



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ И РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

«СОГЛАСОВАНО»
Руководитель ОП



подпись

Пак Т.В.
ФИО

«УТВЕРЖДАЮ»

Заведующий кафедрой информатики,
математического и компьютерного
моделирования протокол



подпись

Чеботарев А.Ю.
ФИО

«11» июля 2019 г.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы алгоритмизации и структуры данных

Направление подготовки 02.03.01 Математика и компьютерные науки

(Сквозные цифровые технологии)

Форма подготовки очная

курс 1 семестр 1, 2

лекции 18 час.

практические занятия час.

лабораторные работы 36

в том числе с использованием МАО лек 9 / пр. 0 / лаб. 18 час.

всего часов аудиторной нагрузки 42 час.

в том числе с использованием МАО 0 час.

самостоятельная работа 90 час.

в том числе на подготовку к экзамену час.

контрольные работы (количество)

курсовая работа / курсовой проект

зачет 2

экзамен семестр

Рабочая программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта по направлению подготовки 02.03.01 Математика и компьютерные науки, утвержденного приказом Министерства образования и науки Российской Федерации от 23 августа 2017 № 807

Рабочая учебная программа обсуждена на заседании кафедры информатики, математического и компьютерного моделирования, протокол № 18 от «09» июля 2019 г.

Заведующий кафедрой информатики, математического и компьютерного моделирования протокол
Чеботарев А.Ю.

Составитель: ст.преподаватель И.А.Малыкина

Владивосток
2019

Оборотная сторона титульного листа РПД

II. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «__» _____ 202_ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

III. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «__» _____ 202_ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании кафедры:

Протокол от «__» _____ 202_ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

Аннотация к рабочей программе дисциплины «Алгоритмы и структуры данных»

Дисциплина относится к модулю «Алгоритмизация и программирование» основной образовательной программы подготовки бакалавров по направлению подготовки 01.04.02 Прикладная математика и информатика.

Знания, полученные по освоению дисциплины, необходимы для успешного освоения курсов: Языки и методы программирования, Введение в Web-программирование, Базы данных, Программное обеспечение обработки данных, Технология разработки программного обеспечения, а также, при выполнении учебной и производственной практики и выпускной квалификационной работы.

Общая трудоемкость освоения дисциплины составляет 4 зачетных единиц, 144 (контакт. 42) часа. Учебным планом предусмотрены лекции, лабораторные работы и самостоятельная работа студента. Дисциплина реализуется на 1 курсе в 1 и 2 семестре. Программа является продолжением программы 1 семестра.

Цель: получение фундаментальных основ и навыков алгоритмизации и программирования. Знакомство с технологическим циклом создания программного продукта и подготовка к решению прикладных задач программирования из любой предметной области с использованием любого подходящего языка программирования.

Задачи:

- изучить понятие алгоритма, свойства алгоритма, типы алгоритмических моделей, способы описания алгоритмов
- получить представление об основах программирования и этапах решения задачи программирования;
- владеть языками программирования;
- знать стандартные алгоритмы, лежащие в основе решения задач программирования и уметь применять их на практике;
- овладеть практическими навыками решения задач, начиная от ее постановки и формализации и заканчивая отладкой и тестированием.
- научиться методам практической реализации программ на примере языков программирования C++, Python, PascalABC.

Для успешного изучения дисциплины «Основы алгоритмизации и структуры данных» у обучающихся должны быть сформированы следующие предварительные компетенции:

- Способность управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования
- Способность осуществлять социальное взаимодействие и реализовывать свою роль в команде

В результате изучения данной дисциплины у обучающихся формируются следующие общекультурные/ общепрофессиональные/ профессиональные компетенции (элементы компетенций).

Код и формулировка компетенции	Этапы формирования компетенции
<p>ОПК-3 Способен самостоятельно представлять научные результаты, составлять научные документы и отчеты</p>	<p>ОПК-3.1 Знать: алгоритмические и программные решения в области системного и прикладного программного обеспечения Знает фундаментальные идеи и понятия из разделов курса и основы программирования; стандартные алгоритмы, лежащие в основе решения задач программирования; основные понятия, структуры и инструментарий, которые применяются в языках программирования, основные структуры и типы данных, основные методы проектирования и разработки компьютерных программ.</p> <p>ОПК-3.2 Уметь: разрабатывать и применять алгоритмические и программные решения в области системного и прикладного программного обеспечения Умеет применять стандарты оформления технической документации на различных стадиях жизненного цикла программной системы.</p> <p>ОПК-3.3 Владеть: навыками и методами разработки, и применения алгоритмических и программных решений в области системного и прикладного программного обеспечения Владеет технологическим циклом создания программного продукта и подготовки к решению прикладных задач программирования из любой предметной области с использованием любого подходящего языка программирования; методами практической реализации программ на языках программирования Python, C++, Pascal и др.</p>
<p>ОПК-4</p> <p>Способен комбинировать и адаптировать существующие информационно-коммуникационные технологии для решения задач в области профессиональной деятельности с учетом требований информационной безопасности</p>	<p>ОПК-4.1 знает основные методики и технологии использования ИКТ в профессиональной деятельности с учетом основных требований информационной безопасности.</p> <p>ОПК-4.2 умеет решать типовые задачи профессиональной деятельности с использованием ИКТ и с учетом основных требований информационной безопасности.</p> <p>ОПК-4.3 владеет навыками использования ИКТ в профессиональной деятельности с учетом основных требований информационной безопасности.</p>

I. СТРУКТУРА И СОДЕРЖАНИЕ ЛЕКЦИОННОЙ ЧАСТИ КУРСА

Темы лекций

- 1 тема. Конечные автоматы на примере разбора текстовых цепочек.
- 2 тема. Длинная арифметика
 - Ввод и вывод длинных чисел
 - Хранение
 - Операции сложения, сравнения, вычитания, умножения
- 3 тема. Алгоритмы сжатия данных
 - Равномерное кодирование
 - RLE
 - Код Шеннона-Фано
 - Код Хаффмана
 - Алгоритм Jpeg, Png
- 4 тема. Передача данных
 - Помехоустойчивый код, расстояние Хэмминга
 - Бит четности, утроение бита
 - Код Хэмминга
- 5 тема. Шифрование
 - Симметричный шифр
 - Шифр Цезаря, Виженера
 - Хэширование и пароли
 - Алгоритм RSA
 - Стеганография
- 6 тема. Динамические структуры
 - Связные списки
 - Стек
 - Очередь
 - Кольцо
 - Дек
- 7 тема. Деревья.
 - Двоичные деревья. Представление.
 - Прямой, обратный и симметричный обходы.
 - Деревья двоичного поиска.
 - Дерево арифметических выражений
 - Сбалансированное дерево
 - Добавление и удаление узлов
- 8 тема. Графы.
 - Представление графов.
 - Остовное дерево наименьшей стоимости.
 - Метод поиска в глубину.
 - Задача Штейнера.

- Алгоритм Беллмана-Форда.
- Алгоритм Дейкстры
- Алгоритм Прима Крускала
- Задача коммивояжера
- Паросочетания. Задачи о наибольшем паросочетании.
- Плоская укладка графа

9 тема. Хеширование.

10 тема. Введение в ООП

II. ЛАБОРАТОРНЫЕ РАБОТЫ

Лабораторная работа №1. Программа как единство алгоритма и структуры данных. Данные базового типа (целые, вещественные, символьные, логические). Структура хранения данных в ЭВМ. Определение типов и выражения с данными базового типа в языке Паскаль. Механизмы приведения типов. (2 ч.)

Лабораторная работа №2. Операции и их приоритет в языке Паскаль. Преобразование типов. Перечислимый и диапазонный типы. Введение в систему конструируемых типов языка. Основные алгоритмические структуры. Операторы ветвления и цикла в языке Паскаль. (4 ч.)

Лабораторная работа №3. Вычислительные задачи.

Точность вычислений. Погрешности. Решение уравнений методом перебора, деления отрезка пополам. Дискретизация. Вычисление длины кривой. Вычисление площади фигуры методами прямоугольников и трапеций. (6 ч.)

Лабораторная работа №4. Массивы. Строковые структуры данных. Хранение и алгоритмы их обработки. (6 ч.)

Лабораторная работа №5. Ввод\вывод данных. Файловая система. Логические и физические файлы. Общие процедуры работы с файлами. Типизированные и нетипизированные файлы. (4 ч.)

Лабораторная работа №6. Процедуры и функции. Механизм передачи параметров. Глобальные и локальные описания. Области объявления и области действия имен. (4 ч.)

Лабораторная работа №7. Данные типа "запись". Механизм хранения и организации. Массив записей. Массив как компонента записи. (4 ч.)

Лабораторная работа №8. Алгоритмы сортировки: включением, выбором, обменом, трехленточное слияние и др. Алгоритм двоичного поиска. Алгоритм Ктута, Мориса, Пратта поиска подстроки в строке. Организация таблиц расстановки и функции расстановки (метод хеширования). (6 ч.)

Лабораторная работа №9. Длинная арифметика

1. Ввод и вывод длинных чисел
2. Операции сложения, сравнения, вычитания, деления (8 ч.)

Лабораторная работа №10.

Лабораторная работа №11. Динамическая память. Адреса и указатели. Объявление указателей. Выделение и освобождение динамической памяти. Анализ и управление состоянием кучи. Организация структур. (10 ч.)

Лабораторная работа №12. Создание и просмотр динамического списка. Добавление и удаление элементов. Списки типа стек, очередь и основные операции с ними. (18 ч.)

Лабораторная работа №13. Деревья. Основные понятия. Двоичные деревья. Идеально сбалансированные деревья. Упорядоченные деревья. Основные операции (обход дерева, добавление и удаление элемента). Сортировка с помощью дерева. (10 ч.)

Лабораторная работа №14. Графы. Основные понятия. Ориентированные и неориентированные графы. Способы представления графов. (6 ч.)

Лабораторная работа №15. Задачи на графах (Прима-Краскала, Штейнера, Дейкстры, коммивояжера). (10 ч.)

Лабораторная работа №16. Работа с памятью, в том числе с внешней. Алгоритм Дойча. Метод близнецов. Выделение памяти разного размера. Уплотнение памяти. (10 ч.)

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Материалы лекций (включая авторские видеоматериалы), перечень заданий с описанием требований и формы выполнения, график работ и рейтинг студентов опубликованы на платформе **GOGLE DRIVE**. Для сдачи работ, их комментирования и оценивания используется функционал ресурса.

При необходимости, могут быть использованы платформы онлайн конференций **MS TEAMS** или **ZOOM**

Для изучения дисциплины приводится перечень рекомендуемой литературы, методические указания и вопросы к контрольным заданиям и экзамену.

В качестве основы для изучения дисциплины можно взять учебники, учебные пособия, электронные материалы и методические указания, приведенные в списке литературы.

В Приложении 1 представлены:

план-график выполнения самостоятельной работы по дисциплине;

характеристика заданий для самостоятельной работы обучающихся;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина «Основы алгоритмизации и структуры данных» является базисом для программиста любого профиля.

Процесс изучения дисциплины осуществляется в следующих организационных формах:

- выполнение аудиторных лабораторных работ;
- самостоятельное изучение материала;
- выполнение контрольных работ;
- подготовка и сдача экзамена и зачета.

В дисциплине можно выделить две области:

- базовые знания, относительно стабильные, составляющие ядро дисциплины;
- технологические знания, связанные с освоением конкретных программных сред и языков программирования.

Базовые знания основных принципов алгоритмизации, понимание процесса работы программы, обработки компьютером данных образуют понятийное ядро дисциплины и служат основой для изучения многих дисциплин специальности. Эта область включает в себя системный подход к решению информационных задач, алгоритмическое мышление, знание терминологии и современных средств разработки программного обеспечения.

Технологическая часть дисциплины связана с практическим освоением умений и навыков построения алгоритмов и программирования в наиболее распространенных программных средах. Отдельное внимание на занятиях уделяется различным способам организации данных в программе, решению стандартных алгоритмических задач.

Лабораторные работы проводятся в компьютерных классах и подкреплены методическими указаниями, рекомендациями и требованиями к представлению и оформлению результатов работы.

Самостоятельная работа включает изучение теоретического материала дисциплины и выполнение индивидуальных работ.

Для изучения дисциплины приводится перечень рекомендуемой литературы, методические указания и вопросы к контрольным заданиям и экзамену.

В качестве основы для изучения дисциплины можно взять учебники, учебные пособия, электронные материалы и методические указания, приведенные в списке литературы.

При изучении теоретического материала следует по методическим указаниям ознакомиться с планом темы. Освоив теоретический материал, необходимо самостоятельно, без помощи литературы, сделать попытку ответить на вопросы по теме. С каждой темой связан перечень ключевых понятий. После изучения темы необходимо уметь самостоятельно давать определение понятий.

1. Рекомендации по планированию и организации времени, необходимого для изучения дисциплины.

Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины: Изучение конспекта лекции в тот же день после лекции – 10-15 минут.

Повторение лекции за день перед следующей лекцией – 10-15 минут. Изучение теоретического материала по учебнику и конспекту – 1 час в неделю.

Подготовка к лабораторному занятию и работе в компьютерном классе – 1 час.

Тогда общие затраты времени на освоение курса студентами составят около 2,5 часа в неделю.

2. Описание последовательности действий студента («сценарий изучения дисциплины»). При изучении численных методов следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий:

После окончания учебных занятий для закрепления материала просмотреть и обдумать текст лекции, прослушанной сегодня, разобрать рассмотренные примеры (10-15 минут).

При подготовке к лекции следующего дня повторить текст предыдущей лекции, подумать о том, какая может быть следующая тема (10- 15 минут).

В течение недели выбрать время для работы со специальной литературой в библиотеке и для занятий на компьютере (по 1 часу).

При подготовке к лабораторным занятиям следующего дня необходимо сначала прочитать основные понятия и теоремы по теме домашнего задания. При выполнении задания нужно сначала понять, что требуется в задаче, какой теоретический материал нужно использовать, наметить план решения задачи.

3. Рекомендации по использованию материалов учебно-методического комплекса.

Рекомендуется использовать методические указания и материалы по курсу, текст лекций, а также электронные пособия, имеющиеся на сервере Школы естественных наук.

4. Рекомендации по работе с литературой.

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекций изучаются и книги. Литературу по курсу желательно изучать в библиотеке. Полезно использовать несколько учебников, однако легче освоить курс, придерживаясь одного учебника и конспекта. Рекомендуется, кроме «заучивания» материала, добиться понимания изучаемой темы дисциплины. Кроме того, очень полезно мысленно задать себе и попробовать ответить на следующие вопросы: о чем эта глава, какие новые понятия в ней введены.

5. Советы по подготовке к экзамену.

Дополнительно к изучению конспектов лекций необходимо пользоваться учебниками. Вместо «заучивания» материала важно добиться понимания изучаемых тем дисциплины. При подготовке к экзамену нужно освоить теорию: разобрать определения всех понятий и численных методов, рассмотреть примеры и самостоятельно решить несколько типовых задач из каждой темы. При решении задач всегда необходимо комментировать свои действия и не забывать о содержательной интерпретации.

6. Указания по организации работы с контрольно-измерительными материалами.

При подготовке к лабораторной работе, коллоквиуму, контрольной работе необходимо сначала прочитать теорию по каждой теме. Отвечая на поставленный вопрос, предварительно следует понять, что требуется от Вас в данном случае, какой теоретический материал нужно использовать, наметить общий план решения.

VI. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература

(электронные и печатные издания)

1. Семакин И.Г. Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования М. Издательский центр «Академия», 2018. 304 с.

2. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2015. -

416 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0279-0 - Режим доступа: <http://znanium.com/catalog/product/484837>

3. Т.Кормен, Ч.Лейзерсон, Р.Ривест, К.Штайн - Алгоритмы. Построение и анализ. Издание 3-е (2013)

4. С.М. Окулов -Программирование в алгоритмах-БИНОМ. Лаборатория знаний (2014)

5. Искусство программирования для ЭВМ : пер. с англ. . т. 1 . Основные алгоритмы / Д. Кнут ; пер. Г. П. Бабенко, Ю. М. Баяковский. Кнут, Дональд Эрвин. Издание; 2-е изд. Место публикации; Москва Издатель; Вильямс Год; 2007. Физическое описание; 735 с.

6. [Искусство программирования : пер. с англ. . т. 3 . Сортировка и поиск / Д. Э. Кнут ; под общ. ред. Ю. В. Козаченко. Издание; 2-е изд. Место публикации; Москва Издатель; Вильямс Год; 2012. Физическое описание; 822 с.](#)

7. [С/С++. Программирование на языке высокого уровня :учебник / Т. А. Павловская. Санкт-Петербург : Питер, 2011. 461 с.](#)

8. [Паскаль : Программирование на языке высокого уровня : учебник для вузов / Т. А. Павловская. Санкт-Петербург : Питер, 2010.](#)

9. [С/С++. Структурное программирование. Практикум : учебное пособие / Т. А. Павловская, Ю. А. Щупак. Санкт-Петербург : Питер, 2010.](#)

10. Построение и анализ вычислительных алгоритмов : [монография] / А. Ахо, Дж. Хопкрофт, Дж. Ульман ; пер. с англ. А. О. Слисенко. Ахо, Альфред. Место публикации; Москва Издатель; Мир Год; 2012. Физическое описание; 536 с.

11. Кормен, Лейзерсон, Ривест: Алгоритмы. Построение и анализ / Издательство: Диалектика, 2019 г.

12. Алгоритмы и структуры данных с примерами на Паскале / Никлаус Вирт ; [пер. с англ. Д. Б. Подшивалова]. Вирт, Никлаус. Издание; [2-е изд.]. Место публикации; Санкт-Петербург Издатель; [Невский Диалект] Год; 2008. Физическое описание; 351 с.

13. Основы программирования : [учебник] / С. Окулов. Окулов, Станислав Михайлович. Издание; 4-е изд. Место публикации; Москва Издатель; Лаборатория знаний Год; 2008. Физическое описание; 440 с.

Дополнительная литература

(печатные и электронные издания)

14. Дж. Ахо, Алгоритмы и структуры данных

15. Н. Вирт, Алгоритмы и структуры данных

16. Н. Вирт, Систематическое программирование

17. Н. Вирт, Алгоритмы + структуры данных = программа

18. П. Холл, Вычислительные структуры

19. В. Бондарев, Основы программирования

20. [Ахо А., Хопкрофт Д., Ульман Д., Построение и анализ вычислительных алгоритмов.](#)

21. Дейкстра Э., Дисциплина программирования.

22. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ.

23. Скопин И.Н., Основы конструирования программ и языки программирования.

Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

24. Язык PascalABC.NET, Современное программирование на языке Паскаль, <http://pascalabc.net/>

25. Programming Taskbook, Электронный задачник по программированию, <http://www.ptaskbook.com/ru/>

26. Система автоматического тестирования программ, <http://imcs.dvfu.ru/cats/>

Перечень информационных технологий и программного обеспечения

PascalABC.NET + Microsoft .NET Framework v4.0 (Setup, 66 Mb)

Система программирования PascalABC.NET

Интегрированная среда разработки Visual Studio

- Задачник РТ4
- Платформа Microsoft .NET Framework v4.0
- Russian Language Pack for .NET v4.0 (русификация сообщений об ошибках времени выполнения)
- Framework Class Library Help (документация для всплывающей подсказки)
- Автоматическая тестирующая система Московского центра непрерывного математического образования www.informatics.mcsme.ru
- Автоматическая тестирующая система ДВФУ <https://imcs.dvfu.ru/cats/>

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Компьютерный класс: 15 Моноблоков/HP-В0G08ES#ACB| HP 8200E AiO i52400S 500G 4/0G 28PC

Проектор 3-chip DLP, 10 600 ANSI-лм, WUXGA 1 920x1 200 (16:10) PT-DZ110XE Panasonic; экран 316x500 см, 16:10 с эл. приводом; крепление настенно-потолочное Elpro Large Electrol Projecta; профессиональная ЖК-панель 47", 500 Кд/м2, Full HD M4716CCBA LG; подсистема видеоисточников документ-камера CP355AF Avertvision; подсистема видеокоммутации; подсистема аудиокоммутации и звукоусиления; подсистема интерактивного управления; беспроводные ЛВС обеспечены системой на базе точек доступа

Корпус 20, ауд. D 734, 734а



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ
РАБОТЫ ОБУЧАЮЩИХСЯ**
по дисциплине **Алгоритмы и структуры данных**
Направление подготовки 02.03.01 Математика и компьютерные науки
Сквозные цифровые технологии

Владивосток
2020

Самостоятельная работа студентов состоит из подготовки к лабораторным работам в компьютерном классе, работы над рекомендованной литературой и текстами лекций в процессе изучения теоретического материала, выполнения заданий для самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Алгоритмизация и программирование	ОПК-3; ОПК-4	знает	Контрольная работа (КР-1)	Зачет
			умеет	Контрольная работа (КР-1)	
			владеет	Лабораторная работа (ЛР-1)	
2	Алгоритмы поиска и сортировки	ОПК-3; ОПК-4	знает	Коллоквиум (КР-1)	Зачет
			умеет	Коллоквиум (КР-2)	
			владеет	Лабораторная работа (ЛР-3)	
3	Динамические структуры	ОПК-3; ОПК-4	знает	Самостоятельная работа (ЛР-4)	Зачет
			умеет	Самостоятельная работа (КР-4)	
			владеет	Лабораторная работа (ЛР-4)	
4	Алгоритмы на графах	ОПК-3; ОПК-4	знает	Самостоятельная работа (ЛР-5)	Зачет
			умеет	Самостоятельная работа (ЛР-6)	
			владеет	Лабораторная работа (ЛР-6)	

Рекомендации по самостоятельной работе студентов

Самостоятельная работа включает изучение теоретического материала дисциплины и выполнение индивидуальных работ.

Материалы лекций (включая авторские видеоматериалы), перечень заданий с описанием требований и формы выполнения, график работ и рейтинг студентов опубликованы на платформе **GOOGLE DRIVE**. Для сдачи работ, их комментирования и оценивания используется функционал ресурса, автоматические системы тестирования **CATS-DVFU, Informatics.msk.ru**

При необходимости, могут быть использованы платформы онлайн конференций **MS TEAMS** или **ZOOM**.

Для изучения дисциплины приводится перечень рекомендуемой литературы, методические указания и вопросы к контрольным заданиям и экзамену.

В качестве основы для изучения дисциплины можно взять учебники, учебные пособия, электронные материалы и методические указания, приведенные в списке литературы.

При изучении теоретического материала следует по методическим указаниям ознакомиться с планом темы. Освоив теоретический материал, необходимо самостоятельно, без помощи литературы, сделать попытку выполнить практическую работу. С каждой темой связан перечень ключевых понятий. После изучения темы необходимо уметь самостоятельно давать определение понятий, иметь навыки и умения выполнять практические задания.

Требования к представлению и оформлению результатов самостоятельной работы

Результатом самостоятельной работы являются отчеты по лабораторным работам.

В процессе подготовки отчетов к лабораторным работам у студентов развиваются навыки составления письменной документации и систематизации имеющихся знаний. При составлении отчетов рекомендуется придерживаться следующей структуры:

- Постановка задачи;
- Метод решения;
- Алгоритм метода;
- Спецификация используемых функций и типов данных;
- Описание тестов, на которых программа проходила проверку;

Критерии оценки выполнения самостоятельной работы

Отчет по лабораторной работе должен полностью удовлетворять условию задачи. В случае некачественно выполненных отчетов (не соответствующих заявленным требованиям) результирующий балл за работу может быть снижен. Студент должен продемонстрировать отчетливое и свободное владение концептуально-понятийным аппаратом, научным языком и терминологией. Наличие всех отчетов является допуском к экзамену.

На экзамене оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.

Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.

Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.

Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Оценка по дисциплине может быть выставлена по результатам коллоквиумов и контрольных работ. При этом критерии оценки те же, что и на экзамене.

Методические указания

Все методические указания по типам задач размещены на ресурсе дисциплины и включают теоретическое описание проблемы и решений, содержание задач, пояснения к их выполнению и требования.

Примеры заданий:

Задание **Моделирование стека.**

Цель работы: Отработка практических навыков по разработке основных этапов решения задачи моделирования стека.

Рекомендации к самостоятельной работе:

Изучить материал лекции «Понятие о структурах данных. Моделирование ряда структур данных: стека, очереди, списка».

Содержание работы:

1. Заполнить стек 10 случайными числами из интервала $[-10;20]$. Просмотреть содержимое стека. Найти сумму положительных чисел, хранящихся в стеке.

2. Сформировать стек из 5 чисел. Найти произведение 3-го и 4-го чисел из стека. Результат поместить в стек.

3. Заполнить стек 10 случайными числами из интервала $[-10;20]$. Найти максимальное число.

Форма представления отчета:

Отчет представить в письменном виде, который должен содержать: алгоритм в виде блок-схемы, программу и результат ее выполнения;

Задание Рекурсия.

Цель работы: Изучить рекурсивные алгоритмы.

Рекомендации к самостоятельной работе:

Изучить материал лекции «Рекурсия».

Содержание работы:

1. Вычислить $(a! + b!)/a!$, используя рекурсивную функцию вычисления факториала
2. Вычислить $(1+2+3+4+5)/(1+2+3+4+5+6+7+8)$, используя рекурсивную функцию вычисления суммы первых n натуральных чисел.
3. Составить рекурсивную функцию вычисления n -го члена последовательности: $a_1=0$, $a_i = 2*a_{i-1}+i$. Найти произведение 3-го и 7-го членов последовательности.
4. Составить рекурсивную функцию нахождения суммы n членов арифметической прогрессии 1, 3, ... Найти сумму с 5-го по 10-й членов прогрессии

Форма представления отчета:

Отчет представить в письменном виде, который должен содержать: алгоритм в виде блок-схемы, программу и результат ее выполнения;



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

НАЗВАНИЕ ШКОЛЫ (ФИЛИАЛА)

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине Алгоритмы и структуры данных
Направление подготовки 02.03.01 Математика и компьютерные науки
Сквозные цифровые технологии
Форма подготовки очная

Владивосток
2020

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		Критерии	Показатели
ПК-3 способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	Знает (пороговый уровень)	базовые принципы разработки и отладки программного обеспечения, этапы разработки простого ППО	демонстрирует теоретическую подготовку	2 из 3 ответов правильны
	Умеет(продвинутый)	разрабатывать поэтапно ППО простой и средней сложности.	применяет знания на практике	решает в основном правильно практическое задание
	Владеет (высокий)	практическим опытом разработки ППО средней сложности	быстро выбирает метод решения	качественно выполняет задание аналитического характера
ПК-4 способен к обоснованному выбору, проектированию и внедрению специальных технических и программно-математических средств в избранной профессиональной области	Знает(пороговый уровень)	базовые алгоритмы обработки информации различных типов. Принципы оценки сложности алгоритмов	демонстрирует теоретическую подготовку	2 из 3 ответов правильны
	Умеет(продвинутый)	использовать набор инструментальных средств для разработки программного обеспечения	применяет знания на практике	решает в основном правильно практическое задание
	Владеет(высокий)	выполнять анализ существующих решений и обосновывать принятые решения	быстро выбирает метод решения	качественно выполняет задание аналитического характера

Текущая аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ в форме коллоквиумов, контрольных и лабораторных работ по оцениванию фактических результатов обучения студентов. Объектами оценивания выступают:

- учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);
- степень усвоения теоретических знаний;
- уровень овладения практическими умениями и навыками;
- результаты самостоятельной работы.

Промежуточная аттестация студентов по дисциплине проводится в соответствии с локальными нормативными актами ДВФУ в виде экзамена в устной форме.

Критерии выставления оценки студенту на экзамене

Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал монографической литературы, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.

Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.

Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ.

Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

Оценка по дисциплине может быть выставлена по результатам коллоквиумов и контрольных работ. При этом критерии оценки те же, что и на экзамене.

Оценочные средства для промежуточной аттестации

Примеры зачетных промежуточных заданий.

Описать идею указанного алгоритма, дать оценку его алгоритмической сложности. Рассмотреть особенности реализации и область применимости:

- Описать функцию, которая вычисляет среднее арифметическое элементов
- непустого списка.
- Описать функцию, которая меняет местами первый и последний элементы
- непустого списка.
- Описать функцию, которая вставляет новый элемент перед каждым
- вхождением заданного элемента.
- Описать функцию, которая проверяет на равенство списки $L1$ и $L2$:
- Описать функцию, которая определяет, входит ли список $L1$ в список $L2$;
- Описать функцию, которая переносит в конец непустого списка L его первый
- элемент;
- Описать функцию, которая переносит в начало непустого списка L его
- последний элемент;
- Описать функцию, которая копирует в список L за каждым вхождением
- заданного элемента все элементы списка $L1$.
- Описать процедуру, которая объединяет два упорядоченных по неубыванию
- списка $L1$ и $L2$ в один упорядоченный по неубыванию список, сменив
- соответствующим образом ссылки в $L1$ и $L2$.
- Описать функцию, которая проверяет, упорядочены ли элементы списка по алфавиту.
- сортировка бинарным деревом
- шейкерная сортировка

- циклическая перестановка списка сортировка
- сортировка стеком
- сортировка TimSort
- сортировка вставками
- сортировка выбором
- сортировка слияние
- сортировка с помощью очереди
- быстрая сортировка
- сортировка четн/нечетн
- сортировка JSORT
- цифровая (поразрядная) сортировка
- блочная сортировка
- сортировка Шелла
- поиск Армстронга,
- градиентный спуск,
- поразрядный поиск
- бинарный поиск,
- интерполяционный поиск,
- поиск подстроки в строке
- поиск Ахо-Карасик
- двоичный поиск,
- индексный поиск,
- случайный поиск
- линейный поиск,
- троичный поиск
- метод касательных,
- метод приближения,
- метод хорд

- алгоритм Эратосфен,
- последовательный поиск
- обезьяний поиск,
- алгоритм Дейкстры,
- поиск МиниМакс
- поиск Фиббоначчи,
- поиск по бору,
- поиск Мориса-Пратта
- Кеширование

Этапы выполнения работы.

1. Выполнить необходимую формализацию постановки задачи.
 2. Провести исследование возможности применения известных алгоритмов к решению данной задачи.
 3. Подготовить проверочные тесты.
 4. Записать и отладить код программы. Провести тестирование.
 5. Оформить отчет, содержащий основные результаты работы.
- В письменном отчете должны содержаться:
1. Постановка задачи. Исходные данные.
 2. Обоснование возможности применения данного алгоритма.
 3. Программа, реализующая поставленную задачу.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

НАЗВАНИЕ ШКОЛЫ (ФИЛИАЛА)

КОНТРОЛЬНО-ИЗМЕРИТЕЛЬНЫЕ МАТЕРИАЛЫ
по дисциплине Алгоритмы и структуры данных
Направление подготовки 02.03.01 Математика и компьютерные науки
Сквозные цифровые технологии
Форма подготовки очная

г. Владивосток
2020

Темы экзаменационных вопросов

1. Поиск подстроки в строке.
Префикс-функция
Алгоритм Кнута, Морриса, Пратта.
Алгоритм Бойера и Мура
2. Длинная арифметика
Ввод и вывод длинных чисел
Операции сложения, сравнения, вычитания, умножения
3. Конечные автоматы на примере разбора текстовых цепочек.
4. Алгоритмы сжатия данных
Равномерное кодирование
RLE
Код Шеннона-Фано
Код Хаффмана
Алгоритм Jpeg, Png
5. Передача данных
Помехоустойчивый код, расстояние Хэмминга
Бит четности, утроение бита
Код Хэмминга
6. Шифрование
Симметричный шифр
Шифр Цезаря, Виженера
Хэширование и пароли
Алгоритм RSA
Стеганография
7. Динамические структуры
Связные списки
Стек
Очередь
Кольцо
Дек
8. Деревья.
Двоичные деревья. Представление.
Прямой, обратный и симметричный обходы.
Деревья двоичного поиска.
Дерево арифметических выражений
Сбалансированное дерево
Добавление и удаление узлов
9. Хэширование.
10. Графы.
Алгоритмы на графах.
Остовное дерево наименьшей стоимости.
Метод поиска в глубину.

Алгоритм нахождения кратчайшего пути.
Задача Штейнера.
Паросочетания.

Образец экзаменационного билета

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

«Дальневосточный федеральный университет»

(ДФУ)

Школа естественных наук

02.03.0 Математика и компьютерные науки

Дисциплина	<i>Основы информатики</i>
Форма обучения	<i>очная</i>
Семестр	<i>осенний 2019 - 2020 учебного года</i>
Реализующая кафедра	<i>информатики, математического и программного обеспечения</i>

Экзаменационный билет № 16

1. Универсальный исполнитель. Машина Тьюринга.

2. Задача №113657 (*informatics.msk.ru*)

Дана строка, содержащая только английские буквы (большие и маленькие) и открывающиеся скобки. Сформировать новую строку добавлением справа «зеркальной» строки с закрывающимися скобками. "(abc(def(g" -> "(abc(def(gg)fed)cba)"

Составитель, ст. преподаватель _____ И.А.Малькина

Зав. кафедрой информатики, математического

и программного обеспечения _____ А.Ю.Чеботарев

Пример тестов для промежуточной проверки

1. Задан массив $X[1..N]$. Какое условие надо поставить вместо многоточия, чтобы в результате в переменную i был записан номер элемента, равного R ? Вводите ответ без пробелов.

```
i:=1  
while (i<=N) and ( ... ) do
```

```
i:=i+1;
```

Ответ:

2. Задан массив $X[1..N]$. Какое условие надо поставить вместо многоточия, чтобы найти минимальный элемент массива в переменной M ? Вводите ответ без пробелов.

```
M:=X[1];
```

```
for k:=2 to N do
```

```
if ... then M := X[k];
```

Ответ:

3. Задан массив $X[1..N]$. Какой оператор надо поставить вместо многоточия, чтобы найти номер максимального элемента массива в переменной M ? Вводите ответ без пробелов.

```
M:=1;
```

```
for k:=2 to N do
```

```
if X[k]>X[M] then
```

```
...
```

Ответ:

4. Требуется поменять местами значения элементов массива $X[a]$ и $X[b]$. Какой оператор нужно добавить вместо многоточия? В ответе не используйте пробелы.

```
p := X[a];
```

```
...
```

```
X[b] := p;
```

Ответ:

5. Задан массив $X[1..N]$. Какой оператор надо поставить вместо многоточия, чтобы в результате в переменную nR был записан номер элемента, равного R ? Вводите ответ без пробелов.

```
nR:= 0;
```

```
for k:=1 to N do
```

```
if X[k]=R then begin
```

```
...
```

```
break
```

end;

Ответ:

6. Задан массив $X[1..N]$. Что нужно поставить вместо многоточия, чтобы в результате элементы в массиве X были переставлены в обратном порядке?
for k:=1 to ... do begin

c:=X[k];

X[k]:=X[N+1-k];

X[N+1-k]:=c

end;

Ответ:

7. Задан массив $X[1..N]$. Какой оператор нужно поставить вместо многоточия, чтобы в результате элементы в массиве X были сдвинуты на один вправо (циклически)? В ответе не используйте пробелы.

c:=X[N];

for k:=N downto 2 do begin

...

end;

X[1]:=c;

Ответ:

8. Требуется выделить все отрицательные элементы массива $A[1..N]$ в начало массива $B[1..N]$. Какой оператор надо вставить в программу вместо многоточия? Вводите ответ без пробелов.

c:=1;

for k:=1 to N do

if A[k]<0 then begin

...

c:=c+1

end;

Ответ:

9. Что будет выведено на экран после выполнения программы?

s := '123';

s := s + '0' + s;

s := s + s;

writeln (s);

Ответ:

10. Что будет выведено на экран после выполнения этой программы?

s := '123';

s := s + s[2] + s + s[3];

```
writeln ( s );
```

Ответ:

11. Что будет выведено на экран после выполнения этой программы?

```
s := '#12345#';  
n := Length(s) +  
Length('#456#');  
writeln ( n );
```

Ответ:

12. Что будет выведено на экран после выполнения программы?

```
s := '#123456789#';  
s1 := '#abcdef#';  
s := Copy(s,5,2) + '#0#'; +  
Copy(s1,3,3);  
writeln ( s );
```

Ответ:

13. Что будет выведено на экран после выполнения этой программы?

```
s := '#123456#';  
q := '#abc#';  
Delete ( s, 2, 3 );  
Insert ( s, q, 2 );  
writeln ( q );
```

Ответ:

14. Что будет выведено на экран после окончания работы программы?

```
s := '#123456789#';  
Delete ( s, 1, 2 );  
n := Pos ( '#6#', s );  
writeln ( n );
```

Ответ:

15. Что будет выведено на экран после выполнения этой программы?

```
s := '#123456789#';  
n := Pos ( '#456#', s );  
Delete ( s, n+2, 3 );  
writeln ( s );
```

Ответ:

16. Сколько единиц будет выведено на экран?

```
s := '#123456#';  
while Length(s) > 0 do begin  
Delete ( s, 1, 2 );
```

```
writeln ( '#1' );  
end;
```

Ответ:

17. Программа должна считать, сколько раз в строке S встречается подстрока '#12'. Какой оператор надо вставить вместо многоточия? Вводите ответ без пробелов.

```
c := 0;  
repeat  
...  
if p <> 0 then begin  
  c := c + 1;  
  Delete ( S, 1, p+1)  
end;  
until p = 0;
```

Ответ:

18. Программа должна считать, сколько раз в строке S встречается цифра '#0'. Какой оператор надо вставить вместо многоточия? Вводите ответ без пробелов.

```
c := 0;  
for k:=1 to Length(S) do begin  
if S[k] = '#0'; then  
...  
end;
```

Ответ:

19. Какой оператор надо вставить вместо многоточия, чтобы в строке S были записаны символы строки [tt]Q[/] в обратном порядке? Вводите ответ без пробелов.

```
Q := '#123456789';  
S := '#1';  
for k:=2 to 9 do begin  
...  
end;
```

Ответ:

20. Задан массив X[1..N]. Определите число операций сложения, которые выполняются при работе этой программы.

```
S:=X[1]+X[N];  
for k:=1 to N do  
X[k]:=X[k]+X[k]+S;
```

Для обозначения операции умножения используйте символ *.

Ответ:

21. Задан массив $X[1..N]$. Определите число операций умножения, которые выполняются при работе этой программы:

```
S:=X[1]*X[N];  
for k:=1 to N do begin  
  X[k]:=2*X[k]+S;  
  for i:=1 to 3 do  
    S:=S*2;  
  end;
```

Для обозначения операции умножения используйте символ $*$.

Ответ:

22. Задан массив $X[1..N]$. Определите число операций сложения, которые выполняются при работе этой программы:

```
S:=X[1]+X[N]+3;  
for k:=1 to N do  
  for m:=1 to N do  
    X[k]:=X[k]+S;
```

Для обозначения операции умножения используйте символ $*$.

Ответ: