

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ  
Н.Г. ЧЕРНЫШЕВСКОГО»  
Факультет компьютерных наук и информационных технологий

УТВЕРЖДАЮ  
Декан факультета



"13" 09 2021 г.

**Рабочая программа дисциплины**  
**СТРУКТУРЫ ДАННЫХ И АЛГОРИТМЫ**

Направление подготовки бакалавриата  
44.03.01 – Педагогическое образование

Профиль подготовки бакалавриата  
Информатика

Квалификация (степень) выпускника  
Бакалавр

Форма обучения  
Очная

Саратов,  
2022

Статус	ФИО	Подпись	Дата
Преподаватель-разработчик	Огнева М.В. Грецова А.П.		13.09.2021
Председатель НМК	Кондратова Ю.Н.		13.09.2021
Заведующий кафедрой	Огнева М.В.		13.09.2021
Специалист Учебного управления			

## 1. Цели освоения дисциплины

Целями освоения дисциплины "Структуры данных и алгоритмы" являются:

- **освоение и систематизация знаний**, относящихся к абстрактным структурам данных и алгоритмам в математике и информатике, а также способов применения их на практике;
- **изучение** абстрактных структур данных и алгоритмов, которые используются для решения практических задач получения, хранения, обработки и передачи информации, а также способов их программной реализации;
- **овладение умениями** самостоятельно выбирать структуры данных и проектировать алгоритмы обработки данных, проводить их анализ и предлагать эффективную программную реализацию;
- **развитие** способностей применять полученные знания и умения в профессиональной деятельности.

## 2. Место дисциплины в структуре ООП

Данная учебная дисциплина относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (Модули)» ООП и направлена на формирование у обучающихся универсальных и профессиональных компетенций.

Для изучения дисциплины необходимы компетенции, сформированные в результате изучения дисциплины «Программирование».

Компетенции, сформированные при изучении данной дисциплины, используются при изучении дисциплин: «Языки программирования в школьном курсе информатики», «Языки программирования высокого уровня», а также при выполнении курсовых и выпускной квалификационной работы.

## 3. Результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора (индикаторов) достижения компетенции	Результаты обучения
<b>УК-1</b> Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	<b>1.1_Б.УК-1.</b> Анализирует задачу, выделяя ее базовые составляющие. Осуществляет декомпозицию задачи. <b>2.1_Б.УК-1.</b> Находит и критически анализирует информацию, необходимую для	Знать: структуры данных, базовые конструкции для построения алгоритмов, математические методы анализа алгоритмов. Уметь: анализировать алгоритмы и структуры данных,

	<p>решения поставленной задачи.</p> <p><b>3.1_ Б.УК-1.</b> Рассматривает различные варианты решения задачи, оценивая их достоинства и недостатки.</p>	<p>необходимые для решения поставленной задачи.</p> <p>Владеть: навыками проведения анализа существующих структуры данных и алгоритмов, оценки их достоинств и недостатков.</p>
<p><b>ПК-7.</b> Способен использовать математический аппарат, методы программирования и современные информационно-коммуникационные технологии для решения практических задач получения, хранения, обработки и передачи информации</p>	<p><b>ПК-7.1.</b> Решает практические задачи получения, хранения, обработки и передачи информации.</p> <p><b>ПК-7.2.</b> Использует математический аппарат, методы программирования и современные информационно-коммуникационные технологии для решения учебных задач</p>	<p>Знать: алгоритмы обработки информации</p> <p>Уметь: использовать структуры данных и алгоритмы для обработки информации</p> <p>Владеть: навыками использования структур данных и алгоритмов для решения учебных задач.</p>

#### 4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 8 зачетных единиц 288 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости и (по неделям семестра)  Формы промежуточ ной аттестации (по семестрам)
				лекц ии	Практические занятия	СРС	
					Общая трудоемко	Из них – практиче	

					сть	ская подготов ка		
1	Основы алгоритмизации и программирования на языке C++	3	1-4	8	4	2	12	Опрос по контрольным вопросам
2	Типы данных, определяемые пользователем, сложные типы данных	3	5-9	10	14	4	16	Опрос по контрольным вопросам
3	Понятие подпрограммы: описание и использование; шаблоны функций	3	10-14	10	10	2	10	Опрос по контрольным вопросам
4	Анализ алгоритмов	3	15-16	4	4	2	6	Опрос по контрольным вопросам, контрольная работа на 15-16 неделе
3	Промежуточная аттестация							Экзамен
	ИТОГО			<b>32</b>	<b>32</b>	<b>10</b>	<b>44</b>	<b>36</b>
4	Объектно-ориентированное программирование на C++.	4	1-2	4	4	2	12	Опрос по контрольным вопросам
5	Алгоритмы сортировки	4	3-8	12	12	2	12	Опрос по контрольным вопросам, доклады
6	Алгоритмы STL.	4	9-10	4	4	2	12	Опрос по контрольным вопросам
7	Абстрактный тип данных "список" и алгоритмы с ними	4	11-15	10	10	4	12	Опрос по контрольным вопросам, контрольная работа на 14-15 неделе, защита курсовой работы
	Промежуточная аттестация							Экзамен
	ИТОГО			<b>30</b>	<b>30</b>	<b>10</b>	<b>48</b>	<b>36</b>
	ВСЕГО			<b>62</b>	<b>62</b>	<b>20</b>	<b>92</b>	<b>72</b>

*Основы алгоритмизации и программирования на языке C++.* Состав языка. Структура программы на C++. Стандартные типы данных. Консольный ввод-вывод данных. Операции и операторы языка. Описания величин: область действия и область видимости, класс памяти. Операторы языка C++, операторы простые, составные и структурные. Реализация базовых алгоритмов обработки данных на языке C++.

*Типы данных, определяемые пользователем, сложные типы данных.* Указатели и работа с указателями. Массивы – одномерные и двумерные, статические и динамические, работа с массивами. Строковый тип данных в C и C++, стандартные функции для работы со строками в C и C++, строка, как массив символов, особенности работы. Ввод и вывод строковых величин, работа со строками. Структурный тип данных в C и C++, описание величин строкового типа и работа с полями данными. Файлы последовательного и прямого доступа к данным, текстовые файлы, функции для работы с файлами, файловый ввод и вывод.

*Понятие подпрограммы, функции и процедуры: описание и использование; шаблоны функций.* Необходимость и достоинства использования подпрограмм. Средства определения подпрограмм, правила передачи параметров, параметры формальные и фактические, входные и выходные, локальные и глобальные. Подпрограммы в C и C++, функции возвращающие значения и не возвращающие значения, их назначение и особенности их использования. Параметры массивы и параметры функции. Перегрузка функций и шаблоны функций. Описание и определение шаблонов функций, их назначение и правила их использования.

*Анализ алгоритмов.* Время выполнения программ. Асимптотические соотношения. Правила анализа программ. Анализ и сравнение алгоритмов, основные характеристики. Понятие временной сложности или времени выполнения. Степень роста функции, асимптотические соотношения. Основные правила анализа программ.

*Объектно-ориентированное программирование на C++.* Объектно-ориентированное программирование: инкапсуляция, наследование и полиморфизм. Понятие объекта, его состояние и поведение, классы, определение класса и объявление класса. Статические, дружественные и виртуальные поля и методы, особенности их использования. Абстрактные классы, их назначение и использование. Понятие области видимости: общие, личные, защищенные и опубликованные поля и методы объекта. Перегрузка методов. Конструкторы и деструкторы, их назначение и использование

*Алгоритмы сортировки.* Понятие сортировки. Простые алгоритмы сортировки: вставка, обмен, выбор. Реализация и оценка. Быстрая сортировка, сортировка деревом: реализация, вычисление времени выполнения. Сортировки за линейное время. Анализ и сравнительная характеристика алгоритмов.

*Алгоритмы STL.* Класс-контейнер vector. Итераторы. Алгоритмы поиска, замены, удаления в векторе.

*Абстрактный тип данных "список" и алгоритмы с ним.* Понятие абстрактного типа данных. Линейные списки как абстрактный тип данных. Частные случаи линейного списка: стек и очередь. Реализация стека, очереди с помощью массива. Связная реализация. Основные операции. Реализация линейного списка общего вида: односвязная, двухсвязная реализация. Решение прикладных задач с использованием списков.

### План лабораторных занятий

На лабораторных занятиях студенты выполняют индивидуальные задания (приведены в ФОС) в соответствии со следующей таблицей. Задания выполняются на компьютере. Каждая программа должна быть протестирована на 3-5 примерах.

№ занятия	Тема	Практикум
<b>3 семестр</b>		
1-4	Основы алгоритмизации и программирования на языке C++	Практикум «Базовые элементы языка C++» Практикум «Операторы C++» Практикум «Рекуррентные соотношения» Практикум «Вычисление конечных и бесконечных сумм и произведений»
5-9	Типы данных, определяемые пользователем, сложные типы данных	Практикум «Массивы» Практикум «Строки» Практикум «Организация файлового ввода/вывода данных» Практикум «Структуры»
10-14	Понятие подпрограммы: описание и использование; рекурсивные функции, шаблоны функций	Практикум «Подпрограммы»
15-16	Анализ алгоритмов	Практикум «Анализ и сравнение алгоритмов»
<b>4 семестр</b>		
1-2	Объектно-ориентированное программирование на C++.	Практикум «Классы и объекты»
3-8	Алгоритмы сортировки	Практикум «Сортировки»
9-10	Алгоритмы STL.	Практикум «Класс-контейнер vector»
11-15	Абстрактный тип данных "список" и алгоритмы с ними	Практикум «Абстрактный тип данных "список"»

## **5. Образовательные технологии, применяемые при освоении дисциплины**

Предусматривается широкое использование в учебном процессе активных и интерактивных форм проведения занятий: организация временных творческих коллективов при разборе алгоритмов, при тестировании задач, организация дискуссий и обсуждений спорных вопросов, использование метода мозгового штурма, организация конкурсов и тренингов, деловой игры.

В рамках курса предусмотрены встречи с представителями российских и зарубежных компаний – разработчиков программного обеспечения (в рамках договора с факультетом КНиИТ).

Использование активных и интерактивных форм проведения занятий:

1. Лекции и семинары на тему «Основы алгоритмизации и программирования на языке C++. Отличительные черты от языка Pascal» – дискуссия, метод мозгового штурма.
2. Лекции и семинары на тему «Алгоритмы работы с массивами» - деловая игра.
3. Лекции и семинары на тему «Анализ алгоритмов» – дискуссия.
4. Лекции и семинары на тему «Алгоритмы сортировки» – моделирование пошагового выполнения алгоритмов сортировок.
5. Итоговая лекция на тему «Абстрактный тип данных "список"» – дискуссия.
6. Все лабораторные занятия, кроме первого – организация временных творческих коллективов при тестировании задач.
7. Итоговые (по темам) – лабораторные занятия – конкурс программ.

В рамках практической подготовки по данной дисциплине используются кейс-задания, выполнение которых направлено на формирование таких профессиональных действий как решение практических задач получения, хранения, обработки и передачи информации, с использованием математического аппарата, методов программирования и современных информационно-коммуникационных технологий.

Примеры кейс-заданий/проектных заданий приведены в фондах оценочных средств.

При обучении лиц с ограниченными возможностями и инвалидов используются подходы, способствующие созданию безбарьерной образовательной среды: технологии дифференциации и индивидуализации обучения, сопровождение тьюторами в образовательном пространстве; увеличивается время на самостоятельное освоение материала. Используется сочетание разных форм и способов передачи учебной информации: вербальный, невербальный, с использованием средств визуализации информации (презентации) и разных способов отчетности (письменно, устно, с использованием электронных дистанционных технологий).

Удельный вес занятий, проводимых в интерактивных формах, определяется главной целью (миссией) программы, особенностью контингента обучающихся и содержанием конкретных дисциплин, и в целом в учебном процессе они должны составлять не менее 70% аудиторных занятий. Занятия лекционного типа для соответствующих групп студентов не могут составлять более 50% аудиторных занятий.

#### **6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.**

Самостоятельная работа студентов заключается в углубленном изучении материала курса с использованием литературы, выполнении заданий для самостоятельной работы, подготовке к выполнению заданий лабораторных практикумов, к опросам, контрольным работам и экзамену.

Для текущего контроля успеваемости проводятся опросы, формируемые с использованием контрольных вопросов.

Контрольная работа выполняется с целью контроля уровня подготовки студентов по дисциплине. Контрольные работы оформляются в виде письменного отчета, в котором приводится подробное решение задач.

Фонд оценочных средств дисциплины включает в себя варианты контрольных работ, контрольные вопросы, вопросы для проведения экзамена, темы курсовых работ.

#### **Задания для контрольных работ**

##### ***Контрольная работа 1.***

Вариант 1.

1. Получить все трехзначные натуральные числа, в записи которых хотя бы две цифры одинаковые. Использовать подпрограмму получения цифр некоторого числа  $N$ .

2. По заданной квадратной матрице размером  $n \times n$  построить вектор длиной  $(2n-1)$ , элементы которого являются максимумами элементов диагоналей, параллельных главной диагонали. Использовать подпрограмму нахождения максимального элемента в последовательности из  $n$  чисел.

Вариант 4.

1. Используя подпрограмму вычисления суммы цифр некоторого числа  $N$ , получить все трехзначные натуральные числа, сумма цифр которых равна  $m$ .

2. Для заданной целочисленной матрицы размером  $n \times n$  найти максимум сумм модулей элементов диагоналей, параллельных главной диагонали. Использовать подпрограмму вычисления суммы  $m$  элементов



## **Контрольная работа 2.**

### **Вариант 1.**

1. Во входном файле записаны фамилии. Создать из них список. Удалить всех однофамильцев из исходного списка, записав их в новый список.

Задачу выполнить с помощью:

- а) стека
- б) списка общего вида.

Структуры реализовать в виде:

- а) массива
- б) односвязного списка

2. Оценить время выполнения программ из первого пункта, провести сравнительную характеристику.

3. Отсортировать список из первого пункта методом выбора.

### **Вариант 2.**

1. Во входном файле записаны фамилии. Создать из них список. Удалить всех однофамильцев из исходного списка, записав их в новый список.

Задачу выполнить с помощью:

- а) очереди
- б) списка общего вида. Структуры реализовать в виде:

- а) массива
- б) двухсвязного списка

2. Оценить время выполнения программ из первого пункта, провести сравнительную характеристику.

3. Отсортировать список из первого пункта методом вставки.

## **Темы докладов (проектов)**

Доклады выполняются студентами индивидуально. Проект выполняется в группе из 2-3 человек и предусматривает исследовательскую составляющую. Т.е. при выполнении проекта необходимо реализовать несколько алгоритмов и сравнить их между собой по разным признакам (в зависимости от темы).

1. Сравнительная характеристика простых схем сортировки (эмпирически).

2. Сравнительная характеристика быстрой и пирамидальной сортировки в среднем (эмпирически).

3. Сравнительная характеристика выполнения быстрой сортировки в зависимости от выбора опорного элемента.

4. Сортировка слиянием.

5. Усовершенствование простых сортировок.

6. Построение синтаксического анализатора математических выражения с использованием стека.

7. Как убрать рекурсию, используя стек.

8. Очереди с приоритетами.
9. Структура "дек" и операции над ней.

**Вопросы для проведения промежуточной аттестации (экзамен,  
3 семестр)**

1. Классификация языков программирования, элементы языка программирования C++: алфавит, слова, константы.
2. Переменные и операции в C++, выражения, порядок вычисления выражений.
3. Базовые типы данных в C++, преобразование типов, стандартные функции.
4. Структура программы на C++, пример программы.
5. Типы операторов в C++, оператор пустой, составной, оператор-выражения, оператор возврата.
6. Оператор условной передачи управления в C++, примеры.
7. Оператор - переключатель в C++, примеры.
8. Оператор цикла с параметром в C++, его использование, примеры.
9. Операторы цикла с предусловием и постусловием в C++, особенности использования.
10. Оператор безусловного перехода, операторы продолжения и завершения, примеры использования.
11. Организация консольного ввода-вывода данных. Управляющие последовательности, используемые при выводе. Манипуляторы вывода вещественных типов данных.
12. Одномерные массивы (статические и динамические). Объявление массива, ввод, вывод массива. Алгоритмы поиска максимума и минимума в массиве. Освобождение памяти, выделенной под динамический массив.
13. Двумерные массивы (статические и динамические). Объявление массива, ввод, вывод массива в виде матрицы. Алгоритмы поиска максимума и минимума в массиве. Освобождение памяти, выделенной под динамический массив.
14. Указатели. Операция взятия адреса и операция разадресации. Операции new и delete. Ссылки. Использование ссылок в качестве параметров функции.
15. Работа со строками в стиле языка C. Объявление и инициализация строковой переменной, ввод, вывод. Основные функции по работе со строками в стиле C. Функции, применимые для символов
16. Работа со строками – объектами класса string. Объявление и инициализация строковой переменной, ввод, вывод. Операции со строками. Основные функции по работе со строками класса string. Функции, применимые для символов
17. Подпрограммы в C++, функции возвращающие и не возвращающие значение, примеры. Параметры, передаваемые по значению и по

адресу.

18. Использование подпрограмм, параметры формальные, локальные, глобальные, обращения к подпрограммам, фактические параметры.
19. Работа с текстовыми файлами. Открытие входных и выходных потоков, связывание потока с физическим файлом. Считывание и запись в файл чисел, строк. Закрытие потока.
20. Синтаксис описания структуры. Члены-данные и члены-функции. Закрытые члены структуры. Обращение к членам структуры в программе. Пример описания структуры и массива структур.
21. Время выполнения программ, проблема выбора алгоритма. Примеры.
22. Асимптотические соотношения. Примеры.
23. Сравнение скорости роста. Примеры.
24. Вычисление времени выполнения. Правило сумм, правило произведений. Некоторые правила анализа программ. Примеры.

#### **Вопросы для проведения промежуточной аттестации (экзамен, 4 семестр)**

1. Сортировка, основные понятия. Обзор основных методов.
2. Простые схемы сортировки: метод “пузырька”: временная сложность и подсчет перестановок. Сравнительная характеристика. Примеры.
3. Простые схемы сортировки: метод выбора: временная сложность и подсчет перестановок. Сравнительная характеристика. Примеры.
4. Простые схемы сортировки: метод вставки: временная сложность и подсчет перестановок. Сравнительная характеристика. Примеры.
5. Быстрая сортировка. Временная сложность быстрой сортировки. Пример.
6. Пирамидальная сортировка. Время выполнения пирамидальной сортировки. Пример.
7. Время выполнения алгоритмов сортировок «сравнениями». Дерево решений.
8. Сортировки за линейное время. Сортировка подсчетом, поразрядная сортировка.
9. Класс-контейнер `vector`. Варианты объявления объекта типа `vector` (пустого или инициализированного с помощью конструктора). Функция добавления элемента в конец вектора. Размер вектора. Функция `empty`. Функции вставки элемента в вектор и удаления.
10. Итератор класса `vector`. Операции с итераторами (\*, ++, +, -). Пример вывода элементов вектора в цикле по итераторам. Использование итераторов в алгоритмах STL (примеры).
11. Понятие линейного списка, операции с линейными списками. Виды списков. Примеры.
12. Стек. Представление стека с помощью массива, с помощью указателей. Сравнительная характеристика этих представлений. Операции со стеком.

13. Очередь. Представление очереди с помощью массива, с помощью указателей. Сравнительная характеристика этих представлений. Операции с очередью.

14. Представление списка общего вида в виде односвязной реализации. Реализация основных операций.

15. Представление списка общего вида в виде двусвязной реализации. Реализация основных операций.

16. Сравнительная характеристика разных реализаций АДТ "список".

17. Стеки и рекурсивные процедуры. Исключение рекурсий на примере задачи о рюкзаке.

18. Технологии программирования, концепции, заложенные в ООП.

19. Основные понятия ООП: абстракция, инкапсуляция, полиморфизм.

20. Понятие объекта, его состояние и поведение, классы, определение класса и объявление класса.

21. Статические, дружественные и виртуальные поля и методы, особенности их использования.

22. Абстрактные классы, их назначение и использование.

23. Понятие области видимости: общие, личные, защищенные и опубликованные поля и методы объекта.

24. Указатель this и перегрузка методов.

25. Использование классов, различные способы инициализации.

26. Использование классов, работа с массивами и указателями на объекты.

27. Наследование, пример использования наследования.

28. Конструкторы и деструкторы, их назначение и использование.

### **Контрольные вопросы**

1. Сформулируйте понятия синтаксиса и семантика языка.  
2. Назовите основные лексические элементы языка  
3. Что представляет собой переменная? Назовите способы ее описания.

4. Перечислите основные типы данных языков программирования. Их представление в памяти.

5. Что такое указатели?

6. Что такое массив? Назовите варианты представления массивов в памяти ЭВМ

7. Строки. Их представление в памяти ЭВМ.

8. Способы преобразование типов данных.

9. Как осуществляется ввод-вывод данных в языке C++?

10. Что такое алгоритм? Способы записи алгоритма.

11. Базовые управляющие структуры: цепочки, ветвления, циклы.

12. Конструкции передачи управления.

13. Принцип модульности программы. Понятие подпрограммы.

14. Объявление и определение функций в языке C++.

15. В каких случаях используют функции возвращающие значение, а в каких не возвращающие значение?

16. Сформулируйте основные принципы объектно-ориентированного программирования.
17. Что такое инкапсуляция, наследование, полиморфизм?
18. Классы и объекты в C++.
19. Поля и методы классов в C++.
20. Что такое конструктор?
21. Какие характеристики используют при выборе алгоритма?
22. Что называют временной сложностью (временем выполнения) алгоритма?
23. Отличается ли время выполнения алгоритма и время выполнения программы на основе этого алгоритма?
24. В каких случаях необходимо выбирать алгоритмы с меньшей степенью роста времени выполнения?
25. Сформулируйте определения верхней границы, нижней границы, асимптотически точной оценки.
26. Сформулируйте лемму сумм. Для чего ее используют?
27. Сформулируйте базовые правила для вычисления времени выполнения.
28. Что такое сортировка?
29. Как можно провести классификацию алгоритмов сортировки?
30. Какие сортировки мы называем "простыми" и почему? Какое у них время выполнения?
31. Сформулируйте алгоритм сортировки "пузырьком".
32. Сформулируйте алгоритм сортировки вставками.
33. Сформулируйте алгоритм сортировки выбором.
34. Какие сортировки мы называем "быстрыми"? Почему? Какое у них время выполнения?
35. Сформулируйте принцип выполнения быстрой сортировки.
36. Что такое опорный элемент в быстрой сортировке? Как его выбирать? Что от этого зависит?
37. Что такое частично упорядоченное дерево (пирамида) или двоичная куча? Каковы ее свойства?
38. Сформулируйте алгоритм пирамидальной сортировки.
39. За какое время выполняется пирамидальная сортировка?
40. Можно ли придумать сортировку на основе сравнений, которая будет выполняться за линейное время? Почему?
41. Сформулируйте алгоритм сортировки подсчетом.
42. Сформулируйте алгоритм поразрядной сортировки.
43. Какие недостатки у рассмотренных нами сортировок за линейное время?
44. Что такое абстрактный тип данных?
45. Что такое линейный список?
46. Что такое стек? Где используются стеки?
47. Что такое очередь? Где используется очередь?
48. Как можно реализовать список?

49. Расскажите о реализации списка с помощью массива.
50. Расскажите о реализации списка с помощью связной структуры.
51. Проведите сравнительную характеристику реализаций списка.

#### **Темы курсовых работ**

1. Проблема выбора алгоритма на примере задач сортировок.
2. Объектно-ориентированное программирование в Паскаль ABC.Net.
3. Построение стекового калькулятора.
4. Пути в графах.
5. Комбинаторный поиск.
6. Метод "разделяй и властвуй" на примере алгоритмов сортировок.
7. Структура данных "словарь", ее реализация и основные алгоритмы.
8. Объектно-ориентированная реализация списка на основе массива и на основе связной структуры.
9. Двоичный поиск и связанные с ним алгоритмы.
10. Задача поиска медианы.
11. Структура данных "множество", ее реализация и основные алгоритмы.
12. Эвристические методы перебора.
13. Динамическое программирование.
14. Алгоритм обхода в глубину и его применение для решения прикладных задач.
15. Алгоритм обхода в ширину и его применение для решения прикладных задач.

Тему курсовой работы студент получает индивидуально в начале 3 семестра. Работа выполняется в течении двух семестров. Защита курсовой проходит на кафедре на 16-й неделе 4 семестра. По итогам выполнения и защиты студент получает зачет по курсовой работе.

### **7. Данные для учета успеваемости студентов в БАРС**

Таблица 1.1 Таблица максимальных баллов по видам учебной деятельности.

1	2	3	4	5	6	7	8	9
Семестр	Лекции	Лабораторные занятия	Практические занятия	Самостоятельная работа	Автоматизированное тестирование	Другие виды учебной деятельности	Промежуточная аттестация	Итого
3	5	40	0	10	0	5	40	100
4	5	40	0	10	0	5	40	100

Итого	10	80	0	20	0	10	80	200
-------	----	----	---	----	---	----	----	-----

## Программа оценивания учебной деятельности студента

### 3-й семестр

#### *Лекции*

Посещаемость, опрос, активность за один семестр – от 0 до 5 баллов.

#### *Лабораторные занятия*

1. Контроль выполнения лабораторных заданий в течение одного семестра - от 0 до 25 баллов.

2. Контрольная работа - от 0 до 15 баллов.

Таким образом, за лабораторные занятия студент может получить 40 баллов.

#### *Практические занятия*

Не предусмотрены.

#### *Самостоятельная работа*

Выполнение домашних заданий - от 0 до 10 баллов.

#### *Автоматизированное тестирование*

Не предусмотрено

#### *Дополнительно*

Изучение факультативного материала по дополнительным главам дисциплины, успешное выступление на лекционном или лабораторном занятии с презентацией и докладом по теме, одобренной преподавателем, своевременность выполнения текущих и дополнительных заданий - от 0 до 5 баллов

#### ***Промежуточная аттестация***

**36-40 баллов** – ответ на «отлично»

**26-35 баллов** – ответ на «хорошо»

**16-25 баллов** – ответ на «удовлетворительно»

**0-15 баллов** – неудовлетворительный ответ.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 3-й семестр по дисциплине «Структуры данных и алгоритмы» составляет 100 баллов.

Таблица 2.2 Таблица пересчета полученной студентом суммы баллов по дисциплине «Структуры данных и алгоритмы» в оценку (экзамен)

меньше 60 баллов	«неудовлетворительно»
------------------	-----------------------

от 60 до 80 баллов	«удовлетворительно»
от 81 до 90 баллов	«хорошо»
более 90 баллов	«отлично»

#### **4-й семестр**

##### *Лекции*

Посещаемость, опрос, активность за один семестр – от 0 до 5 баллов.

##### *Лабораторные занятия*

1. Контроль выполнения лабораторных заданий в течение одного семестра - от 0 до 25 баллов.

2. Контрольная работа - от 0 до 15 баллов.

Таким образом, за лабораторные занятия студент может получить 40 баллов.

##### *Практические занятия*

Не предусмотрены.

##### *Самостоятельная работа*

Выполнение домашних заданий - от 0 до 10 баллов.

##### *Автоматизированное тестирование*

Не предусмотрено

##### *Дополнительно*

Изучение факультативного материала по дополнительным главам дисциплины, успешное выступление на лекционном или лабораторном занятии с презентацией и докладом по теме, одобренной преподавателем, своевременность выполнения текущих и дополнительных заданий - от 0 до 5 баллов

##### ***Промежуточная аттестация***

**36-40 баллов** – ответ на «отлично»

**26-35 баллов** – ответ на «хорошо»

**16-25 баллов** – ответ на «удовлетворительно»

**0-15 баллов** – неудовлетворительный ответ.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 4-й семестр по дисциплине «Структуры данных и алгоритмы» составляет 100 баллов.

Таблица 2.2 Таблица пересчета полученной студентом суммы баллов по дисциплине «Структуры данных и алгоритмы» в оценку (экзамен)

меньше 60 баллов	«неудовлетворительно»
от 60 до 80 баллов	«удовлетворительно»
от 81 до 90 баллов	«хорошо»
более 90 баллов	«отлично»



## 8. Учебно-методическое и информационное обеспечение дисциплины «Структуры данных и алгоритмы»

а) литература:

1. Вирт, Никлаус Алгоритмы и структуры данных [Электронный ресурс]/ Никлаус Вирт ; перевод Ф. В. Ткачева. — 2-е изд. — Саратов : Профобразование, 2019. — 272 с. — URL: <http://www.iprbookshop.ru/88753.html> (дата обращения: 05.02.2020). ✓


2. Самуйлов, С. В. Алгоритмы и структуры обработки данных [Электронный ресурс]: учебное пособие / С. В. Самуйлов. — Саратов : Вузовское образование, 2016. — 132 с. URL: <http://www.iprbookshop.ru/47275.html> (дата обращения: 05.02.2020) ✓

3. Павловская, Т. А. Программирование на языке высокого уровня Паскаль [Электронный ресурс]/ Т. А. Павловская. — 2-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 153 с. — URL: <http://www.iprbookshop.ru/73714.html> (дата обращения: 05.02.2020). ✓

4. Апанасевич, С.А. Структуры и алгоритмы обработки данных. Линейные структуры [Электронный ресурс]: учебное пособие / С.А. Апанасевич. — Санкт-Петербург : Лань, 2019. — 136 с. URL: <https://e.lanbook.com/book/113934> (дата обращения: 05.02.2020). ✓

б) программное обеспечение и Интернет-ресурсы:

1. Образовательный сайт [course.sgu.ru](http://course.sgu.ru).

2. Портал обучения информатике и программированию [school.sgu.ru](http://school.sgu.ru). 

## **9. Материально-техническое обеспечение дисциплины (модуля)**

Для проведения лекционных занятий необходима аудитория с мультимедийным оборудованием и доской.

Для проведения лабораторных занятий необходим компьютерный класс, в котором установлен Microsoft visual studio.

Реализация практической подготовки в рамках учебных занятий запланирована на базе кафедры «Информационных систем и технологий в обучении» и «Учебного центра информационно-коммуникационных технологий в обучении».

Программа составлена в соответствии с требованиями ФГОС ВО с учетом Примерной ООП ВО по направлению и профилю подготовки 44.03.01 «Педагогическое образование» профиль «Информатика».

Автор

Зав.кафедрой ИиП

доцент, к.ф.-м.н

М.В. Огнева

доцент, к.п.н

А.П. Грецова

Программа одобрена на заседании кафедры информатики и программирования от 13 мая 2019 года, протокол № 17.

Программа актуализирована в 2021 году (одобрена на заседании кафедры информатики и программирования от 13 сентября 2021 года, протокол № 2