

Государственное бюджетное образовательное учреждение
высшего образования Московской области
«Университет «Дубна»
(государственный университет «Дубна»)

Филиал «Протвино»
Кафедра «Информационные технологии»



Евсиков А.А./
Фамилия И.О.

2019 г.

Рабочая программа дисциплины (модуля)

Структуры и алгоритмы обработки данных

наименование дисциплины (модуля)

Направление подготовки (специальность)

09.03.01 Информатика и вычислительная техника

код и наименование направления подготовки (специальности)

Уровень высшего образования

бакалавриат

бакалавриат, магистратура, специалитет

Направленность (профиль) программы (специализация)

«Программное обеспечение вычислительной техники и автоматизированных систем»


Форма обучения

очная

очная, очно-заочная, заочная

Протвино, 2019

Преподаватель (преподаватели):

Кульман Т.Н., доцент, к.т.н., кафедра информационных технологий 
Фамилия И.О., должность, ученая степень, ученое звание, кафедра; подпись

Рабочая программа разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) высшего образования
09.03.01 Информатика и вычислительная техника
(код и наименование направления подготовки (специальности))

Программа рассмотрена на заседании кафедры информационных технологий
(название кафедры)

Протокол заседания №8 от «23» апреля 2019 г.

Заведующий кафедрой  Нурматова Е.В.
(Фамилия И.О., подпись)

Оглавление

1 Цели и задачи освоения дисциплины (модуля)	4
2 Объекты профессиональной деятельности при изучении дисциплины (модуля)	4
3 Место дисциплины (модуля) в структуре ОПОП	4
4 Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников).....	5
5 Объем дисциплины (модуля) в зачетных единицах с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся	5
6 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических или астрономических часов и виды учебных занятий	7
7 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю) и методические указания для обучающихся по освоению дисциплины (модулю)	11
8 Применяемые образовательные технологии для различных видов учебных занятий и для контроля освоения обучающимися запланированных результатов обучения.....	12
9 Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю).....	12
10 Ресурсное обеспечение	Ошибка! Закладка не определена.
11 Язык преподавания	21

1 Цели и задачи освоения дисциплины (модуля)

Дисциплина посвящена изучению структур данных, применяемых в программировании и современных методов построения и анализа алгоритмов. Можно сказать, что предмет «Структуры и алгоритмы обработки данных» является математическим обоснованием программирования. Весь курс можно разделить на 3 части и в соответствии с этим сформулировать 3 цели:

1. изучение базовых структур данных и освоение технологии их программирования;
2. изучение алгоритмов обработки, связанных со структурами данных (сортировка, поиск и др.); рассмотрение вопросов эффективности алгоритмов;
3. изучение библиотеки шаблонов (STL), – как мощного инструмента, содержащего реализацию многих структур и алгоритмов, повышающего эффективность труда программистов.

Задачами курса являются:

- изложение основных положений теории структур и алгоритмов обработки данных;
- применение теоретических знаний при создании фрагментов информационных систем;
- обучение использованию библиотеки STL при разработке программ, – при этом повышается надёжность программ, их переносимость, а также уменьшаются расходы на их создание.

2 Объекты профессиональной деятельности при изучении дисциплины (модуля)

Объектами профессиональной деятельности в рамках изучаемой дисциплины (модуля) являются:

- программное обеспечение средств вычислительной техники и автоматизированных систем (программы, программные комплексы и системы),
- автоматизированные системы обработки информации и управления.

3 Место дисциплины (модуля) в структуре ОПОП

Дисциплина «Структуры и алгоритмы обработки данных» относится к базовой части блока дисциплин учебного плана (Б1.Б.18.3) и преподаётся два семестра (IV семестр II курса и V семестр III курса).

Приступая к изучению дисциплины «Структуры и алгоритмы обработки данных», студент должен иметь знания и навыки по дисциплинам:

- Программирование на языке высокого уровня,
- Объектно-ориентированное программирование,
- Дискретная математика,
- Математическая логика и теория алгоритмов,

Входящие компетенции: **ОПК-2.**

Данная дисциплина является предшествующей для следующих дисциплин:

- Базы даны.
- Интерфейсы информационных систем,
- Компьютерная графика,
- Проектирование информационных систем,
- Информационные системы и технологии.

4 Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников)

<p>Формируемые компетенции (код компетенции, уровень (этап) освоения) (последний – при наличии в карте компетенции)</p>	<p>Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций</p>
<p><i>ОПК-2: способность осваивать методики использования программных средств для решения практических задач</i></p>	<p><i>Знать</i></p> <ul style="list-style-type: none"> – базовые структуры организации данных и области применения каждой из структур; основные алгоритмы обработки данных, применяемые в программировании – общие принципы взаимодействия алгоритмов и данных в объектно-ориентированной технологии – основные методы анализа сложности и эффективности алгоритмов сортировки и поиска – способы представления стеков, очередей, деревьев, графов в памяти ЭВМ – основные возможности библиотеки шаблонов STL, контейнеры и алгоритмы. <p><i>Уметь</i></p> <ul style="list-style-type: none"> – при решении конкретной задачи свести её к формулированию задачи программирования – применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы – выбирать оптимальную для данной задачи программирования структуру данных; – уметь работать с библиотекой STL, знать её основные классы и алгоритмы – уметь программировать такие структуры данных, как списки, стеки, очереди, бинарные деревья, множества и др. – выбрать алгоритм сортировки или поиска в соответствии с оценкой эффективности алгоритма. <p><i>Владеть</i></p> <ul style="list-style-type: none"> – навыками использования сложных структур данных и программной реализации алгоритмов обработки данных для решения задач программирования – навыками программной реализации алгоритмов обработки данных на языке высокого уровня; – навыками отладки и тестирования написанной программы; – навыками использованию библиотеки шаблонов при разработке программ для повышения надёжности, переносимости и сокращения времени на разработку.

Результат обучения сформулирован с учетом следующих профессиональных стандартов: «Программист» №4 (приказ Министерства труда и социальной защиты РФ от 18 декабря 2013 г. № 679н)

5 Объем дисциплины (модуля) в зачетных единицах с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Объем дисциплины (модуля) составляет 7 зачетных единицы, всего 252 часа. Дисциплина читается 2 семестра (IV и V):

119 часа составляет контактная работа обучающегося с преподавателем:

133 часов составляет самостоятельная работа обучающегося.

IV семестр:

34 часов – лекционные занятия;

34 часа – практические занятия.

40 часов – самостоятельная работа

Вид промежуточного контроля – зачёт (проводится в рамках занятий семинарского типа)

V семестр:

17 часов – лекционные занятия;

34 часа – практические занятия.

93 часов – самостоятельная работа

Вид промежуточного контроля – зачёт с оценкой (проводится в рамках занятий семинарского типа).

6 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических или астрономических часов и виды учебных занятий

Лекции представлены в виде презентаций.

Наименование и краткое содержание разделов и тем дисциплины (модуля) Форма промежуточной аттестации по дисциплине (модулю)	Всего (часы)	В том числе:										
		Контактная работа (работа во взаимодействии с преподавателем), часы из них							Самостоятельная работа обучающегося, часы, из них			
		Лекционные занятия	Семинарские занятия	Практические занятия	Лабораторные занятия	..	Групповые консультации	Индивидуальные консультации	Учебные занятия, направленные на проведение текущего контроля успеваемости (коллоквиумы, практические контрольные занятия и др.)	Всего	Выполнение домашних заданий, подготовка к к/р	Подготовка творческих заданий.
IV семестр												
Раздел 1. Указатели для формирования структур данных Назначение указателей. Объявление указателей. Операторы работы с указателями. Автоматические и динамические переменные. Использование указателей для формирования динамических структур. Понятие структур данных и их классификация.		2		4					2	8	4	4
Раздел 2. Абстрактные типы данных (АТД). Линейные структуры Определение АТД: тип и структура данных, операторы. Классификация АТД. Линейные структуры: стек, очередь, дек. Операции над линейными списками. Реализация операций с использованием массивов и с помощью указателей. Стеки и функции. Передача параметров в функцию: по значению, по ссылке.		2		4				1	7	2		2
Раздел 3. Связные списки Определение связного списка. Доступ к элементам списка. Работа со связанными списками: 1. Вывод на экран содержимого списка, 2. Удаление указанного узла, 3. Вставка узла в указанную позицию.		2		2					4	4		4

Операторы "." и ">". Двусвязные списки.													
Раздел 4. Текстовые файлы и потоки ввода/вывода Внутреннее представление файлов. Потоки ввода/вывода. Состояние программы после подключения и отключения потоков. Символьный ввод/вывод. Операторы ввода/вывода ">>" и "<<".		2		2						4	4		4
Раздел 5. Деревья Рекурсивное определение структуры дерева. Основные понятия и определения. Бинарные деревья. Обход дерева: прямой, обратный, симметричный. Способы представления бинарного дерева (в виде массива, с помощью указателей). Бинарные деревья поиска, идеально сбалансированные деревья и сбалансированные деревья (АВЛ-деревья). Деревья Фибоначчи.		3		4				1		8	4		4
Раздел 6. Восстановление сбалансированных деревьев. Б-деревья Балансировка узлов. Правило поворотов. RR-поворот (LL-поворот), RL-поворот. Включение и исключение в/из АВЛ-деревьев. Б-деревья, определение, свойства, основные операции. Работа с Б-деревьями.		3						1		4	2		2
Раздел 7. Оценка алгоритмов. Сортировки Сложность алгоритма, обозначение O-большое. Определение порядка алгоритма. Сравнение сложности различных алгоритмов. Внутренние сортировки (выбором, вставками, пузырьком). Рассмотрение алгоритмов и анализ сложности. Быстрая сортировка и анализ сложности.		2		2						4	2		2
Раздел 8. Сортировка слиянием. Сортировка Шелла Разбор алгоритмов сортировки слиянием и сортировки Шелла. Оценки эффективности.		2		2				2		6	2		2
Раздел 9. Графы, алгоритмы работы с графами Ориентированные и неориентированные графы. Определение, основные понятия и свойства. Представления графов. Поиск в глубину, поиск в ширину.		2						1		3	4		4
Раздел 10. Топологическая сортировка. Алгоритмы поиска кратчайшего пути в графах Отношение частичного порядка. Топологическая сортировка. Кратчайшие пути в графе. 1. Поиск кратчайшего расстояния от одной из вершин графа до всех остальных (алгоритм Дейкстры, алгоритм Беллмана – Форда). 2. Поиск кратчайших путей между всеми парами вершин графа (алгоритм Флойда-Уоршелла). 3. Переборные алгоритмы (поиск в глубину, поиск в		2						1		3	2		2

ширину).													
Раздел 11. Алгоритмы поиска. Хеширование Последовательный, бинарный поиск. Понятие хеширования. Хеш-таблицы. Разрешение коллизий. Функции хеширования. Открытое и закрытое хеширование.		2						0,5	2,5	1		1	
Раздел 12. Поиск в тексте. Методы разработки алгоритмов Постановка задачи. Изучение алгоритмов: 1. прямого поиска; 2. Кнута, Морриса и Пратта; 3. Боуера и Мура.		2						0,5	2,5	2		2	
Раздел 13. Рекурсия в алгоритмах и структурах данных Основные понятия, примеры. Рекурсивные алгоритмы. Рекурсия в структурах данных (списки, деревья, графы). Фракталы и их использование.		2		2					4	2		2	
Раздел 14. Кодирование и сжатие данных Основные понятия. Обратимые и необратимые методы. Методы сжатия без потери информации (алгоритм Хаффмана, алгоритм Лемпеля-Зива-Велча). Сжатие данных с потерями.		2							2	1		1	
Раздел 15. Обзор первой части курса		4						2	6	4		4	
Всего:		34		22				12	68	40		40	
V семестр													
Раздел 1. Стандартная библиотека шаблонов в языке программирования C++ Шаблоны функций и классов. Шаблоны и перегрузка функций. Пространства имен. Последовательные контейнеры и итераторы. Свойства последовательных контейнеров.		2		2				1	5	7		7	
Раздел 2. Операции с последовательными контейнерами Различные функции (<i>push_back, pop_back, insert, erase u др.</i>). Алгоритмы (<i>sort, find, copy</i>).		2		2				1	5	8		8	
Раздел 3. Операция объединения <i>merge</i>. Категории итераторов Алгоритмы (<i>merge</i>). Типы, определенные пользователем. Применение <i>merge</i> для структур данных. Категории итераторов и операции, применимые к каждой категории. Алгоритмы <i>replace u reverse</i> .		2		4				2	8	8		8	
Раздел 4. Функциональные объекты. Работа с последовательностями Понятие предиката. Функциональные объекты и их		2		2				1	5	8		8	

использование. Работа с последовательностями. Алгоритмы <i>accumulate</i> , <i>count</i> , <i>find_if</i> и др.														
Раздел 5. Ассоциативные контейнеры Типы ассоциативных контейнеров. Примеры работы со словарями. Алгоритмы работы с ассоциативными контейнерами. Программа «Формирование частотного словаря». Битовые множества.		3		4					2		9	8	15	23
Раздел 6. Адаптеры. Работа со словарём Стек, очередь, очередь с приоритетами. Пары и сравнения. Пример работы со словарём «Телефонный справочник». Основные понятия об отношениях между классами.		2		4					2		8	8	15	23
Раздел 7. NP-полные задачи и труднорешаемые задачи. Методы разработки алгоритмов Основные определения. Классификация задач. Методы разработки алгоритмов (метод декомпозиции, поиск с возвратом, динамическое программирование, метод ветвей и границ и др.). Рассмотрение задач: коммивояжера, Ханойская башня, обход конём шахматной доски, о ранце)		2		2					1		5	7		7
Раздел 8. Обзор курса		2		2					2		6	9		9
Всего:		17		22					12		51	63	30	93
Промежуточная аттестация <u>зачёт</u> (балльно-рейтинговая система)														
Итого		51									119			133

7 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю) и методические указания для обучающихся по освоению дисциплины (модулю)

Методические указания к практическим занятиям

В учебном процессе используются активные и интерактивные формы: обсуждение отдельных разделов дисциплины, опросы на занятиях, совместное и самостоятельное решение студентами практических задач и заданий, разбор конкретных заданий. Выполнение домашних заданий (как общих, так и индивидуальных) закрепляет знания, полученные студентами на практических занятиях.

По каждому разделу студентам на учебном сервере в электронном виде даются:

- теоретические материалы,
- методические указания по решению задач,
- вопросы к работе на практических занятиях,
- готовые проекты решения различных задач.

Методические разработки по разделам (*IV семестр*):

1. Указатели и их применение для решения задач с массивами и со структурами данных (предоставляются готовые проекты).
2. Линейные структуры (стеки) с готовыми проектами.
3. Различные виды сортировок с решением задач.
4. Связные списки с решением задач.
5. Программирование очереди и бинарного дерева с решением задач.
6. Перегрузка функций с готовым проектом.
7. Проекты по созданию приложений в диалоговом режиме.
8. Разработанное приложение «Учёт заявок на авиабилеты».
9. Разработанное приложение «Обход конём шахматной доски» (перебор с возвратом).
10. Вопросы к коллоквиумам по темам:
 - Линейные структуры данных
 - Деревья
 - Графы
 - В-деревья, хеш-функции
 - Алгоритмы сортировки. Эффективность алгоритмов
 - Алгоритмы поиска, включая поиск в тексте
 - Алгоритмы на графах. Рекурсия
 - Кодирование и сжатие данных.

Методические разработки по разделам (*V семестр*):

1. Шаблоны функций и классов с готовыми проектами.
2. Изучение последовательных контейнеров (функции, итераторы, алгоритмы) с решением задач и с заданиями для самостоятельного решения.
3. Изучение ассоциативных контейнеров (функции, итераторы, алгоритмы) с решением задач и с заданиями для самостоятельного решения.
4. Готовые проекты «Частотный словарь» и «Телефонный справочник»
5. Разработанное приложение «Студент» с реализацией различных функций.
6. Разработанное приложение «Морской бой».

Эти материалы позволяют студентам написать по выбранной теме своё приложение в качестве творческой работы.

Методическое обеспечение инновационных форм учебных занятий

Все материалы для практических занятий предоставляются в электронном виде. Готовые творческие работы пересылаются преподавателю через Интернет. Готовые проекты (в Visual Studio) выкладываются на сервер для общего обсуждения и дальнейших самостоятельных доработок.

Методические указания для самостоятельной работы обучающихся и прочее

Для самостоятельной работы студентов (домашние работы, подготовка к контрольным работам и творческие работы) выдаются задания различных уровней сложности, решения аналогичных задач, готовые проекты и решения.

8 Применяемые образовательные технологии для различных видов учебных занятий и для контроля освоения обучающимися запланированных результатов обучения

Перечень обязательных видов учебной работы студента:

- посещение лекций;
- ответы на теоретические вопросы на практических занятиях;
- решение практических задач и заданий на практических занятиях;
- выполнение контрольных работ;
- выполнение домашних работ (выдаются индивидуально);
- выполнение творческих заданий.

Инновационные формы проведения учебных занятий

Семестр	Вид учебных занятий	Используемые инновационные формы проведения учебных занятий	Количество академ. часов
IV, V семестры	Практические занятия	1. Разбор конкретных задач и заданий для выработки навыков при выполнении домашних заданий по всем темам, выполнение самостоятельных работ в соответствии с возможностями студентов (индивидуальные задания). 2. Выполнение контрольных и творческих заданий стимулирует поиск и нахождение самостоятельных решений, нацелены на выработку профессиональных умений и навыков.	15
Всего:			15

9 Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю)

- Основная компетенция ОПК-2: способностью осваивать методики использования программных средств для решения практических задач.
-
- Описание шкал оценивания.

II курс, IV семестр (зачёт)

По итогам работы в семестре студент может получить максимально 100 баллов. Итоговой формой контроля в IV семестре является зачет.

В течение IV семестра студент может заработать баллы за следующие виды работ:

№	Вид работы	Сумма баллов
1	Работа на практических занятиях	34
2	Контрольные работы (ПР-2.1, ПР-2.2, ПР-2.3, ПР-2.4)	49
3	Аудиторные занятия (посещение)	17
	Итого:	100

Если к моменту окончания семестра студент набирает 70 баллов, то он получает оценку «зачтено» автоматически. Если студент не набрал минимального числа баллов (70 баллов), то он в обязательном порядке должен сдавать зачет.

Текущий контроль успеваемости осуществляется в процессе выполнения практических и самостоятельных работ в соответствии с ниже приведенным графиком.

График выполнения самостоятельных работ студентами в II семестре

Виды работ	Недели учебного процесса																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ПР-2.1	ВЗ		ЗЗ														
ПР-2.2				ВЗ			ЗЗ										
ПР-2.3								ВЗ			ЗЗ						
ПР-2.4												ВЗ			ЗЗ		

ПР-2 – контрольная работа (домашняя)

ВЗ – выдача задания

ЗЗ – защита задания

III курс, V семестр (зачёт с оценкой)

По итогам работы в семестре студент может получить максимально 100 баллов. Итоговой формой контроля в V семестре является зачет с оценкой.

В течение V семестра студент может заработать баллы за следующие виды работ:

№	Вид работы	Сумма баллов
1	Работа на практических занятиях	34
2	Контрольные работы (ПР-2.1)	9
3	Контрольные работы (ПР-2.2)	10
4	Творческое домашнее задание (ПР-2.3)	30
5	Аудиторные занятия (посещение)	17
	Итого:	100

По результатам полученных баллов выставляются оценки.

Общая сумма баллов за семестр	Итоговая оценка
86-100	Отлично
71-85	Хорошо
51-70	Удовлетворительно
0-50	Неудовлетворительно

Студент может получить автоматическую оценку «удовлетворительно», «хорошо», «отлично» и зачёт с оценкой не сдавать. При желании повысить свою оценку студент имеет право отказаться от автоматической оценки и сдавать зачёт с оценкой.

В обязательной порядке сдается зачёт с оценкой, если по результатам семестра набрано менее 51 балла.

Текущий контроль успеваемости осуществляется в процессе выполнения практических и самостоятельных работ в соответствии с ниже приведенным графиком.

График выполнения самостоятельных работ студентами в V семестре:

Виды работ	Недели учебного процесса																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ПР-2.1		ВЗ				ЗЗ											
ПР-2.2							ВЗ				ЗЗ						
ПР-2.3										ВЗ					ЗЗ		

ПР-2 – контрольная работа (домашняя)

ВЗ – выдача задания

ЗЗ – защита задания

- Критерии и процедуры оценивания результатов обучения по дисциплине (модулю), характеризующих этапы формирования компетенций

ОПК-2: способностью осваивать методики использования программных средств для решения практических задач.

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	Уровень освоения компетенции	КРИТЕРИИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю) ШКАЛА оценивания <i>(критерии берутся из соответствующих карт компетенций, шкала оценивания (4 или более шагов) устанавливается в зависимости от того, какая система оценивания (традиционная или балльно-рейтинговая) применяется)</i>					ПРОЦЕДУРЫ ОЦЕНИВАНИЯ
		1	2	3	4	5	
<p>31 (ОПК-2)</p> <p>Знать: основные теоретические вопросы, связанные с этапами решения задач на компьютере, с понятием жизненного цикла программы, с критериями качества программ.</p>	I - пороговый	Отсутствие знаний	Не знает основные теоретические вопросы	Удовлетворительно знает основные теоретические вопросы	Хорошо знает основные теоретические вопросы. Допускает отдельные негрубые ошибки	Хорошо знает основные теоретические вопросы. Не допускает ошибок	Устное собеседование
<p>VI (ОПК-2)</p> <p>Уметь: выражать записанные алгоритмы через линейные, циклические и условные операторы на языке программирования.</p>	I - пороговый	Отсутствие умений	Демонстрирует частичное умение использовать знания при написании программ. Допускает множественные грубые ошибки	Демонстрирует удовлетворительное умение использовать знания при написании программ. Допускает достаточно серьезные ошибки.	Демонстрирует достаточно устойчивое умение использовать знания при написании программ. Допускает отдельные негрубые ошибки	Демонстрирует устойчивое умение использовать знания при написании программ. Не допускает ошибок.	Выполнение практических домашних заданий
<p>VI (ОПК-2)</p> <p>Владеть: практическими навыками программирования задач с массивами, строками, базовыми математическими алгоритмами</p>	I - пороговый	Отсутствие владения	Не владеет или демонстрирует низкий уровень владения навыками программирования указанных типов задач	Демонстрирует удовлетворительный уровень владения навыками программирования указанных типов задач	Демонстрирует хороший уровень владения навыками программирования указанных типов задач	Демонстрирует высокий уровень владения навыками программирования указанных типов задач	Выполнение контрольных творческих заданий

–Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций и (или) для итогового контроля сформированности компетенции.

II курс, IV семестр (зачёт)

Вопросы к зачёту по дисциплине «Структуры и алгоритмы обработки данных»

Структуры данных

1. Понятие абстрактного типа данных (АТД).
2. Линейные структуры данных. Области применения.
3. Абстрактный список. Определение, свойства, операции.
4. Реализация списков в виде массива и на основе указателей.
5. Работа со связанными списками: вывод на экран, удаление и вставка узла.
6. Абстрактный стек. Определение, свойства, операции.
7. Реализация стека с помощью массива, односвязного списка.
8. Абстрактная очередь. Определение, свойства, операции.
9. Нелинейные структуры данных. Области применения.
10. Деревья. Рекурсивное определение. Основные понятия.
11. Области применения деревьев.
12. Обход дерева: прямой, обратный, симметричный.
13. Операторы работы с деревьями.
14. Абстрактное бинарное дерево. Определение, свойства, операции.
15. Полное, совершенное и сбалансированное бинарные деревья.
16. AVL — деревья.
17. Реализация бинарного дерева в виде массива и с помощью указателей.
18. Двоичное дерево поиска.
19. Ориентированный граф. Основные понятия и терминология. Операторы работы с орографом.
20. Реализация ориентированного графа с помощью матрицы смежности и списков смежности.
21. Неориентированный граф. Основные понятия и терминология. Неориентированный граф и операторы работы с ним.
22. Реализация неориентированного графа с помощью матрицы смежности и списков смежности.
23. Графы как абстрактные типы данных.
24. Области применения графов.
25. Кратчайшие пути в графе. Различные алгоритмы.
26. Задача о «Кенигсбергских мостах».
27. В-деревья. Основные понятия, определения и примеры использования.
28. Таблицы с прямым доступом и хэш – таблицы.
29. Понятие хеш-функции, требования к хеш-функции, примеры таких функций. Понятие коллизии.
30. Разрешение коллизий с помощью метода цепочек.
31. Методы расстановки: деление, умножение, универсальное хеширование.
32. Основные понятия сжатия данных Обратимые и необратимые методы.

Алгоритмы

33. Простой алгоритм внутренней сортировки (вставками).
34. Простой алгоритм внутренней сортировки (выбором).
35. Простой алгоритм внутренней сортировки (пузырьковая).
36. Усовершенствованные алгоритмы внутренней сортировки (быстрая сортировка).
37. Алгоритм внешней сортировки (сортировка слиянием).
38. Алгоритм внешней сортировки (сортировка Шелла).
39. Последовательный поиск.
40. Бинарный поиск.
41. Поиск по вторичным ключам. Организация инвертированных индексов.
42. Использование деревьев в задачах поиска.
43. Поиск подстрок в тексте (Прямой поиск).
44. Поиск подстрок в тексте (Алгоритм Боуера, Мура, Хорспула).
45. Поиск подстрок в тексте (Алгоритм Кнута, Мориса, Пратта).
46. Алгоритмы на графах (поиск в глубину).

47. Алгоритмы на графах (поиск в ширину).
48. Алгоритмы на графах (топологическая сортировка).
49. Кратчайшие пути в графе. Взвешенные графы.
50. Алгоритм Дейкстры.
51. Алгоритм Беллмана-Форда.
52. Алгоритм Флойда-Уоршелла.
53. Метод сжатия без потерь (алгоритм Хаффмана).
54. Метод сжатия без потерь (алгоритм Лемпеля-Зива-Велча).
55. Рекурсия. Выполнение рекурсивного вызова, использование стека.
56. Рекурсия с возвратом. Решение задачи об обходе конём шахматной доски.
57. Понятие сложности и эффективность алгоритмов обработки структур данных.
58. Эффективность алгоритмов сортировки и поиска.

Пример варианта контрольной работы по алгоритмам сортировки

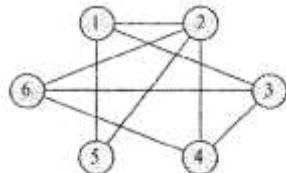
Вариант 3

1. Выполните трассировку алгоритма сортировки слиянием, если нужно упорядочить в порядке возрастания следующий массив: 5 71 24 35 1 48.
 2. Выполните трассировку алгоритма быстрой сортировки, если нужно упорядочить в порядке убывания следующий массив: 5 71 24 18 35 1 48.
 3. Выполните трассировку алгоритма сортировки вставками, если нужно упорядочить в порядке убывания следующий массив: 23 30 20 80 40 15 60.
 4. Выполните трассировку алгоритма сортировки Шелла, если нужно упорядочить в порядке возрастания следующий массив: 5 71 24 18 35 1 48 33.
 5. Выполните трассировку алгоритма сортировки методом выбора, если нужно упорядочить в порядке возрастания следующий массив: 31 25 6 13 78 17 72.
 6. Напишите, какой порядок имеет сложность алгоритмов в п.1 – п.5, используйте $O(f(n))$.
- И т.д.

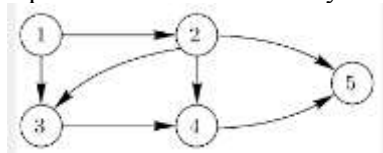
Пример варианта контрольной работы по темам «Деревья и графы»

Вариант 2

1. Деревья. Рекурсивное определение.
2. Прямой обход дерева. Пример.
3. Определение АВЛ — деревьев.
4. Представить алгебраическое выражение в виде бинарного дерева и обойти его, используя обратный обход:
 $(A + B) * (C + D) - E$
5. Определение ориентированного графа. Пример ОГ.
6. Представить неориентированный граф в виде списка смежности:



7. Для какой цели в алгоритме поиска в ширину используется очередь?
8. Дать определение пути в графе.
9. Применить топологическую сортировку к графу:

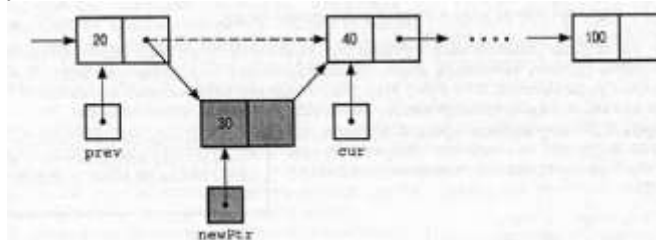


Пример варианта итоговой контрольной работы

Вариант 2

1. Определение линейного списка.

2. Неориентированный граф. Основные понятия и терминология.
 3. Сбалансированное дерево, определение, примеры.
 4. Представить выражение в виде дерева: $(a/b * k - x) * (y + z * c)$.
 5. Обойти полученное в п.4 дерево, методом «слева направо».
 6. Отсортировать последовательность чисел 9, 25, 31, 64, 8, 1, 105, 17 в порядке возрастания методом «пузырька».
- Показать трассировку процесса сортировки.
7. Очередь и операции над ней.
 8. Эффективность алгоритмов сортировки и поиска. Привести 5 оценок различных алгоритмов.
 9. Определение дерева поиска.
 10. Приведите 3 алгоритма, используемых при построении хеш-функций.
 11. В чем преимущества и недостатки списков по сравнению с массивами?
 12. Краткое описание алгоритма на графах: поиск в глубину.
 13. Опишите с помощью средств C++ узел списка.
 14. На рисунке показана схема добавления нового узла. Напишите на C++ операторы вставки этого узла.



III курс, V семестр (зачёт с оценкой)

Вопросы к зачёту (с оценкой) по дисциплине «Структуры и алгоритмы обработки данных»

Стандартная библиотека шаблонов STL

1. Общие положения STL.
2. Понятие шаблона (шаблонные функции, шаблонные классы).
3. Пространства имен.
4. Основные составляющие STL (контейнеры, алгоритмы, итераторы).
5. Последовательные контейнеры (векторы, списки, двусторонние очереди).
6. Методы работы с последовательными контейнерами.
7. Алгоритм сортировки в STL.
8. Алгоритм поиска в STL.
9. Алгоритм копирования и итератор вставки.
10. Алгоритм слияния в STL.
11. Категории итераторов.
12. Алгоритмы replace и reverse в STL.
13. Функциональные объекты.
14. Алгоритм accumulate в STL.
15. Алгоритм count в STL.
16. Функциональные объекты, определенные в STL.
17. Ассоциативные контейнеры в STL.
18. Методы и свойства ассоциативных контейнеров.
19. Множества и словари.
20. Множества и словари с дубликатами.
21. Понятие пары в STL.
22. Приемы работы со словарем.
23. Адаптеры контейнеров.
24. Понятие задач класса P.
25. NP-полная задача. Примеры.

Пример варианта контрольной работы на последовательные контейнеры

Вариант 6

1. Создать очередь. Ввести переменное количество ненулевых целых (ввод завершается нулем) и вывести их на экран.
2. Используя введенные в п.1 данные, применить функции для вставки и удаления для выполнения следующих действий (после каждого действия следует вывод на экран):
 - a. Вставить 5 и 8 в начало.
 - b. Перед последним элементом вставить 33.
 - c. Удалить последний элемент.
 - d. Удалить 4 - элемент.
 - e. Отсортировать.
3. Создать вектор, внести в него 8 любых (отсортированных по возрастанию) чисел. Вывести их на экран.
4. Выполнить слияние элементов вектора и очереди в любой контейнер.

Пример варианта контрольной работы на ассоциативные контейнеры

Во всех вариантах меню должно выглядеть следующим образом:

1. Найти заданное значение.
2. Вывести полный список.
3. Внести новое значение.

Вариант 1

1. Создать словарь Country, в котором ключом является название страны, а значением — название столицы этой страны.
2. Заполнить его несколькими значениями (4-5).
3. Создать меню, в котором будут решаться следующие задачи:
 - a. Ввести название страны и по ней найти и вывести название столицы.
 - b. Вывести на экран полный список стран и их столиц.
 - c. Ввести название страны, если такая страна есть, то вывести её столицу, если нет — ввести в словарь название страны и её столицы. После этого вывести на экран пополненный список стран и их столиц.

Примеры творческих заданий

Вариант 2

Написать программу, отыскивающую проход по лабиринту, с использованием контейнерного класса `stack` из `STL`.

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. Программа находит проход через лабиринт, двигаясь от заданного входа. После отыскания прохода программа выводит найденный путь в виде координат квадратов.

Вариант 8

Написать программу учета заявок на обмен квартир и поиска вариантов обмена. Каждая заявка содержит сведения о двух квартирах: требуемой (искомой) и имеющейся. Сведения о каждой квартире содержат: количество комнат, площадь, этаж, район.

Программа должна обеспечивать выбор с помощью меню и выполнение одной из следующих функций:

- ввод заявки на обмен;
- поиск в картотеке подходящего варианта: при совпадении требований и предложений по количеству комнат и этажности и различии по показателю «площадь» в пределах 10% выводится соответствующая карточка и удаляется из списка, в противном случае поступившая заявка включается в картотеку;
- вывод всей картотеки.

Для хранения данных картотеки использовать контейнерный класс `list`.

10 Ресурсное обеспечение

• Перечень основной и дополнительной учебной литературы

Основная учебная литература

1. Колдаев В.Д. Структуры и алгоритмы обработки данных [Электронный ресурс]: Учебное пособие / В.Д. Колдаев. - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2014. - 296 с.: (Высшее образование: Бакалавриат). ISBN 978-5-369-01264-2 // ЭБС "Znaniium.com". - URL: <http://znaniium.com/catalog/product/418290> (дата обращения: 17.04.2019). - Режим доступа: ограниченный по логину и паролю
2. Белов В. В. Алгоритмы и структуры данных [Электронный ресурс]: Учебник / Белов В.В., Чистякова В.И. - М.:КУРС, НИЦ ИНФРА-М, 2017. - 240 с.: . - (Бакалавриат) ISBN 978-5-906818-25-6 // ЭБС "Znaniium.com". - URL: <http://znaniium.com/catalog/product/766771> (дата обращения: 17.04.2019). - Режим доступа: ограниченный по логину и паролю
3. Парфенов Д В. Язык Си: кратко и ясно: Учебное пособие [Электронный ресурс] / Д.В. Парфенов. - М.: Альфа-М: НИЦ ИНФРА-М, 2014. - 320 с.: ISBN 978-5-98281-397-8. // ЭБС "Znaniium.com". - URL: <http://znaniium.com/catalog/product/459254> (дата обращения: 17.04.2019). - Режим доступа: ограниченный по логину и паролю

Дополнительная учебная литература

1. Григорьев А. А. Методы и алгоритмы обработки данных : учеб. пособие [Электронный ресурс] / А.А. Григорьев. — М. : ИНФРА-М, 2018. — 256 с. — (Высшее образование: Бакалавриат) 978-5-16-011916-8 // ЭБС "Znaniium.com". - URL: <http://znaniium.com/catalog/product/922736> (дата обращения: 17.04.2019). - Режим доступа: ограниченный по логину и паролю
2. Культин Н. С/С ++ в задачах и примерах. - СПб.: БХВ-Петербург, 2009. - 288 с.: ил.;
3. Мартин, Р. Чистый код. Создание, анализ и рефакторинг : Библиотека программиста / Р. Мартин. - СПб : Питер, 2016. - 464с. : ил. - ISBN 978-5-496-00487-9.

• Периодические издания

1. Информационные технологии и вычислительные системы / Учредитель Федеральное государственное учреждение "Федеральный исследовательский центр "Информатика и управление" РАН; гл. ред. С.В. Емельянов, - М.: Федеральное государственное учреждение "Федеральный исследовательский центр "Информатика и управление" РАН". Год основания 1995 г. Полные электронные версии статей журнала доступны на сайте

научной электронной библиотеки «eLIBRARY.RU»:

<https://elibrary.ru/contents.asp?titleid=8746>

- Информация и безопасность / учредители: ФГБОУ Воронежский государственный технический университет; гл. ред. А.Г. Остапенко. – Воронеж.: Воронежский государственный технический университет. Журнал основан в 1998 году. Полные электронные версии статей журнала доступны на сайте научной электронной библиотеки «eLIBRARY.RU»: <http://elibrary.ru/contents.asp?titleid=8748>
- Открытые системы. СУБД / учредитель и издатель: ООО «Издательство «Открытые системы»; гл. ред. Д. Волков. – М.: Издательство «Открытые системы». Журнал основан в 1999 году. Сайт журнала <http://www.osp.ru/os/> Полные электронные версии статей журнала доступны на сайте научной электронной библиотеки «eLIBRARY.RU»: <https://elibrary.ru/contents.asp?titleid=9826>
- Программные продукты и системы / учредители: МНИИПУ (г.Москва), гл. редакция международного журнала «Проблемы теории и практики управления» (г. Москва), ЗАО НИИ «Центрпрограммсистем» (г. Тверь); гл. ред. С.В. Емельянов. – Тверь.: НИИ «Центрпрограммсистем». Журнал основан в 1995 году. Полные электронные версии статей журнала доступны на сайте научной электронной библиотеки «eLIBRARY.RU»: <https://elibrary.ru/contents.asp?titleid=9834>; Сайт журнала www.swsys.ru
- Системный администратор / учредитель и издатель: Общество с ограниченной ответственностью "Издательский дом "Положевец и партнеры" гл. ред. Г. Положевец. – М.: Общество с ограниченной ответственностью "Издательский дом "Положевец и партнеры" Журнал основан в 2002 году. Полные электронные версии статей журнала доступны на сайте научной электронной библиотеки «eLIBRARY.RU»: https://elibrary.ru/title_about.asp?id=9973

• **Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

Электронно-библиотечные системы и базы данных

- ЭБС «Znanium.com»: <http://znanium.com/>
- ЭБС «Лань»: <https://e.lanbook.com/>
- ЭБС «Юрайт»: <https://biblio-online.ru/>
- ЭБС «Университетская библиотека онлайн»: <http://biblioclub.ru/>
- Научная электронная библиотека (РУНЭБ) «eLIBRARY.RU»: <http://elibrary.ru>
- Национальная электронная библиотека (НЭБ): <http://нэб.рф/>
- Базы данных российских журналов компании «East View»: <https://dlib.eastview.com/>

Научные поисковые системы

- ArXiv.org - научно-поисковая система, специализируется в областях: компьютерных наук, астрофизики, физики, математики, квантовой биологии. <http://arxiv.org/>
- Google Scholar - поисковая система по научной литературе. Включает статьи крупных научных издательств, архивы препринтов, публикации на сайтах университетов, научных обществ и других научных организаций. <https://scholar.google.ru/>
- WorldWideScience.org - глобальная научная поисковая система, которая осуществляет поиск информации по национальным и международным научным базам данных и порталам. <http://worldwidescience.org/>
- SciGuide - навигатор по зарубежным научным электронным ресурсам открытого доступа. <http://www.prometeus.nsc.ru/sciguide/page0601.ssi>

Профессиональные ресурсы сети «Интернет»

- Федеральная информационная система «Единое окно доступа к информационным ресурсам»: <http://window.edu.ru/>.

2. Проект Инициативного Народного Фронта Образования - ИНФО-проект. Школа программирования Coding Craft <http://codingcraft.ru/>.
3. Портал Life-prog <http://life-prog.ru/>.
4. OpenNet www.opennet.ru.
5. Алгоритмы, методы, программы algotlist.manual.ru.
6. Сервер министерства высшего образования www.informika.ru.

– **Перечень информационных технологий, используемых при осуществлении образовательного процесса, включая программное обеспечение, информационные справочные системы (при необходимости)**

Проведение практических занятий по дисциплине предполагается использование специализированных аудиторий, оснащенных персональными компьютерами, объединенными в локальную сеть и имеющих доступ к ресурсам глобальной сети Интернет.

Студентам предоставляются электронные материалы, подготовленные преподавателем.

Для выполнения заданий самостоятельной подготовки обучающиеся обеспечиваются литературой, а также в определенном порядке могут получать доступ к информационным ресурсам Интернета.

Microsoft Visual Studio (Программы для ЭВМ DreamSpark Premium Electronic Software Delivery Renewal (MS Imagine Premium, договор Tr000104809/м18 от 01.09.2016 г.)). Дисциплина обеспечена необходимым программным обеспечением, которое находится в свободном доступе (программы Open office, свободная лицензия, код доступа не требуется).

– **Описание материально-технической базы**

Компьютерный класс (15 ПК): оборудование в собственности

11 Язык преподавания

Русский