***На правах рукописи***

Минобрнауки России

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Оренбургский государственный университет»**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Методические указания для обучающихся по освоению дисциплины

*«Б1.Д.В.2 Алгоритмы и структуры данных»*

Уровень высшего образования

БАКАЛАВРИАТ

Направление подготовки

*09.03.04 Программная инженерия*

(код и наименование направления подготовки)

*Разработка программно-информационных систем*

(наименование направленности (профиля) образовательной программы)

Квалификация

*Бакалавр*

Форма обучения

*Заочная*

Год набора 2022

Составитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Климачев С.А.

Методические указания рассмотрены и одобрены на заседании кафедры программного обеспечения вычислительной техники и автоматизированных систем

Заведующий кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Соловьев Н.А.

Методические указания является приложением к рабочей программе по дисциплине *Алгоритмы и структуры данных*, зарегистрированной в ЦИТ под учетным номером \_\_\_\_\_\_\_\_

**Содержание**

|  |  |
| --- | --- |
| Введение........................................................................................................ | 4 |
| 1 Общие рекомендации по изучению дисциплины...................................... | 5 |
| 2 Методические указания по лекционным, практическим и лабораторным занятиям............................................................................... | 6 |
| 2.1 Методические указания по лекционным занятиям........................... | 6 |
| 2.2 Методические указания по практическим занятиям ….................... | 6 |
| 2.3 Методические указания по лабораторным занятиям........................ | 7 |
| 3 Методические указания по самостоятельной работе …..………….......... | 8 |
| 3.1 Указания по работе с литературой..................................................... | 8 |
| 3.2 Методические указания по выполнению контрольной работы | 8 |
| 4 Методические указания по изучению разделов дисциплины | 10 |
| 5 Методические указания по промежуточной аттестации по дисциплине | 11 |

**Введение**

Дисциплина *«Алгоритмы и структуры данных»* должна обеспечивать комплексную подготовку будущего специалиста – профессиональную подготовку, развитие творческих способностей, умение формулировать и решать на высоком научном уровне проблемы изучаемой специальности, умение творчески применять и самостоятельно повышать уровень своих знаний.

Основной задачей при преподавании дисциплины является теоретическая и практическая подготовка бакалавра с цельюформирования знаний современной парадигмы программирования структур и алгоритмов обработки данных, способах их реализации, методологических основах современных инструментальных средств разработки программного обеспечения.

Рабочая программа дисциплины предусматривает контактную работу с преподавателем, которая включает в себя лекции, практические занятия, лабораторные работы, консультации и промежуточную аттестацию, а также самостоятельную работу студента, которая включает в себя освоение инструментального аппарата программирования, проработку и повторение лекционного материала и материала учебников и учебных пособий, подготовку к практическим и лабораторным занятиям, подготовку к рубежному контролю и экзамену.

Цель данных методических рекомендаций – обеспечить студенту организацию процесса изучения дисциплины, а также выполнения различных форм самостоятельной работы.

**1 Общие рекомендации по изучению дисциплины**

Перед изучением дисциплины «*Алгоритмы и структуры данных*» студент должен подробно ознакомиться с содержанием рабочей программы дисциплины, с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, с методическими разработками кафедры.

При изучении дисциплины целесообразно руководствоваться следующими общими рекомендациями:

* изучение дисциплины должно вестись систематически и сопровождаться составлением подробного конспекта. В конспект рекомендуется включать все виды учебной работы: лекции, самостоятельную проработку учебника, упражнения, решение задач, лабораторный практикум, ответы на вопросы самопроверки;
* после изучения какого-либо раздела по учебнику или конспекту лекции рекомендуется по памяти записать в тетрадь определения, выводы формул, начертить схемы, графики и ответить на вопросы для самопроверки. Такой метод дает возможность проверить усвоение материала;
* после усвоения теории по одной теме нужно разобрать решения задач, относящихся к этой теме, и самостоятельно решить несколько задач;
* практические занятия направлены на формирование умений выполнения отдельных этапов алгоритмизации и программирования;
* лабораторные занятия, проводимые в компьютерных классах, формируют умения разработки программы на языке высокого уровня;
* следует иметь в виду, что темы программы являются в равной мере важными. Как и в любой другой науке, нельзя приступать к изучению последующих глав, не усвоив предыдущих. Теоретический материал каждой темы имеет существенное практическое назначение.

**2 Методические указания по лекционным, практическим и лабораторным занятиям**

**2.1 Методические указания по лекционным занятиям**

Лекции по дисциплине формируют знания теоретический материал, являющийся базой для восприятия практического умений. Перед очередной лекцией необходимо просмотреть по конспекту материал предыдущей лекции. При затруднениях в восприятии материала следует обратиться к рекомендованным литературным источникам или за консультацией к преподавателю. После прослушивания лекции необходимо прочитать соответствующие темы, уяснить основные термины, проблемные вопросы и подходы к их решению, а также рассмотреть дополнительный материал по теме (в т. ч. практический). Лекционный материал следует использовать при подготовке к практическим и семинарским занятиям.

Краткие записи лекций, их конспектирование помогут усвоить учебный материал. Конспект будет полезным тогда, когда записано самое существенное, основное и сделано это самим студентом. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» принесёт больше вреда, чем пользы. Запись лекций рекомендуется вести по возможности собственными формулировками. Можно запись осуществлять на одной странице, а следующую оставлять для проработки учебного материала самостоятельно в домашних условиях. Конспект лекции лучше подразделять на пункты, параграфы, соблюдая красную строку. Этому в большой степени будут способствовать пункты плана лекции, предложенные преподавателям. Принципиальные места,  
определения и другое следует сопровождать замечаниями  
«важно», «особо важно», «хорошо запомнить» и т.п. Можно делать это и с  
помощью разноцветных маркеров или ручек.

**2.2 Методические указания по практическим занятиям**

Практические занятия по дисциплине способствуют закреплению знаний теоретического материала, освоению компетенций, предусмотренных рабочей программой дисциплины, вырабатывают умения самостоятельной творческой работы, развивают навыки и мыслительные способности. Основой практикума выступают типовые задачи программирования и алгоритмизации, которые должен уметь решать студент, изучающий дисциплину.

Чтобы подготовиться к практическому занятию, необходимо:

1. выполнить предыдущее задание к практическому занятию;
2. внимательно прочитать материал лекции по теме практического занятия, выписать необходимые для себя сведения, правила и т. п.;
3. составить по лекционному материалу алгоритм, с помощью которого будет проще работать на практическом занятии;
4. прочитать материалы учебников (учебных пособий, методических указаний), рекомендуемых к изучаемому разделу, сделать необходимые записи (сведения, которых нет в лекциях).

Решение задач алгоритмизации и программирования данных во время аудиторного занятия, а также при самостоятельной работе лучше производить в специально предназначенной для этого рабочей тетради. При этом рекомендуется придерживаться следующих правил:

1. рисунки, схемы, символы, размерности физических величин выполнять в соответствии с требованиями ГОСТ;
2. в ходе решения задачи не следует изменять первоначально принятые обозначения и т.д. При решении задачи различными методами одна и та же величина должна обозначаться одним и тем же буквенным символом.

На практических занятиях необходимо стремиться к самостоятельному решению задач программирования, находя для этого наиболее эффективные алгоритмы.

**2.3 Методические указания по лабораторным занятиям**

Необходимые для освоения дисциплины «*Алгоритмы и структуры данных*» навыки приобретаются на лабораторных занятиях и требуют серьезной внеаудиторной подготовки к ним.

В ходе подготовки к лабораторным занятиям студент должен:

1. изучить теоретический и практический материал предстоящей лабораторной работы;
2. выполнить все задания, содержащиеся в методических указаниях, которые требуют подготовительной работы;
3. продумать порядок выполнения всех пунктов задания.

Выполнение работы студент осуществляет самостоятельно.

Преподаватель проверяет готовность студента к предстоящей работе. Неподготовленные студенты к работе не допускаются.

Получив допуск к работе, студенты разрабатывают алгоритм решения задачи согласно варианту и, используя интегрированную среду разработки программного обеспечения, выполняют его кодирование.

По окончании работы результат решения задачи предъявляется преподавателю и оформляется отчёт.

В отчёте (электронный формат) приводятся: цель, задачи работы, разработанный алгоритм решения задачи, тестирование программного средства, которые сопровождаются пояснениями, анализом полученных результатов и выводами.

Защита лабораторных работ проводится в часы, отведенные на лабораторные занятия или по графику консультаций преподавателя.

**3 Методические указания по самостоятельной работе**

**3.1 Указания по работе с литературой**

При самостоятельной работе над учебными пособиями рекомендуется придерживаться определенной последовательности. Читая и конспектируя тот или иной раздел учебного пособия, необходимо твердо усвоить основные определения и понятия, а также те закономерности их использования.

**3.2 Методические указания по выполнению курсовой работы**

Учебным планом по направлению 09.03.04 – Программная инженерия предусмотрено проведение курсовой работы. В рамках рабочей программы дисциплины «*Алгоритмы и структуры данных*» для студентов заочной формы обучения предусмотрено выполнение курсовой работы «Использование алгоритмов обработки данных при разработке приложений» с целью обоснования выбора структур данных и алгоритмов их обработки, реализация выбранных алгоритмов обработки данных.

Задание и график выполнения выдается преподавателем в начале семестра. К выполнению курсовой работы предъявляются следующие требования: задание должно выполняться студентом самостоятельно, представляться в установленный срок и соответствовать установленным требованиям по оформлению. Демонстрационные варианты заданий представлены ниже.

Вариант №1

Обычной проблемой анализа текстов являются определение частоты и расположения слов в документе. Эта информация запоминается в конкордансе, где различные слова перечислены в алфавитном порядке и каждое слово снабжено ссылками на строки текста (предложения), в которых оно встречается.

Разработать программу, создающую конкорданс для текстового файла. Вход: открыть документ как текстовый файл и считать текст по словам, отслеживая текущее предолжение. Действие: определить запись, которая состоит из слова, счетчика появлений и списка номеров предложений, содержащих это слово. При первой встрече некоторого слова в тексте создать запись и включить ее в связный список в соответствии с алфавитным порядком. Если слово уже есть в списке, обновить его частоту и список номеров предложений.

Выход: после обработки файла распечатать слова в алфавитном порядке вместе со счетчиками частоты и упорядоченными списками номеров предложений, где встречается каждое слово.

Обеспечить возможность перехода к требуемому предложению, в котором содержится выбранное слово, и визуальному отображению этого слова.

Вариант №2

В 16 этажном доме функционирует лифт. Грузоподъемность лифта ограничена количеством 6-и человек. На каждом этаже могут быть очереди жильцов дома, желающих воспользоваться лифтом. Вызовы лифта образуют очередь вызовов. Кроме того, на каждом этаже потенциально возможны очереди пассажиров. Если нет очередей пассажиров, то и очередь вызовов пуста. Разработать программу, имитирующую работу лифта. Предусмотреть возможность образования очередей случайным образом. Отобразить движение лифта в графическом режиме.

Вариант №3

Разработать программу, выполняющую поиск прохода по лабиринту. Лабиринт задается пользователем в графическом режиме. Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. Программа должна находить проход через лабиринт, двигаясь от заданного входа. После отыскания прохода программа выводит найденный путь в виде координат квадратов. Процесс прохождения лабиринта отобразить графически. Предусмотреть возможность сохранения/открытия проектов лабиринтов.

Вариант №4

Построение кратчайшего маршрута такси от пункта А до пункта Б осуществляется в соответствии с текущей дорожно-транспортной ситуацией и загруженностью дорожной сети, которая может меняться. Участки дороги с высокой загруженностью являются препятствием для движения и должны быть исключены из маршрута. Порог исключения определяется пользователем. Шкала загруженности может варьироваться от 0 (дорога полностью свободна) до 10 (движение со скорость не более 1 м/мин) баллов. Разработать программу, позволяющую строить маршрут в соответствии с указанными условиями.

Вариант №5

Передвижение персонажей игры осуществляется на основе карты лабиринта. При переходе к новой локации карта генерируется случайным образом. Разработать программу, позволяющую генерировать случайные лабиринты.

При выполнении курсовой работы целесообразно руководствоваться предложенными в рабочей программе учебно-методическими материалами. Оформить задание необходимо в соответствии с действующим стандартом ОГУ СТО 02069024.101–2015 «РАБОТЫ СТУДЕНЧЕСКИЕ. Общие требования и правила оформления»

**4 Методические рекомендации по изучению разделов дисциплины**

Раздел 1. Введение в алгоритмы

Определение алгоритма. Свойства алгоритма. Показатели эффективности алгоритмов. Асимптотический анализ сложности алгоритмов. Классы сложности алгоритмов. Анализ рекурсивных алгоритмов.

Раздел 2. Элементарные структуры данных

Типы данных. Абстрактный тип данных. Структуры данных. Классификация структур данных. Способы представления структур данных в памяти ЭВМ. Массив. Табличные структуры данных, классификация. Линейные, логически связанные таблицы. Список, стек, очередь. Очередь с приоритетами.

Раздел 3. Алгоритмы поиска и сортировки данных

Задача поиска. Условия поиска. Поиск в упорядоченных и неупорядоченных таблицах. Линейный поиск. Бинарный поиск. Индексирование. Задача сортировки. Свойства и виды алгоритмов сортировки. Простые методы сортировки (простым обменом, вставкой, выбором). Быстрая сортировка. Сортировка слиянием. Сортировка Шелла. Карманная сортировка. Сортировка подсчетом. Цифровая сортировка. Внешняя сортировка (простым слиянием, естественным слиянием).

Раздел 4 Хеш-таблицы

Таблицы с прямой адресацией. Хеш-таблицы. Понятие и свойства хеш-функции. Методы построения хеш-функций. Коллизия. Методы разрешения коллизий.

Раздел 5. Деревья

Корневое дерево. Бинарное дерево. Представление деревьев в памяти. Обходы бинарного дерева. Бинарное дерево поиска. Оптимальное бинарное дерево поиска. Красно-черное дерево. AVL-дерево. Префиксное дерево. Двоичная куча. Пирамидальная сортировка. Биномиальная куча. Фибоначчиева куча. B-дерево.

Раздел 6. Алгоритмы на графах

Представление графов. Поиск в ширину. Поиск в глубину. Топологическая сортировка. Минимальное остовное дерево. Алгоритм Прима. Алгоритм Крускала. Алгоритм Дейкстры. Алгоритм А\*. Алгоритм Беллмана-Форда. Алгоритм Флойда-Уоршелла. Алгоритм Джонсона. Раскраска графа.

Раздел 7. Алгоритмы на строках

Простейший алгоритм поиска подстрок. Алгоритм Рабина-Карпа. Алгоритм Кнута-Морриса-Пратта. Алгоритм Ахо-Корасик.

Раздел 8. Усовершенствованные методы разработки и анализа

Определение динамического программирования. Элементы динамического программирования. Жадные алгоритмы. Элементы жадной стратегии. Алгоритм Хаффмана. NP-полные задачи.

**5 Методические рекомендации по промежуточной аттестации**

Изучение дисциплины завершается промежуточной аттестацией. Учебным планом по дисциплине «*Алгоритмы и структуры данных*» предусмотрен экзамен.

Подготовка к промежуточной аттестации способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к промежуточной аттестации, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. При подготовке к промежуточной аттестации основное направление дают программа учебной дисциплины и студенческий конспект, которые указывают, что наиболее важно знать и уметь делать. Основной материал должен прорабатываться по учебникам и учебным пособиям, так как конспекта далеко недостаточно для изучения дисциплины. Подготовку по каждому разделу следует заканчивать восстановлением по памяти его краткого содержания в логической последовательности. За один - два дня до промежуточной аттестации назначается консультация. Во время консультации студент имеет полную возможность получить ответ на неясные ему вопросы. А для этого он должен проработать до консультации все темы дисциплины. Кроме того, преподаватель будет отвечать на вопросы других студентов, что будет для вас повторением и закреплением знаний. Кроме того преподаватель на консультации, как правило, обращает внимание на те вопросы, по которым на предыдущих экзаменах ответы были неудовлетворительными, а также фиксирует внимание на наиболее трудных темах дисциплины. Поэтому посещение консультаций обязательно.

Оценка знаний студентов на промежуточной аттестации осуществляется по четырехбальной шкале (таблица 1).

Таблица 1 – Оценивание ответа на экзамене

| 4-балльная шкала | Показатели | Критерии |
| --- | --- | --- |
| Отлично | 1. Полнота изложения теоретического материала;  2. Правильность и/или аргументированность изложения (последовательность действий);  3. Самостоятельность ответа. | Дан полный, в логической последовательности развернутый ответ на поставленный вопрос, где студент демонстрирует знания предмета в полном объеме учебной программы, достаточно глубоко осмысливает дисциплину, самостоятельно, и исчерпывающе отвечает на дополнительные вопросы, приводит собственные примеры по проблематике поставленного вопроса. |
| Хорошо | Дан развернутый ответ на поставленный вопрос, где студент демонстрирует знания, приобретенные на лекционных, практических и лабораторных занятиях, а также полученные посредством изучения обязательных учебных материалов по курсу, дает аргументированные ответы, приводит примеры, в ответе присутствует свободное владение монологической речью, логичность и последовательность ответа. Однако допускается неточность в ответе. |
| Удовлетворительно | Дан ответ, свидетельствующий в основном о знании материала дисциплины, отличающийся недостаточной глубиной и полнотой раскрытия темы, знанием основных вопросов теории, слабо сформированными навыками анализа проблемной области, недостаточным умением давать аргументированные ответы и приводить примеры, недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. |
| Неудовлетворительно | Дан ответ, который содержит ряд серьезных неточностей, обнаруживающий незнание материала изучаемой дисциплины, отличающийся неглубоким раскрытием темы, незнанием основных вопросов теории, несформированными навыками анализа проблемной области, неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Выводы поверхностны. Студент не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя. |