

Документ подписан простой электронной подписью
Информация о владельце: Министерство науки и высшего образования Российской Федерации
ФИО: Смирнов Сергей Николаевич ФГБОУ ВО «Тверской государственный университет»
Должность: врио ректора
Дата подписания: 16.10.2023 21:40:08
Уникальный программный ключ:
69e375c64f7e975d4e8830e7b4fcc2ad1bf35f08

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тверской государственный университет»

Утверждаю:
Руководитель ООП
Н.А. Семькина

« 4 » 09 2023


Рабочая программа дисциплины (с аннотацией)

Принципы алгоритмизации


Специальность

10.05.01 Компьютерная безопасность

Специализация

«Математические методы защиты информации»

Для студентов I курса очной формы обучения

Составитель: 

ст. преподаватель Е.В. Тишина

Тверь 2023

I. Аннотация

1. Наименование дисциплины (модуля) в соответствии с учебным планом Принципы алгоритмизации

2. Цель и задачи дисциплины (модуля)

Дисциплина «Принципы алгоритмизации» имеет целью обучение студентов современным приемам и методам составления алгоритмов и программ, реализующих решение вычислительных задач, а также общих задач обработки информации. В настоящее время существует большое количество разнообразных языков и методологий программирования, с помощью которых можно эффективно решать широкий круг задач. Но залогом успешной разработки программного обеспечения является знание основных принципов алгоритмизации и понимание процесса работы программы. Предлагаемый курс способствует развитию навыков алгоритмического мышления.

Задачей преподавания дисциплины является формирование у студентов профессиональных компетенций, связанных с использованием теоретических знаний в области алгоритмизации, программирования и защиты компьютерных систем, и практических навыков составления алгоритмов и программирования на языке высокого уровня. Основные задачи дисциплины: обучение разработке алгоритмов на основе структурного подхода; закрепление навыков алгоритмизации; подготовка к осознанному использованию как языков программирования, так и методов программирования.

3. Место дисциплины (модуля) в структуре ООП

Дисциплина входит в вариативную часть ООП.

Дисциплина читается на 1 курсе (2 семестр), заканчивается зачетом. Для освоения дисциплины студент должен владеть современными методами и средствами информационных технологий. Необходимы знания, умения и компетенции, полученные студентами на занятиях по предмету информатика в средней общеобразовательной школе и компетенции, сформированные в процессе изучения дисциплины «Информатика».

Дисциплина «Принципы алгоритмизации» является основой для изучения дисциплин: Языки программирования, Объектно-ориентированное программирование, Операционные системы, Системы управления базами данных, Методы программирования, Программное обеспечение компьютерных систем. Знания и практические навыки, полученные из курса «Принципы алгоритмизации», используются студентами при изучении научных дисциплин, а также при разработке курсовых и дипломных работ.

4. Объем дисциплины (или модуля):

3 зачетных единицы, **108** академических часов, в том числе

контактная работа: практические занятия **36** часов, **самостоятельная работа:** **72** часа.

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

<p>Планируемые результаты освоения образовательной программы (формируемые компетенции)</p>	<p>Планируемые результаты обучения по дисциплине (или модулю)</p>
<p>ПК-4. Способность проводить анализ и участвовать в разработке математических моделей безопасности компьютерных систем.</p>	<p>Уметь: строить формальные модели безопасности для защищаемых компьютерных систем.</p> <p>Знать: критерии оценки эффективности и надежности средств защиты компьютерных систем; основные средства и способы обеспечения безопасности компьютерных систем.</p>
<p>ПК-7. Способность проводить анализ проектных решений по обеспечению защищенности компьютерных систем</p>	<p>Уметь: провести анализ компьютерной системы с целью поиска уязвимостей; формулировать рекомендации по параметрам системы обнаружения атак для защищаемой системы.</p> <p>Знать: основные типы компьютерных атак; основные принципы выбора и развертывания системы обнаружения атак.</p>
<p>ПК-19. Способность производить проверки технического состояния и профилактические осмотры технических средств защиты информации.</p>	<p>Уметь: составлять алгоритмы проверки технического состояния средств защиты информации и профилактического осмотра технических средств защиты информации;</p> <p>Знать: классификацию инженерно-технических средств защиты информации; алгоритмы проектирования системы защиты информации.</p>
<p>ПК-20. Способность выполнять работы по восстановлению работоспособности средств защиты информации при</p>	<p>Уметь: составлять алгоритмы работ по восстановлению работоспособности средств защиты информации при возникновении нештатных ситуаций;</p> <p>Знать: средства и меры по защите информации при возникновении нештатных ситуаций.</p>

возникновении нештатных ситуаций.	
---	--

6. Форма промежуточной аттестации зачёт.

7. Язык преподавания русский.

II. Содержание дисциплины (или модуля), 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.)		Самостоятельная работа (час.)
		Лекции	Практические (лабораторные) занятия	
Раздел 1. Введение				
Предмет и задачи курса. История развития языков программирования. Классификация языков программирования и степень абстракции от машинной архитектуры. Поколения (историческая справка). Понятие парадигмы программирования. Основные модели программирования (Императивное программирование, Декларативное программирование, Структурное программирование, Функциональное программирование, Объектно-ориентированное программирование). Некоторые подходы и приёмы программирования: Структурное программирование, Процедурное программирование, Рекурсия, Автоматное программирование.	6		2	4
Раздел 2. Алгоритмические основы программирования				
Определения алгоритма, свойства алгоритмов. Типы алгоритмов. Способы описания алгоритмов: словесный, графический (блок-схемы, ДРАКОН-схемы, диаграммы Насси-Шнейдермана),	18		6	12

<p>псевдокод. Стиль структурного программирования (базовые структуры, дополнительные структуры). Структурная теорема Бома — Джакопини. Построение алгоритма. Исходные данные, промежуточные данные и результаты Алгоритм Дейкстры. Основные характеристики алгоритмов. Временные характеристики. Объемные характеристики. Методы анализа алгоритмов. Оценка сложности алгоритмов.</p>				
Раздел 3. Структурный подход к алгоритмизации и программированию				
<p>Методика разработки алгоритмов. Формализация и моделирование задачи. Основные этапы компьютерного решения задач. Процедурное программирование. Работа с функциями.</p>	10		4	6
Раздел 4. Элементарные структуры данных				
<p>Массивы и Строки. Методы линейного и бинарного поиска. Алгоритмы обработки двумерных массивов. Работа со строками.</p>	14		4	10
Раздел 5. Автоматное программирование				
<p>Автоматное программирование (программирование от состояний). Элементы теории конечных автоматов. Детерминированный конечный автомат. Диаграмма состояний, Таблица переходов. Практика построения ДКА.</p>	18		6	12
<p>Стиль автоматное проектирование программ. Написание программы в автоматном стиле.</p>	12		4	8
Раздел 6. Динамические структуры данных				
<p>Элементарные структуры данных: Стек и очередь. Реализация очереди на базе массива. Реализация стека на базе массива. Использование стека в программировании. Задачи на применение стека. Прямая и обратная польская запись.</p>	18		6	12
Раздел 7. Рекурсивные алгоритмы				
<p>Понятие рекурсии Хвостовая (концевая) рекурсия.</p>	12		4	8

Реализация рекурсивных алгоритмов				
ИТОГО	108		36	72

Учебная программа

1. Раздел 1. Введение

- 1.1. Предмет и задачи курса. История развития языков программирования. Классификация языков программирования и степень абстракции от машинной архитектуры. Поколения (историческая справка). Понятие парадигмы программирования.
- 1.2. Основные модели программирования (Императивное программирование, Декларативное программирование, Структурное программирование, Функциональное программирование, Объектно-ориентированное программирование).
- 1.3. Некоторые подходы и приёмы программирования: Структурное программирование, Процедурное программирование, Рекурсия, Автоматное программирование.

2. Раздел 2. Алгоритмические основы программирования

- 2.1. Определения алгоритма, свойства алгоритмов. Типы алгоритмов.
- 2.2. Способы описания алгоритмов: словесный, графический (блок-схемы, ДРАКОН-схемы, диаграммы Насси-Шнейдермана), псевдокод.
- 2.3. Стиль структурного программирования (базовые структуры, дополнительные структуры).
- 2.4. Структурная теорема Бома — Джакопини.
- 2.5. Построение алгоритма. Исходные данные, промежуточные данные и результаты Алгоритм Дейкстры.
- 2.6. Основные характеристики алгоритмов. Временные характеристики. Объемные характеристики. Методы анализа алгоритмов. Оценка сложности алгоритмов.

3. Раздел 3. Структурный подход к алгоритмизации и программированию

- 3.1. Методика разработки алгоритмов. Формализация и моделирование задачи.
- 3.2. Основные этапы компьютерного решения задач.
- 3.3. Процедурное программирование.
- 3.4. Работа с функциями.

4. Раздел 4. Элементарные структуры данных

- 4.1. Массивы и Строки.
- 4.2. Методы линейного и бинарного поиска.
- 4.3. Алгоритмы обработки двумерных массивов.
- 4.4. Работа со строками.

5. Раздел 5. Автоматное программирование

- 5.1. Автоматное программирование (программирование от состояний).
- 5.2. Элементы теории конечных автоматов. Детерминированный конечный автомат.
- 5.3. Диаграмма состояний, Таблица переходов.
- 5.4. Практика построения ДКА.
- 5.5. Стиль автоматное проектирование программ.
- 5.6. Написание программы в автоматном стиле.

6. Раздел 6. Динамические структуры данных

- 6.1.Элементарные структуры данных: Стек и очередь.
- 6.2.Реализация очереди на базе массива.
- 6.3.Реализация стека на базе массива.
- 6.4.Использование стека в программировании.
- 6.5.Задачи на применение стека.
- 6.6.Прямая и обратная польская запись.
- 7. **Раздел 7. Рекурсивные алгоритмы**
 - 7.1.Понятие рекурсии
 - 7.2.Хвостовая (концевая) рекурсия.
 - 7.3.Реализация рекурсивных алгоритмов

III. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (или модулю)

Планы практических (семинарских) занятий и методические рекомендации к ним.

Методические рекомендации по организации самостоятельной работы студентов.

Тематика рефератов и рекомендации по их оформлению

Тестовые вопросы для самоконтроля

IV. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (или модулю)

1 Типовые контрольные задания для проверки уровня сформированности компетенций

•

Этап формирования компетенции, в котором участвует дисциплина	Типовые контрольные задания для оценки знаний, умений, навыков (2-3 примера)	Показатели и критерии оценивания компетенции, шкала оценивания
владеть	Оценка популярности языков программирования	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл • Допущены фактические и логические ошибки, свидетельствующие о непонимании темы – 0 баллов
	Дана дата сегодняшнего дня (день, месяц, год). Нарисовать блок-схему	<ul style="list-style-type: none"> • Имеется полное верное решение, включающее правильный ответ – 3 балла • Дано верное решение, но в

	алгоритма определения даты следующего/предыдущего дня.	решении имеются неверные записи, не отделенные от решения – 2 балла <ul style="list-style-type: none"> • Имеется верное решение части блок-схемы из-за логической ошибки – 1 балл • Решение не дано ИЛИ дано неверное решение – 0 баллов
уметь	Перечислите основные алгоритмические конструкции;	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл • Допущены фактические и логические ошибки, свидетельствующие о непонимании темы – 0 баллов
	По блок-схеме записать структуру программы и алгоритмические цепочки;	<ul style="list-style-type: none"> • Имеется полное верное решение, включающее правильный ответ – 3 балла • Дано верное решение, но в решении имеются неверные записи, не отделенные от решения – 2 балла • Имеется верное решение части блок-схемы из-за логической ошибки – 1 балл • Решение не дано ИЛИ дано неверное решение – 0 баллов
знать	Написание реферата	<ul style="list-style-type: none"> • Отражение в плане ключевых аспектов темы – 2 балла; • Фрагментарное отражение ключевых аспектов темы – 1 балл; • верно оформлены ссылки на используемую литературу – 1 балл • соблюдены требования к объёму реферата – 1 балл.
	Что означает стиль структурного программирования (базовые структуры, дополнительные структуры)	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл • Допущены фактические и логические ошибки, свидетельствующие о

		непонимании темы – 0 баллов
владеть	Перечислите и охарактеризуйте основные модели программирования.	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл • Допущены фактические и логические ошибки, свидетельствующие о непонимании темы – 0 баллов
	Что означает термин парадигмы программирования	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл • Допущены фактические и логические ошибки, свидетельствующие о непонимании темы – 0 баллов
	Задания на построение детерминированного конечного автомата проверки, является ли строка числом;	<ul style="list-style-type: none"> • Имеется полное верное решение, включающее правильный ответ – 3 балла • Дано верное решение, но в решении имеются неверные записи, не отделенные от решения – 2 балла • Имеется верное решение части программы из-за алгоритмической ошибки – 1 балл • Решение не дано ИЛИ • дано неверное решение – 0 баллов

<p>уметь</p>	<p>Демонстрация умения строить логически правильные и эффективные программы; Определить, является ли точка с координатами X, Y точкой пересечения диагоналей квадрата со стороной R, одна вершина которого расположена в начале координат.</p>	<ul style="list-style-type: none"> • Имеется полное верное решение, включающее правильный ответ – 3 балла • Дано верное решение, но в решении имеются неверные записи, не отделенные от решения – 2 балла • Имеется верное решение части программы из-за алгоритмической ошибки – 1 балл • Решение не дано ИЛИ дано неверное решение – 0 баллов
<p>знать</p>	<p>Особенности и преимущества рекурсивного подхода в описании алгоритма.</p>	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл • Допущены фактические и логические ошибки, свидетельствующие о непонимании темы – 0 баллов
	<p>Области применения языка ДРАКОН.</p>	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл Допущены фактические и логические ошибки, свидетельствующие о непонимании темы – 0 баллов

владеть	<p>Написать программу в автоматном стиле распознающую корректные адреса электронной почты в текстовом потоке</p>	<ul style="list-style-type: none"> • Имеется полное верное решение, включающее правильный ответ – 3 балла • Дано верное решение, но в решении имеются неверные записи, не отделенные от решения – 2 балла • Имеется верное решение части автомата из-за алгоритмической ошибки – 1 балл • Решение не дано ИЛИ дано неверное решение – 0 баллов
	<p>Задание. Дан массив. Требуется расположить отрицательные элементы в порядке убывания</p>	<ul style="list-style-type: none"> • Имеется полное верное решение, включающее правильный ответ – 3 балла • Дано верное решение, но в решении имеются неверные записи, не отделенные от решения – 2 балла • Имеется верное решение части программы из-за алгоритмической ошибки – 1 балл • Решение не дано ИЛИ дано неверное решение – 0 баллов
уметь	<p>Нарисовать блок-схему алгоритма. В одномерном массиве определить первый отрицательный элемент и его номер.</p>	<ul style="list-style-type: none"> • Имеется полное верное решение, включающее правильный ответ – 3 балла • Дано верное решение, но в решении имеются неверные записи, не отделенные от решения – 2 балла • Имеется верное решение части блок-схемы из-за логической ошибки – 1 балл • Решение не дано ИЛИ дано неверное решение – 0 баллов
	<p>Демонстрация умения использовать язык программирования Си; Ввести двумерный массив. Составить алгоритм подсчета среднего арифметического значений двумерного массива.</p>	<ul style="list-style-type: none"> • Имеется полное верное решение, включающее правильный ответ – 3 балла • Дано верное решение, но в решении имеются неверные записи, не отделенные от решения – 2 балла • Имеется верное решение части программы из-за алгоритмической ошибки – 1 балл • Решение не дано ИЛИ дано неверное решение – 0 баллов

знать	Опишите методы сортировки одномерного массива.	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл • Допущены фактические и логические ошибки, свидетельствующие о непонимании темы – 0 баллов
	В чем заключается сущность метода двоичного поиска?	<ul style="list-style-type: none"> • Факты и примеры в полном объеме обосновывают выводы – 2 балла • Допущена фактическая ошибка, не приведшая к существенному искажению смысла – 1 балл • Допущены фактические и логические ошибки, свидетельствующие о непонимании темы – 0 баллов

При оценивании результатов освоения дисциплины применяется «рейтинговая» технология (балльно-накопительная) система. Оценка уровня сформированности компетенций осуществляется в процессе следующих форм контроля:

1) **слеящегоо** (проводится оценка выполнения студентами заданий в ходе аудиторных занятий). Дает возможность квалифицировать степень сформированности знаний, умений, навыков, а также их глубину и прочность. Его задача - регулярное управление учебной деятельностью студентов и ее корректировка. Он позволяет получать первичную информацию о ходе и качестве усвоения учебного материала, а также стимулировать регулярную, напряженную и целенаправленную работу студентов. Данный контроль позволяет вовремя выявить пробелы в знаниях и оказать им помощь в усвоении программного материала. Данными формами контроля являются: ответы с места и у доски, проверка работ выполненных в тетради.

- 2) **текущего** (оценивается работа студентов вне аудиторных занятий). Текущими формами контроля являются: проверка выполнения домашних работ, ответы у доски, рефераты, доклады, проверка самостоятельной работы студентов.
- 3) **промежуточного** (рейтинговые точки) позволяет определять качество изучения студентами учебного материала по разделам и темам. Контроль проводится два раза в семестр. С помощью периодического контроля обобщаются и усваиваются целые темы и разделы, выявляются взаимосвязи с другими разделами, предметами. Контроль охватывает студентов и всей группы и проводится в виде теста, письменных практических работ.
- 4) **итогового** (зачёт). Максимальная сумма рейтинговых баллов по дисциплине составляет 100 баллов. Студенту, набравшему 50 баллов и выше по итогам работы в семестре, в экзаменационной ведомости и зачетной книжке выставляется оценка «зачтено». Студент, набравший от 20 до 49 баллов включительно, сдаёт зачет в последнюю неделю семестра по данной дисциплине. Баллы, полученные на зачете проставляются в ведомости. Студенту, набравшему меньше 20 баллов, в экзаменационной ведомости выставляется оценка «незачтено». Данному студенту разрешается передача зачета по направлению деканата на последней неделе семестра.

Вид учебной работы, за которую ставятся баллы	Баллы
оценка выполнения студентами заданий в ходе аудиторных занятий, устный опрос	0-10
реферат	0-2
Практическая работа 1	0-5
Практическая работа 2	0-5
Практическая работа 3	0-5
Практическая работа 4	0-6
Практическая работа 5	0-7
Контрольный тест к модулю 1	0-10
Контрольный тест к модулю 2	0-10
Контрольная работа 1	0-10
Контрольная работа 2	0-10
Итоговый контрольный тест	0-20
Всего	0-100

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

Для оценки уровня теоретических и практических знаний используется тест или контрольный письменный опрос. Перечень некоторых вопросов теста и практических заданий представлен ниже.

Тест 1.

Вариант 1

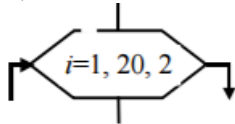
- 1) Свойством алгоритма не является...

- a) Дискретность
- b) Определенность
- c) Массовость
- d) Наглядность

2) Фрагмент алгоритма: «сравним X и Y. если X меньше Y, значит большее число Y.» является...

- a) Графическим
- b) Программным
- c) Псевдокодом
- d) Словесным описанием

3) Элемент блок-схемы: означает...



- a) Условием перехода
- b) Вычислительным блоком
- c) Началом цикла

4) Какой подход в программировании называется структурным?

- a) подход, ориентированный на непосредственно выполняемые компьютером операции
- b) подход, неориентированный на непосредственно выполняемые компьютером операции
- c) подход, ориентированный на то, что отдельные группы операторов могут объединяться во вспомогательные алгоритмы
- d) подход, ориентированный на то, что логическая структура программы может быть выражена комбинацией 3-х базовых структур — следования, ветвления, цикла ☒

5) Основой метода структурного программирования являются ...

- e) принцип модульности разработки сложных программ
- f) использование композиции трех базовых элементов – линейной, ветвления и циклической
- g) использование композиции двух базовых элементов – ветвления и циклической
- h) использование большого количества подпрограмм

6) По Вашему мнению, с точки зрения C++ верен ли приведенный код:

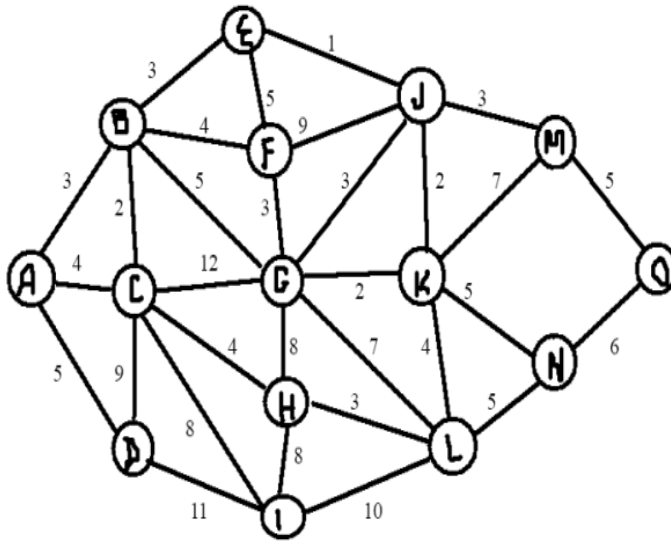
```
double a;
void main (){
int a;
a=5;
::a = 3.14;
}
```

7) Какой из перечисленных операторов является оператором немедленной передачи управления в начало следующей итерации цикла?

- a) if;
- b) switch;
- c) break;
- d) continue.

Задание 1.

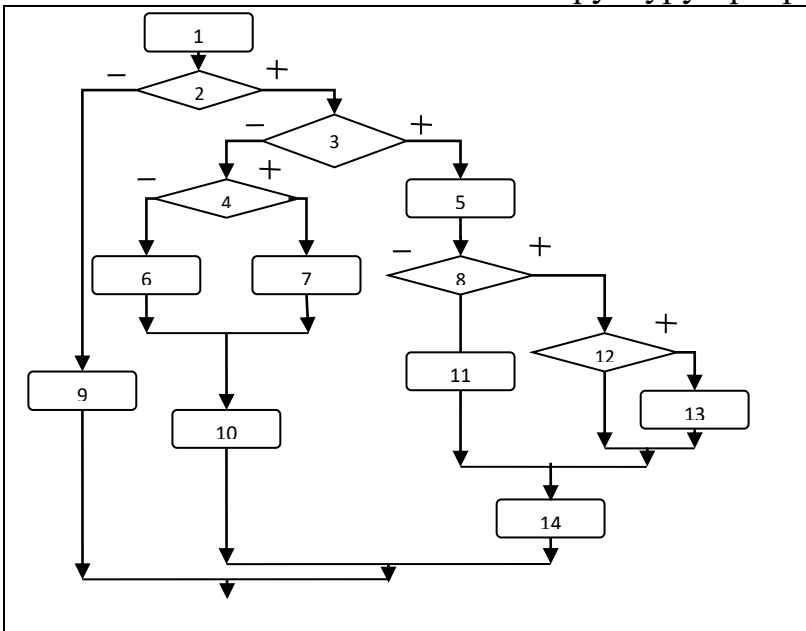
1. С помощью алгоритма Дейкстры найти кратчайшее расстояние между вершиной А и всеми остальными.



Заполнить массивы

dist и prev

2. По блок-схеме записать структуру программы и алгоритмические цепочки.



3. по цепочкам восстановить структуру программы и блок-схему

- 3,4,6,10,11,12,13,14,17
- 3,4,6,10,11,12,14,17
- 3,4,6,10,11,14,17
- 3,4,6,7,8,9,15,16,17
- 3,4,6,7,8,9,15,17
- 3,4,6,7,15,16,17
- 3,4,6,7,15,17
- 3,4,5,17

Задание 2.

1. Записать алгоритм, который переводит время из секунд в часы, минуты и секунды. Например: 4506 секунд – это 1 часа 15 мин 6 секунд. Алгоритм должен проверять правильность введенных пользователем данных и в случае, если данные неверные, выводить соответствующее сообщение.

2. Выяснить из четного или нечётного количества цифр состоит введенное число типа int.

3. Дана дата сегодняшнего дня (день, месяц, год). Нарисовать блок-схему алгоритма определения даты следующего дня.

Задание 3

Работа с датами. Написать программу выполняющую след задачи (каждую задачу оформить в виде функции).

1. Определение високосного года
2. Определение числа дней в году
3. Определение числа дней в месяце (2 варианта)
4. Проверка правильности даты
5. Получение даты следующего дня
6. Подсчёт количества дней от до нужной даты
7. Определение даты по количеству дней
8. Будущая дата (дата + n дней)
9. Прошедшая дата (дата – n дней)
10. Число дней между двумя датами
11. Сортировка массива дат

Задание 4.

1. Построить диаграмму для автомата. Построить язык.

$M = (\{s, p, q\}, \{0, 1\}, \square, s, \{q\})$

\square	0	1
s	p	s
p	q	s
q	q	q

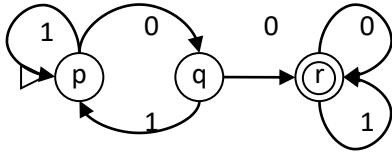
2. Построить диаграмму для автомата.

$M = (\{q_1, q_2, q_3, q_4\}, \{a, b\}, \square, q_1, \{q_3, q_4\})$

	i	
1		2
1		2
2		3
2		4
3		4
3		3
4		4

4		3
---	--	---

3. По данному автомату построить язык, который этим автоматом распознаётся:



4. Построить детерминированный КА, допускающий в алфавите $Z=\{0,1\}$ все цепочки нулей и единиц, содержащие подцепочку 11, например $\omega=01100$. Язык $L_A=\{x11y \mid x \text{ и } y - \text{некоторые цепочки, состоящие только из } 0 \text{ и } 1\}$. Тогда $\omega=0101001 \notin L_A$.

5. Построить конечный автомат, распознающий язык L , состоящий из таких слов w алфавита $\Sigma = \{a, b\}$, которые начинаются на aa и содержат нечётное число символов b . (Проверить работу автомата на входном слове $w = aababa$.)

6. Построить КА с входным алфавитом $V=\{a, b, c, d\}$, распознающий все цепочки, начинающиеся и заканчивающиеся различными символами;

7. Построить КА с входным алфавитом $V=\{a, b, c\}$, распознающий все цепочки, в которых две последние буквы не совпадают;

8. Построить конечный автомат, распознающий язык L , состоящий из слов алфавита $\Sigma = \{a, b\}$, в которых буква a входит чётное число раз, а число вхождений буквы b делится на 3.

9. Каждое слово языка содержит подслово $*abc*$.

10. Построить ДКА, допускающий в алфавите $Z\{0,1\}$ все цепочки нулей и единиц, содержащие в себе множество цепочек, которые начинаются или оканчиваются (или и то и другое) последовательностью 01;

11. Построить ДКА, допускающий в алфавите $Z\{0,1\}$ все цепочки нулей и единиц, содержащие в себе множество всех цепочек, в которых всякая подцепочка из пяти последовательных символов содержит хотя бы два 0.

12. Каждое слово языка содержит подслово $*a?b*$. Алфавит $L = \{a, b, c\}$.

Задание 5.

1. отсортировать список слов
2. сколько слов начинаются на букву А
3. определить среднюю длину слова
4. сколько слов начинаются на согласную
5. подсчитать сколько раз встречается каждое слово

Список вопросов к зачету

1. Каковы особенности императивного программирования, структурного программирования, функционального программирования, объектно-ориентированного программирования?

2. Опишите приём программирования: Структурное программирование. Структурная теорема Боба — Джакопини.

3. Опишите приём программирования: Процедурное программирование,

4. Опишите приём программирования: Рекурсия

5. Опишите приём программирования: Автоматное программирование.
6. Определения алгоритма. Свойства алгоритмов.
7. Способы описания алгоритмов.
8. Основные характеристики алгоритмов. Временные характеристики.

Объемные характеристики.

9. Как оценить сложность алгоритма?
10. Основные этапы компьютерного решения задач.
11. Величины в алгоритмах, их тип, имя, значение.
12. Что такое рекурсивный алгоритм?
13. Что такое сортировка? Какие алгоритмы сортировки Вы знаете?
14. Что такое детерминированный конечный автомат. Диаграмма состояний,

Таблица переходов.

15. Что такое поиск? Какие алгоритмы поиска Вы знаете?
16. Базовые конструкции структурного программирования C++. Линейные алгоритмы. Условные операторы в C++. Примеры. (Управляющие структуры: инструкции и блоки, конструкция if-else, if-else-if, switch.)
17. Базовые конструкции структурного программирования C++. Операторы цикла в C++. Примеры. (циклы while и for, цикл do-while и for). Вложенные циклы.
18. Управляющие структуры: инструкции break, continue, goto.
19. Простейшие структуры данных: массивы
20. Описание массива. Ввод и вывод массива.
21. Обработка элементов массива.
22. Описание строкового типа.
23. Стандартные процедуры и функции обработка текстовой информации.
24. Выделение слов из текста
25. Стеки, очереди и реализация базовых операций над ними.
26. Использование стека в программировании. Задачи на применение стека.
27. Прямая и обратная польская запись.
28. Понятие рекурсии
29. Что такое хвостовая (концевая) рекурсия.

V. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (или модуля)

а) Основная литература:

1. Окулов, С.М. Программирование в алгоритмах [Электронный ресурс] — Электрон. дан. — Москва : Издательство "Лаборатория знаний", 2017. — 386 с. — Режим доступа: <https://e.lanbook.com/book/94140>
2. Кирнос В.Н. Информатика 2. Основы алгоритмизации и программирования на языке C++ [Электронный ресурс]: учебно-методическое пособие/ Кирнос В.Н.— Электрон. текстовые данные.— Томск: Томский государственный университет систем управления и радиоэлектроники, Эль Контент, 2013.— 160 с.— Режим доступа: <http://www.iprbookshop.ru/14011.html>
3. Самуйлов С.В. Алгоритмы и структуры обработки данных [Электронный ресурс]: учебное пособие/ Самуйлов С.В.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2016.— 132 с.— Режим доступа: <http://www.iprbookshop.ru/47275.html>

б) Дополнительная литература:

1. Задачник-практикум по основам программирования [Электронный ресурс] : учебное пособие по курсу «Информатика» / Н.И. Амелина [и др.]. — Электрон. текстовые данные. — Ростов-на-Дону: Южный федеральный университет, 2009. — 192 с. — 978-5-9275-0704-7. — Режим доступа: <http://www.iprbookshop.ru/46954.html>
2. Никлаус Вирт Алгоритмы и структуры данных [Электронный ресурс]/ Никлаус Вирт— Электрон. текстовые данные.— Саратов: Профобразование, 2017.— 272 с.— Режим доступа: <http://www.iprbookshop.ru/63821.html>
3. Синюк В.Г. Алгоритмы и структуры данных [Электронный ресурс]: лабораторный практикум. Учебное пособие/ Синюк В.Г., Рязанов Ю.Д.— Электрон. текстовые данные.— Белгород: Белгородский государственный технологический университет им. В.Г. Шухова, ЭБС АСВ, 2013.— 204 с.— Режим доступа: <http://www.iprbookshop.ru/28363.html>

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

1. ЭБС Лань <https://e.lanbook.com/> Договор № 4-е/23 от 02.08.2023 г.
2. ЭБС Znanium.com <https://znanium.com/> Договор № 1106 эбс от 02.08.2023 г.
3. ЭБС Университетская библиотека online <https://biblioclub.ru> Договор № 02-06/2023 от 02.08.2023 г.
4. ЭБС ЮРАЙТ <https://urait.ru/> Договор № 5-е/23 от 02.08.2023 г.
5. ЭБС IPR SMART <https://www.iprbookshop.ru/> Договор № 3-е/23К от 02.08.2023 г.

VII. Методические указания для обучающихся по освоению дисциплины (или модуля)

Материал дисциплины распределен по главным разделам (темам). В результате изучения дисциплины у студентов должно сформироваться научное представление о принципах разработки базовых алгоритмов для решения различных прикладных задач. В процессе обучения студенты, наряду с текстами лекций и учебными пособиями, должны пользоваться дополнительными научными изданиями, академическими периодическими изданиями. После каждой практической темы рекомендуется проработать вопросы для повторения и самоконтроля. В аспекте самостоятельной работы рекомендуется составлять конспект с наиболее важными методами и приемами создания приложений. Рекомендуется использовать справочники и руководства.

Для успешного освоения дисциплины важно соблюсти следующие рекомендации: На первом занятии важно обратить внимание на конкретные требования к прохождению и сдаче курса. Активная работа на занятиях, выполнение творческих заданий сформирует о Вас дополнительное положительное представление как об активном участнике познавательного процесса. На данном курсе практические

занятия являются самым важным компонентом обучающего процесса. На занятиях будет представлен необходимый теоретически материал по темам и представлены практические задания для решения на занятиях в аудитории под руководством преподавателя и самостоятельно. Многие задачи являются стандартными и имеют уже готовые шаблоны (алгоритмы) решения, тем не менее, для получения большего познавательного и учебного эффекта, настоятельно рекомендуется написание собственного оригинального кода.

Самостоятельная работа студентов в рамках данной дисциплины в основном состоит в подготовке к практическим занятиям и написании алгоритмов и программ, в работе с разными источниками. Освоению учебного материала большую помощь окажет личный творческий подход, связанный с дополнительным просмотром материала по отдельным темам в библиотеках и системе «Интернет». Самостоятельная работа является необходимой на всей стадиях и при всех формах изучения предмета. Важно помнить: без самостоятельной работы невозможно серьезное освоение любого курса. Надо быть готовым к тому, что по времени, затраченном на дисциплину, самостоятельная работа будет превалировать над иными видами работы. Важно продумать стиль фиксации нового и важного материала. Рекомендуется немедленно обсуждать любые возникшие в процессе обучения вопросы, проблемы и неясности с преподавателем, не откладывая это обсуждение до контрольной точки. Проконсультироваться с преподавателем можно во время и после практических занятий, во время консультаций, а также по электронной почте.

Самостоятельная работа обучающихся направлена на освоение учебного материала и развитие практических умений. Самостоятельная работа включает следующие виды самостоятельной работы студентов: работа с рекомендованной литературой и документацией; выполнение практических заданий; подготовка к контрольным тестам и итоговому зачёту.

Список практических заданий

- 1) По блок-схеме записать структуру программы и алгоритмические цепочки. Дана дата сегодняшнего дня (день, месяц, год). Нарисовать блок-схему алгоритма определения даты следующего/предыдущего дня.
- 2) С помощью алгоритма Эдгера Дейкстры найти кратчайшее расстояние между вершинами по заданному графу. Заполнить массивы `dist` и `prev`.
- 3) Работа с датами. Определение високосного года, Определение числа дней в году, Определение числа дней в месяце (2 варианта), Проверка правильности даты, Получение даты следующего дня, Подсчёт количества дней от ... до нужной даты, Определение даты по количеству дней, Будущая дата (дата + n дней), Прошедшая дата (дата – n дней), Число дней между двумя датами, Сортировка массива дат.
- 4) Задания на построение детерминированного конечного автомата.
- 5) Написать программу в автоматном стиле: автомат для проверки, является ли строка числом; автомат, распознающий корректные адреса электронной почты.
- 6) Использование стека в программировании. Задачи на применение стека. Прямая и обратная польская запись.
- 7) Реализация рекурсивных алгоритмов

Обязательным условием подготовки студентов к практическим занятиям является повторение ранее изученного материала по дисциплине, чтение рекомендованной дополнительной литературы. Текущий контроль усвоения знаний осуществляется путем подготовки и сдачи отчетов по итогам выполнения практических работ.

Тематика рефератов и рекомендации по их оформлению;

1. Объем от 16 стр.
2. Шрифт Times New Roman.
3. Размер 14.
4. Интервал 1,5.
5. Абзацный отступ стандартный (без отступов до и после абзаца).
6. Содержание реферата: Титульный лист, оглавление (автоматическое), Введение, главы 1,2, ..., заключение, список литературы.
7. Реферат сдается в двух вариантах: напечатанном и в электронном виде.

Темы рефератов и выступлений:

- 1) Перспективные направления развития информационных технологий в области образования.
- 2) Государственная политика России в области информационных ресурсов: состояние и проблемы.
- 3) Какие существуют премии в области информатики. Лауреаты премий в области информатика. (Рассказать об одном, двух ученых.)
- 4) Информатика как научная дисциплина. Место информатики в научном мировоззрении.
- 5) Визуальный алгоритмический язык программирования и моделирования ДРАКОН (Дружелюбный русский алгоритмический язык, который обеспечивает наглядность).
- 6) Языки программирования высокого уровня.
- 7) Языки программирования: классификация и история развития.
- 8) Современные технологии программирования.
- 9) Жизненный цикл программных средств.
- 10) Оценка популярности языков программирования. Мировые и российские рейтинги.
- 11) Этические нормы поведения в информационной сети.
- 12) Современное состояние и перспективы дальнейшего развития языков программирования высокого уровня
- 13) Патентование ПО. Интеллектуальная собственность.

Контрольные вопросы для самопроверки

- 1) Что называют алгоритмом?
- 2) Определите понятие «исполнитель алгоритмов».
- 3) Перечислите свойства алгоритмов.
- 4) Каким образом используются алгоритмы при решении задач?
- 5) Для чего применяют алгоритмы?
- 6) Укажите способы задания алгоритмов?

- 7) Какие существуют средства графического представления алгоритмов?
- 8) Какие существуют базовые алгоритмические структуры? Приведите пример.
- 9) Приведите примеры линейных и разветвленных алгоритмов.
- 10) Какие алгоритмы относятся к циклическим?
- 11) Что называют телом цикла?
- 12) Какие существуют виды циклов?
- 13) Чем отличается цикл с предусловием от цикла с постусловием?
- 14) Понятие о структурном программировании.
- 15) Дайте определение массива.
- 16) Чем массив отличается от последовательности значений?
- 17) Что определяет номер элемента массива?
- 18) Сформулируйте характеристики массива.
- 19) Какие типы алгоритмических структур применяются для обработки массива?
- 20) Какие существуют способы обработки одномерного массива?
- 21) В чем заключается сортировка одномерного массива?
- 22) Опишите методы сортировки одномерного массива.
- 23) Приведите пример и алгоритм сортировки массива.
- 24) Опишите методы обработки упорядоченных массивов.
- 25) В чем заключается сущность метода двоичного поиска?
- 26) Как осуществляется доступ к элементу двумерного массива?
- 27) Какие существуют способы обработки двумерного массива?
- 28) Рекурсия. Механизм рекурсии. Примеры.
- 29) Стек. Реализация. Примеры использования.
- 30) Очередь. Реализация. Примеры использования.
- 31) Решение задач разветвляющей структуры и программная реализация.
- 32) Решение задач разветвляющей структуры: построение блок-схемы и ее программная реализация.
- 33) Решение задач разветвляющей структуры с использованием оператора выбора: построение блок-схемы и ее программная реализация.
- 34) Решение задач циклической структуры со счетчиком: построение блок-схемы и ее программная реализация.
- 35) Решение задач циклической структуры с предусловием: построение блок-схемы и ее программная реализация.
- 36) Решение задач циклической структуры с постусловием: построение блок-схемы и ее программная реализация.
- 37) Решение задач с обработкой одномерных массивов.
- 38) Решение задач с обработкой двумерных массивов.
- 39) Обработка строк, как одномерных массивов.
- 40) Разработка программ с использованием функций.
- 41) Программирование рекурсивных алгоритмов

Макет оформления задания для кейс-задачи

Кейс-задача по дисциплине.

Задание:

Необходимо помочь российскому Интернет-магазину собрать базу для email-рассылок.

Дано. Сырой текст в виде списка строк, некоторые из которых содержат электронные адреса

Требуется:

1. Выполнить лексический анализ текста, который заключается в разборе текста на отдельные слова.
2. Выбрать слова являющиеся электронными адресами и сохранить результат.

Что нужно знать?

1. Состав электронного адреса.
2. Какие алгоритмы используются при выполнении лексического анализа текста, как их реализовать на языке программирования?
3. Какие алгоритмы используются при выполнении отбора слов в тексте, как их реализовать на языке программирования?
4. Реализовать интерфейс программы

Сборник упражнений

1. Определить, является ли точка с координатами X, Y точкой пересечения диагоналей квадрата со стороной R , одна вершина которого расположена в начале координат.
2. Определить, лежит ли точка с координатами (X, Y) вне круга радиуса R с центром в точке (A, B) или внутри него.
3. Составить программу, которая бы реализовала следующий алгоритм: переменной T присвоить значение `true` если сочетание день, месяц, год образует правильную дату, и значение `false` – иначе (учитывая количество дней в месяце и название месяца).
4. Составить программу, которая бы реализовала следующий алгоритм: по порядковому номеру дня года определить дату, т. е. число и месяц.
5. В одномерном массиве определить первый отрицательный элемент и его номер.
6. Исключить из массива a_1, \dots, a_n первый отрицательный элемент.
7. Ввести массив a_1, a_2, \dots, a_{15} . Расположить ненулевые элементы по убыванию.
8. Ввести массив x_1, x_2, \dots, x_{20} . Элементы на нечетных местах, расположить в порядке возрастания, а на нечетных – в порядке убывания.
9. Ввести массив a_1, a_2, \dots, a_{15} . Требуется упорядочить его по возрастанию абсолютных значений элементов.
10. Ввести массив x_1, x_2, \dots, x_{20} . Требуется расположить отрицательные элементы в порядке убывания
11. Ввести упорядоченный по неубыванию массив. Найти количество различных чисел среди элементов этого массива.
12. Ввести два упорядоченных по возрастанию числовых массива X, Y . Найти количество общих элементов в этих массивах, то есть количество K таких чисел $X(i)=Y(j)$.
13. Перенести в хвост одномерного массива все отрицательные элементы.
14. Перенести в начало одномерного массива все нечетные элементы.
15. Составить алгоритм поиска максимального значения в двумерном массиве.

16. Составить алгоритм вычисления количества нечетных элементов в каждой строке двумерного массива.
17. Составить алгоритм вычисления суммы элементов двумерного массива $A(1..N, 1..M)$, расположенных выше главной диагонали.
18. Ввести двумерный массив $A(N, M)$. Составить алгоритм замены всех нулевых элементов на минимальный элемент.
19. Ввести двумерный массив $A(N, N)$. Составить алгоритм подсчета среднего арифметического значений двумерного массива
20. Составить алгоритм подсчета среднего арифметического значения двумерного массива. Вычислить отклонение от среднего для всех элементов двумерного массива.
21. Ввести двумерный массив $A(N, N)$. Составить алгоритм замены всех отрицательных элементов на среднее арифметическое значение элементов двумерного массива.
22. Составить алгоритм нахождения числа строк двумерного массива $A(N, N)$, количество отрицательных элементов в которых больше P .
23. Ввести двумерный массив. Найти наибольший элемент двумерного массива. Удалить строку с максимальным элементом.
24. Ввести двумерный массив. Поменять столбец с максимальным элементом с первым столбцом двумерного массива.
25. Ввести двумерный массив размером. Найти максимальный элемент двумерного массива, расположенный ниже побочной диагонали.
26. Ввести двумерный массив. Найти наименьший элемент двумерного массива. Перенести строку, содержащую этот элемент в конец.
27. Ввести двумерный массив размером. Найти минимальный элемент двумерного массива. Переставляя строки и столбцы, добиться того, чтобы элемент оказался в правом нижнем углу.
28. Дана квадратная матрица $A(N, N)$. Составить программу подсчета количества отрицательных элементов, расположенных ниже главной и ниже побочной диагонали.
29. Дана вещественная матрица $A(N, M)$. Составить программу нахождения минимального положительного элемента матрицы и нахождения его местоположения.
30. Дана вещественная матрица $A(N, M)$. Составить программу нахождения максимального отрицательного элемента матрицы и нахождения его местоположения.
31. Составить программу замены всех отрицательных элементов матрицы на 0, если сумма минимального и максимального элементов этой матрицы окажется меньше P .
32. Составить программу нахождения минимального положительного элемента в каждом столбце матрицы $A(N, M)$.
33. Дан массив $A(N, M)$. Определить количество «особых» элементов массива, считая элемент «особым», если он больше суммы остальных элементов своего столбца. Напечатать индексы «особых» элементов.
34. Дан массив $A(N, M)$. Определить количество «особых» элементов массива, считая элемент «особым», если в его строке слева от него находятся элементы меньше его, а справа – большие.

35. С помощью функции сортировки одномерного массива реализовать следующие задачи:

1. Расположить по возрастанию элементы каждой строки двумерного массива.
2. Расположить по убыванию элементы каждого столбца двумерного массива.

36. Найти и вывести все гласные буквы (без повторений), которые встретились в словах и количество слов.

37. Найти и вывести все гласные буквы (без повторений), которые встречаются в самом коротком слове.

38. Дана строка состоящая из буквы, цифры. Все цифры, входящие в самое длинное слово, заменить на символ «*».

39. Найти и вывести слово, содержащее наибольшее количество согласных букв.

40. Проверить является ли слово палиндромом. (Палиндром - это выражение (слово), которое читается одинаково слева направо и справа налево Например: ШАЛАШ или 12321).

41. Подсчитать количество слов и количество символов во всех словах.

42. Вывести все согласные буквы, содержащиеся в слове наибольшей длины, и вывести число повторений каждой буквы.

43. Даны два текста. Получить список символов, входящих в оба текста.

44. Задан текст и слово. Выяснить сколько раз слово встречается в тексте.

45. Задан текст и слово. Удалить это слово из текста.

46. Задан текст и два слова. Заменить в тексте первое слово на второе.

47. Задан текст. Разбить его на массив слов и напечатать буквы, встречающиеся в каждом слове.

48. Задан текст. Разбить его на массив слов и напечатать те слова, в которых гласные чередуются с согласными

49. Задан текст. Разбить его на массив слов, отсортировать список слов

50. Задан текст. Разбить его на массив слов, сколько слов начинаются на букву А

51. Задан текст. Разбить его на массив слов, определить среднюю длину слова

52. Задан текст. Разбить его на массив слов, сколько слов начинаются на согласную

53. Задан текст. Разбить его на массив слов, подсчитать сколько раз встречается каждое слово.

54. Построить КА с входным алфавитом $V = \{a, b, c, d\}$, распознающий все цепочки, начинающиеся и заканчивающиеся различными символами;

55. Построить КА с входным алфавитом $V = \{a, b, c\}$, распознающий все цепочки, в которых две последние буквы не совпадают;

56. Построить конечный автомат, распознающий язык L , состоящий из слов алфавита $\Sigma = \{a, b\}$, в которых буква a входит чётное число раз, а число вхождений буквы b делится на 3.

57. Каждое слово языка содержит подслово $*abc*$.

58. Построить ДКА, допускающий в алфавите $Z\{0,1\}$ все цепочки нулей и единиц, содержащие в себе множество цепочек, которые начинаются или оканчиваются (или и то и другое) последовательностью 01;

59. Построить ДКА, допускающий в алфавите $Z\{0,1\}$ все цепочки нулей и единиц, содержащие в себе множество всех цепочек, в которых всякая подцепочка из пяти последовательных символов содержит хотя бы два 0.

60. Каждое слово языка содержит подслово $*a?b*$. Алфавит $L = \{a, b, c\}$.

VIII. Перечень педагогических и информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (или модулю), включая перечень программного обеспечения и информационных справочных систем (по необходимости)

Процесс изучения дисциплины включает практические занятия и самостоятельную работу студента. Во время обучения применяется контактная технология преподавания (за исключением самостоятельно изучаемых студентами вопросов). При проведении занятий применяется имитационный подход (метод деловой игры, анализ конкретных ситуаций), когда преподавателем разбирается на конкретном примере проблемная ситуация, все шаги решения задачи студентам демонстрируются при помощи мультимедийной техники. Затем студенты самостоятельно решают аналогичные задания. Так же при проведении занятий применяется частично-поисковый метод: студенты осуществляют поиск решения поставленной проблемы (задачи). При этом, постановочные задачи опираются на уже имеющиеся у студентов знания и умения, полученные в предшествующих темах. На занятиях практикуется выполнение заданий в малых группах, письменные работы, работа с раздаточным материалом, привлекаются ресурсы сети Интернет. Курс предусматривает выполнение тестов, контрольных и самостоятельных работ, письменных домашних заданий. В качестве форм контроля используются различные варианты взаимопроверки и взаимоконтроля.

Программное обеспечение:

Наименование специальных* помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
<p>Помещение для самостоятельной работы, учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, Компьютерный класс математического факультета № 16 (Корпус 3, 170002, Тверская обл., г.Тверь, пер. Садовый, дом 35)</p>	<p>Google Chrome бесплатно Kaspersky Endpoint Security 10 для Windows Акт на передачу прав ПК545 от 16.12.2022 Lazarus бесплатно OpenOffice бесплатно Многофункциональный редактор ONLYOFFICE бесплатное ПО бесплатно ОС Linux Ubuntu бесплатное ПО бесплатно</p>
<p>Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, Учебная аудитория № 314 (Корпус 3, 170002, Тверская обл., г.Тверь, пер. Садовый, дом 35)</p>	<p>Google Chrome бесплатно Kaspersky Endpoint Security 10 для Windows Акт на передачу прав ПК545 от 16.12.2022 Lazarus бесплатно OpenOffice бесплатно Многофункциональный редактор ONLYOFFICE бесплатное ПО бесплатно ОС Linux Ubuntu бесплатное ПО бесплатно</p>

IX. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине (или модулю)

Наименование специальных* помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
<p>Помещение для самостоятельной работы, учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, Компьютерный класс математического факультета № 16 (Корпус 3, 170002, Тверская обл., г.Тверь, пер. Садовый, дом 35)</p>	<p>Набор учебной мебели, меловая доска, Компьютер INT Allegro, монитор Benq 24" GL2460 – 10 шт.</p>
<p>Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, Учебная аудитория № 314 (Корпус 3, 170002, Тверская обл., г.Тверь, пер. Садовый, дом 35)</p>	<p>Набор учебной мебели, меловая доска, Мультимедийный комплект учебного класса (вариант № 2): Проектор Casio XJ-140 настенный проекц. экран Lumien 180*180, Ноутбук Dell N4050, сумка 15,6", мышь; Усилитель Roxton AA-120; Радиосистема Shure PG288/PG58; Микшер Mackie 402 VLZ; Стационарный микрофон SOUNDKING EG002 с настольным держателем; Мультимедийный проектор Casio XJ-H2650 с потолочным креплением и моториз. экраном; Шкаф напольный 19".</p>

X. Сведения об обновлении рабочей программы дисциплины (или модуля)

№п. п.	Обновленный раздел рабочей программы дисциплины (или модуля)	Описание внесенных изменений	Дата и протокол заседания кафедры, утвердившего изменения
1.	V. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	Обновление списка литературы.	Протокол № 11 от 26.06.2013
2.	VII. Методические	Корректировка планов практических	Протокол № 10 от 24.06.2014

	указания для обучающихся по освоению дисциплины	(семинарских) занятий и методических рекомендаций к ним.	
3.	V. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	Обновление списка литературы. Обновление ссылок из ЭБС.	Протокол № 1 от 27.09.2015
4.	VII. Методические указания для обучающихся по освоению дисциплины.	Корректировка планов практических (семинарских) занятий и методических рекомендаций к ним.	Протокол № 1 от 01.09.2016
5.	I - X	Корректировка всех разделов в соответствии с новым стандартом	Протокол № 6 от 28.02.2017
6.	V. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	Дополнение списков. Обновление ссылок из ЭБС.	Протокол № 1 от 01.09.2017
7.	V. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	Дополнение списков. Обновление ссылок из ЭБС.	Протокол № 1 от 01.09.2023