начальной конфигурации управляющая головка находится слева от слова)

	a_0		
q_1	$a_0 \text{ Н } q_0$	$a_0 \text{ П } q_1$	

вычисляет функцию:

- a) $\varphi(n) = n$;
- b) $\varphi(n) = n - 1$;
- c) $\varphi(n) = n + 1$;
- d) $\varphi(n) = 0$.

25. Машина Тьюринга с двумя выходами используется для решения:

- a) проблемы вычислимости произвольной функции;
- b) задачи установления изоморфности двух множеств;
- c) проблемы разрешимости;
- d) задачи установления эквивалентности двух алгоритмов.

26. Какую функцию реализует нормальный алгоритм Маркова, заданный алфавитом $A = \{+, 1\}$ и системой подстановок $'1+1' \rightarrow '11'$, $'1' \rightarrow \bullet'1'$?

- a) функцию суммирования двух чисел;
- b) функцию суммирования трех чисел;
- c) функцию суммирования произвольного конечного множества чисел;
- d) функцию суммирования бесконечного множества чисел.

27. Операция объединения двух нормальных алгоритмов Маркова A и B заключается в:

- a) объединении выходных слов алгоритмов A и B , обработавших одно и то же входное слово;
- b) объединении всех подстановок обоих алгоритмов;
- c) объединении входных слов для последующей совместной его обработки алгоритмами A и B ;
- d) обработке входного слова тем алгоритмом, области определения которого оно принадлежит.

28. Какая из операций не используется для получения нумерации программного типа из способов программирования?

- a) прямое произведение,
- b) кортежное распространение,
- c) рекурсия;
- d) факторизация.

29. Установить нижнюю оценку сложности алгоритма – это значит доказать, что:

- a) никакая программа, реализующая данный алгоритм, не имеет сложности вычисления меньше заданной функции φ ;
- b) никакой алгоритм вычисления на данной модели не имеет сложности вычисления меньше заданной функции φ ;
- c) никакая машина Тьюринга, реализующая данный алгоритм, не имеет сложности вычисления меньше заданной функции φ ;
- d) никакая частично рекурсивная функции, реализующая данный алгоритм, не имеет сложности вычисления меньше заданной функции φ .

30. Предел Бреммермана составляет:

- a) $10^{300\,000}$ бит;
- b) 10^{100} бит в секунду;
- c) 10^{187} бит в секунду;
- d) 10^{93} бит.

Критерии оценивания. Максимальное количество баллов – 30:

- 21-30 баллов, если студент перечисляет все существенные характеристики обозначенного в вопросе предмета и возможные варианты дальнейшего развития решения проблемы, если это возможно;

- 12-20 баллов, если студент раскрыл только часть основных положений вопроса, продемонстрировал неточность в представлениях о предмете вопроса;
- 6-11 баллов, если студент обозначил общую траекторию ответа, но не смог конкретизировать основные компоненты;
- 0-5 балла, если студент не продемонстрировал знаний основных понятий, представлений об изучаемом предмете.

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

Промежуточная аттестация проводится в форме зачета, это аттестация, которая проводится в соответствии с действующим в РГЭУ (РИНХ) Положением о курсовых, экзаменах и зачётах.

Зачет проводится в соответствии с расписанием в компьютерном классе. Количество вопросов контрольном задании – 3. Результаты аттестации заносятся в зачетную ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины адресованы студентам всех форм обучения.

Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные работы.

Важным условием успешного освоения дисциплины «Теория алгоритмов» является создание системы правильной организации труда, позволяющей распределить учебную нагрузку равномерно в соответствии с графиком образовательного процесса. Большую помощь в этом может оказать составление плана работы на семестр, месяц, неделю, день. Его наличие позволит подчинить свободное время целям учебы, трудиться более успешно и эффективно. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подвести итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине они произошли. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана. Все задания к лабораторным занятиям, а также задания, вынесенные на самостоятельную работу, рекомендуется выполнять непосредственно после соответствующей темы лекционного курса, что способствует лучшему усвоению материала, позволяет своевременно выявить и устранить «пробелы» в знаниях, систематизировать ранее пройденный материал, на его основе приступить к овладению новыми знаниями и навыками.

Знакомство с дисциплиной происходит уже на первой лекции, где от студента требуется не просто внимание, но и самостоятельное оформление конспекта. При работе с конспектом лекций необходимо учитывать тот фактор, что одни лекции дают ответы на конкретные вопросы темы, другие – лишь выявляют взаимосвязи между явлениями, помогая студенту понять глубинные процессы развития изучаемого предмета как в истории, так и в настоящее время.

Конспектирование лекций – сложный вид вузовской аудиторной работы, предполагающий интенсивную умственную деятельность студента. Конспект является полезным тогда, когда записано самое существенное и сделано это самим обучающимся. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Целесообразно вначале понять основную мысль, излагаемую лектором, а затем записать ее. Желательно запись осуществлять на одной странице листа или оставляя поля, на которых позднее, при самостоятельной работе с конспектом, можно сделать дополнительные записи, отметить непонятные места.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает лектор, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

В процессе подготовки к лабораторным занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме.

Изучение дисциплины проходит с акцентом на лабораторные работы. Лабораторные занятия проводятся в компьютерных классах с применением специально разработанных учебно-методических материалов, в которых изложены подробные методические рекомендации по изучению каждой темы и выполнению заданий. Наличие таких учебно-методических и дидактических материалов позволяет каждому студенту работать в своем индивидуальном темпе, а также дополнительно прорабатывать изучаемый материал во время самостоятельных занятий.

Перед выполнением лабораторной работы требуется получить вариант задания. Далее необходимо ознакомиться с заданием. Выполнение лабораторной работы следует начать с изучения теоретических сведений, которые приводятся в соответствующих методических указаниях. Результаты работы необходимо оформить в виде отчета.

Лабораторная работа считается выполненной, если

- предоставлен отчет о результатах выполнения задания;
- проведена защита проделанной работы.

Защита проводится в два этапа:

- 1) Демонстрируются результаты выполнения задания.
- 2) В случае лабораторной работы, предусматривающей разработку программного приложения при помощи тестового примера, доказывается, что результат, получаемый при выполнении программы правильный.
- 3) Далее требуется ответить на ряд вопросов из перечня контрольных вопросов, который приводится в задании на лабораторную работу.

Вариант задания выдается преподавателем, проводящим лабораторные занятия.

Для успешного овладения предлагаемым курсом студент должен обладать определённой информационной культурой: навыками работы с литературой, умением определять и находить информационные ресурсы, соответствующие целям и задачам образовательного процесса, получать к ним доступ и использовать в целях повышения эффективности своей профессиональной деятельности. При изучении данного курса необходимо максимально использовать компьютер, изучать дополнительные информационные ресурсы.

Подготовка к промежуточной аттестации.

При подготовке к промежуточной аттестации целесообразно:

- внимательно изучить перечень вопросов и определить, в каких источниках находятся сведения, необходимые для ответа на них;
- внимательно прочитать рекомендованную литературу;
- составить краткие конспекты ответов (планы ответов).