

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
ПиИТ

наименование кафедры, отвечающей за реализацию дисциплины



проф. Махортов С.Д.,
подпись, расшифровка подписи

05.03.2024

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.56.10 Алгоритмы и структуры данных

Код и наименование дисциплины в соответствии с учебным планом

1. Код и наименование направления подготовки/специальности:

10.05.01 Компьютерная безопасность

2. Профиль подготовки/специализация: Разработка защищенного программного обеспечения

3. Квалификация (степень) выпускника: специалист

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины: ПиИТ

6. Составители программы: Вахтин А.А., к.ф.-м.н., доц.
(ФИО, ученая степень, ученое звание)

7. Рекомендована: НМС ФКН протокол № 5 от 05.03.2024

(наименование рекомендующей структуры, дата, номер протокола,

отметки о продлении вносятся вручную)

8. Учебный год: 2024/2025

Семестр(ы): 2

9. Цели и задачи учебной дисциплины:

Целью учебной дисциплины является знакомство студентов с различными способами представления данных в программе, с различными классами задач и типами алгоритмов, встречающихся при решении задач на современных ЭВМ. Изучение структур данных и алгоритмов их обработки, знакомство с фундаментальными принципами построения эффективных и надежных программ.

Задачей учебной дисциплины является становление математика-программиста, повышение культуры мышления. Овладение компьютерными методами обработки информации путем развития профессиональных навыков разработки, выбора и преобразования алгоритмов, что является важной составляющей эффективной реализации программного продукта.

10. Место учебной дисциплины в структуре ООП:

Учебная дисциплина относится к блоку Б1, обязательная часть.

Специальные требования к входным знаниям, умениям и компетенциям студента не предусматриваются. Данный курс расширяет и углубляет знания студентов, полученные в ходе изучения дисциплины «Математический анализ».

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ОПК-7	Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ОПК-7.7	знает базовые структуры данных	Знать: базовые классы, реализующие структуры в языке C#, динамические структуры данных (стеки, очереди), деревья, графы. Уметь: применять базовые классы, реализующие структуры на языке C#, динамические структуры данных (стеки, очереди), деревья графы. Владеть (иметь навык(и)): Базовыми классами, реализующими структуры на языке C#, динамические структуры данных (стеки, очереди), деревья графы.
		ОПК-7.8	знает основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы	Знать: основные алгоритмы сортировки и поиска данных, рекурсивные алгоритмы, алгоритмы с возвратом, применяемые в комбинаторных алгоритмах и алгоритмах на графах. Уметь: реализовать алгоритмы сортировки и поиска данных, рекурсивные алгоритмы, алгоритмы с возвратом, применяемые в комбинаторных алгоритмах и алгоритмах на графах. Владеть (иметь навык(и)): алгоритмами сортировки и поиска данных, рекурсивные алгоритмы и алгоритмы с возвратом, применяемые в комбинаторных алгоритмах и алгоритмах на графах.
ОПК-13	Способен разрабатывать компоненты	ОПК-13.10	знает базовые структуры данных	Знать: базовые классы, реализующие структуры в языке C#, динамические структуры данных (стеки, очереди),

<p>программных и программно-аппаратных средств защиты информации в компьютерных системах и проводить анализ их безопасности</p>			<p>деревья, графы.</p> <p>Уметь: разрабатывать компоненты для программ, с использованием базовых классов, реализующие динамические структуры данных (стеки, очереди), деревья графы.</p> <p>Владеть (иметь навык(и)): навыками разработки компонент для программ, реализующие динамические структуры данных (стеки, очереди), деревья графы.</p>
	ОПК-13.11	<p>знает основные комбинаторные и теоретико-графовые алгоритмы, а также способы их эффективной реализации и оценки вычислительной сложности;</p>	<p>Знать: основные алгоритмы сортировки и поиска данных, рекурсивные алгоритмы, алгоритмы с возвратом, применяемые в комбинаторных алгоритмах и алгоритмах на графах.</p> <p>Уметь: реализовать компоненты для программ с использованием алгоритмов сортировки и поиска данных, рекурсивных алгоритмов, алгоритмов с возвратом, применяемых в комбинаторных алгоритмах и алгоритмах на графах.</p> <p>Владеть (иметь навык(и)): навыками разработки компонент для программ с использованием алгоритмов сортировки и поиска данных, рекурсивных алгоритмов и алгоритмов с возвратом, применяемых в комбинаторных алгоритмах и алгоритмах на графах.</p>
	ОПК-13.12	<p>умеет формализовать поставленную задачу</p>	<p>Знать: основы современной технологии разработки программ.</p> <p>Уметь: формализовать поставленную задачу для реализации алгоритмов и программ.</p> <p>Владеть (иметь навык(и)): формализации поставленной задачи для разработки алгоритмов и программ.</p>
	ОПК-13.13	<p>умеет разрабатывать эффективные алгоритмы и программы</p>	<p>Знать: основы современной технологии разработки программ.</p> <p>Уметь: разрабатывать алгоритмы для поставленной задачи.</p> <p>Владеть (иметь навык(и)): разработки алгоритмов для поставленной задачи.</p>
	ОПК-13.14	<p>умеет проводить оценку вычислительной сложности алгоритма</p>	<p>Знать: методику оценки вычислительной сложности алгоритма.</p> <p>Уметь: проводить оценку вычислительной сложности алгоритма.</p> <p>Владеть (иметь навык(и)): оценки вычислительной сложности</p>

				алгоритма.
		ОПК-13.15	умеет планировать разработку сложного программного обеспечения	Знать: основы современной технологии разработки программ. Уметь: планировать разработку сложного программного обеспечения. Владеть (иметь навык(и)): планирования разработки сложного программного обеспечения.
		ОПК-13.16	владеет методами оценки качества готового программного обеспечения	Знать: методы тестирования и отладки программного обеспечения. Уметь: проводить тестирование и отладку программного обеспечения. Владеть (иметь навык(и)): тестирования и отладки программного обеспечения.
		ОПК-13.17	владеет навыками разработки алгоритмов для решения типовых профессиональных задач	Знать: основы современной технологии разработки программ. Уметь: разрабатывать алгоритмы для решения сложных задач. Владеть (иметь навык(и)): разработки алгоритмов для решения сложных задач.

12. Объем дисциплины в зачетных единицах/час — 4/144.

Форма промежуточной аттестации экзамен

13. Виды учебной работы

Вид учебной работы	Трудоемкость			
	Всего	По семестрам		
		2 семестр		
Аудиторные занятия	68	68		
в том числе:	лекции	34	34	
	практические			
	лабораторные	34	34	
Самостоятельная работа	40	40		
в том числе: курсовая работа (проект)				
Форма промежуточной аттестации (зачет – 0 час. / экзамен – __ час.)	36	36		
Итого:	144	144		

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Лекции		
1.1	Сортировки и поиск	Сортировка простым выбором. Сортировка включениями. Сортировка простыми включениями. Сортировка бинарными включениями. Сортировка обменом. Сортировка простым обменом. Шейкер-сортировка. Сортировка Шелла. Сортировка подсчетом. Поиск в неупорядоченном массиве.

		Поиск с барьером. Поиск в упорядоченном массиве. Бинарный поиск. Хеширование
1.2	Динамические структуры данных	Указатели. Физическая структура указателя. Представление указателей в языках программирования. Операции над указателями. Действия с указателями. Операторы для указателей (pointer). Динамические переменные. Динамические структуры данных. Указатели и ссылки. Линейные списки. Основные операции. Списки, стеки, очереди. Упорядоченный список. Частотный словарь. Слияние двух упорядоченных списков. Двусвязный список. Кольцевой список.
1.3	Списки, стеки, очереди	Стеки. Динамическая реализация стека. Стек, реализованный с помощью массива. Очереди. Динамическая реализация очереди. Очередь, реализованная с помощью массива
1.4	Рекурсия	Рекурсивные определения и рекурсивные алгоритмы. Когда рекурсия необходима. Примеры рекурсивных программ. "Ханойские башни". Быстрая сортировка. Алгоритмы с возвратом. Расстановка ферзей. Задача оптимального выбора
1.5	Алгоритмы на деревьях	Понятия и определения. Основные операции с бинарными деревьями. Упорядоченные деревья. Поиск по дереву с включением. Поиск по дереву с включением. Удаление из упорядоченного дерева. Сбалансированные деревья. Включение в сбалансированное дерево. Удаление из сбалансированного дерева. Сортировки на деревьях. Турнирная сортировка. Сортировка частично упорядоченным деревом. Основы работы интерпретатора. Интерпретатор для простых операторов
1.6	Алгоритмы на графах	Проект для алгоритмов на графах. Структура данных. Изображение графов. Чтение и запись графов. Поиск в графах. Поиск в глубину. Поиск в ширину. Остов графа. Кратчайшие пути. Волновой алгоритм. Алгоритм Дейкстры. Алгоритм Форда-Мура-Беллмана. Алгоритм Флойда. Циклы на графах. Циклы на графах. Эйлеровы циклы. Гамильтонов цикл. Алгоритмы с возвратом. Гамильтоновы циклы и задача коммивояжера. Метод перебора. Кратчайший незамкнутый путь. Алгоритм штрафов вершин. Комбинаторные задачи на графах. Минимальная раскраска графа. Переборный алгоритм. Приближенные алгоритмы раскраски графа, основанные на понятии соцветных вершин. Алгоритмы о связности графа. Топологическая сортировка. Остовное дерево наименьшей стоимости. Алгоритмы Прима и Краскала. Выделение компонент связности
2. Лабораторные работы		
2.1	Сортировки и поиск	Сортировка простым выбором. Сортировка включениями. Сортировка простыми включениями. Сортировка бинарными включениями. Сортировка обменом. Сортировка простым обменом. Шейкер-сортировка. Сортировка Шелла. Сортировка подсчетом. Поиск в неупорядоченном массиве. Поиск с барьером. Поиск в упорядоченном массиве. Бинарный поиск. Хеширование
2.2	Динамические структуры данных	Указатели. Физическая структура указателя. Представление указателей в языках программирования. Операции над указателями. Действия с указателями. Операторы для указателей (pointer). Динамические переменные. Динамические структуры данных. Указатели и ссылки. Линейные списки. Основные операции. Списки, стеки, очереди. Упорядоченный список. Частотный словарь. Слияние двух упорядоченных списков. Двусвязный список. Кольцевой список.
2.3	Списки, стеки, очереди	Стеки. Динамическая реализация стека. Стек, реализованный с помощью массива. Очереди.

		Динамическая реализация очереди. Очередь, реализованная с помощью массива
2.4	Рекурсия	Рекурсивные определения и рекурсивные алгоритмы. Когда рекурсия необходима. Примеры рекурсивных программ. "Ханойские башни". Быстрая сортировка. Алгоритмы с возвратом. Расстановка ферзей. Задача оптимального выбора
2.5	Алгоритмы на деревьях	Понятия и определения. Основные операции с бинарными деревьями. Упорядоченные деревья. Поиск по дереву с включением. Поиск по дереву с включением. Удаление из упорядоченного дерева. Сбалансированные деревья. Включение в сбалансированное дерево. Удаление из сбалансированного дерева. Сортировки на деревьях. Турнирная сортировка. Сортировка частично упорядоченным деревом. Основы работы интерпретатора. Интерпретатор для простых операторов
2.6	Алгоритмы на графах	Проект для алгоритмов на графах. Структура данных. Изображение графов. Чтение и запись графов. Поиск в графах. Поиск в глубину. Поиск в ширину. Остов графа. Кратчайшие пути. Волновой алгоритм. Алгоритм Дейкстры. Алгоритм Форда-Мура-Беллмана. Алгоритм Флойда. Циклы на графах. Циклы на графах. Эйлеровы циклы. Гамильтонов цикл. Алгоритмы с возвратом. Гамильтоновы циклы и задача коммивояжера. Метод перебора. Кратчайший незамкнутый путь. Алгоритм штрафов вершин. Комбинаторные задачи на графах. Минимальная раскраска графа. Переборный алгоритм. Приближенные алгоритмы раскраски графа, основанные на понятии соцветных вершин. Алгоритмы о связности графа. Топологическая сортировка. Остовное дерево наименьшей стоимости. Алгоритмы Прима и Краскала. Выделение компонент связности

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1	Сортировки и поиск	4		4	6	14
2	Динамические структуры данных	4		4	6	14
3	Списки, стеки, очереди	4		4	8	16
4	Рекурсия	4		4	8	16
5	Алгоритмы на деревьях	5		5	6	16
6	Алгоритмы на графах	5		5	6	16
	Итого:	34		34	40	144

14. Методические указания для обучающихся по освоению дисциплины

Студентам рекомендуется читать рекомендованную литературу, во время проверки выполнения лабораторных работ, преподавателю рекомендуется проводить теоретический опрос с целью определения степени усвоения материала.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1.	учебное пособие Тюкачев Н.А., Хлебостроев В.Г. С#. Программирование. Алгоритмы и структуры данных Часть 2 С#. Основы программирования ВГУ.- Воронеж, 2013. –308 с. (Учебник ВГУ)

б) дополнительная литература:

№ п/п	Источник
1.	Либерти Джесс. Программирование на С#. «Символ Плюс». 684 с.
2.	Нортроп Тони, Уилдермьюс Шон, Райан Билл. Основы разработки приложений на платформе Microsoft .NET Framework. Учебный курс Microsoft / Пер. с англ. – М.: «Русская Редакция», СПб.: «Питер», 2007. – 864 стр.: ил.
3.	Уотсон К. и др. С#. / Карли Уотсон, Марко Беллиназо, Олли Корне, Дэвид Эспиноза, Зах Гринфосс,
4.	Кристиан Нейджел, Джейкоб Хаммер Педерсен, Джон Рейд, Мэттью Рейнольде, Морган Скиннер, Эрик Уайт. Из-во «Лори», 2005, 861 с.
5.	Уотсон К. и др. Microsoft, Visual C# 2008. Базовый курс. Карли Уотсон, Кристиан Нейджел, Якоб Хаммер Педерсен, Джон Д. Рид, Морган Скиннер, Эрик Уайт. "Диалектика" Москва • Санкт-Петербург • Киев, 2009. 1211 с.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет):

№ п/п	Ресурс
1.	https://lib.vsu.ru/?p=4&t=8
2.	http://e.lanbook.com/books/Глухов М.М., Козлитин О.А., Шапошников В.А., Шишков А.Б. Задачи и упражнения по математической логике, дискретным функциям и теории алгоритмов
3.	http://e.lanbook.com/books/Асанов М.О., Баранский В.А., Расин В.В. Дискретная математика: графы, матроиды, алгоритмы

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	учебное пособие Тюкачев Н.А., Хлебостроев В.Г. С#. Программирование. Алгоритмы и структуры данных Часть 2 С#. Основы программирования ВГУ.- Воронеж, 2013. –308 с. (Учебник ВГУ)
2	Либерти Джесс. Программирование на С#. «Символ Плюс». 684 с.
3	http://e.lanbook.com/books/Глухов М.М., Козлитин О.А., Шапошников В.А., Шишков А.Б. Задачи и упражнения по математической логике, дискретным функциям и теории алгоритмов

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные системы (при необходимости)

Среда программирования Visual Studio

18. Материально-техническое обеспечение дисциплины:

Мультимедийная лекционная аудитория, компьютерный класс, программное обеспечение: среда программирования Visual Studio

19. Фонд оценочных средств:

19.1. Перечень компетенций с указанием этапов формирования и планируемых результатов обучения

Код и содержание компетенции (или ее части)	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенции посредством формирования знаний, умений, навыков)	Этапы формирования компетенции (разделы (темы) дисциплины или модуля и их наименование)	ФОС (средства оценивания)
ОПК-7.7. Знает базовые структуры данных	Знать: базовые классы, реализующие структуры в языке С#, динамические структуры данных (стеки, очереди), деревья, графы.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: применять базовые классы, реализующие	1. Сортировки и поиск 2. Динамические структуры данных	Задания лабораторных

	структуры на языке C#, динамические структуры данных (стеки, очереди), деревья графы.	3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	работ, КИМ
	Владеть (иметь навык(и)): Базовыми классами, реализующими структуры на языке C#, динамические структуры данных (стеки, очереди), деревья графы.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-7.8. Знает основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы	Знать: основные алгоритмы сортировки и поиска данных, рекурсивные алгоритмы, алгоритмы с возвратом, применяемые в комбинаторных алгоритмах и алгоритмах на графах.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: реализовать алгоритмы сортировки и поиска данных, рекурсивные алгоритмы, алгоритмы с возвратом, применяемые в комбинаторных алгоритмах и алгоритмах на графах.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Владеть (иметь навык(и)): алгоритмами сортировки и поиска данных, рекурсивные алгоритмы и алгоритмы с возвратом, применяемые в комбинаторных алгоритмах и алгоритмах на графах.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-13.10. Знает базовые структуры данных	Знать: базовые классы, реализующие структуры в языке C#, динамические структуры данных (стеки, очереди), деревья, графы.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: разрабатывать компоненты для программ, с использованием базовых классов, реализующие динамические структуры данных (стеки, очереди), деревья графы.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Владеть (иметь навык(и)): навыками разработки компонент для программ, реализующие динамические структуры данных (стеки, очереди), деревья графы.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-13.11. Знает основные комбинаторные и теоретико-графовые алгоритмы, а также способы их эффективной реализации и оценки вычислительной	Знать: основные алгоритмы сортировки и поиска данных, рекурсивные алгоритмы, алгоритмы с возвратом, применяемые в комбинаторных алгоритмах и алгоритмах на графах.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: реализовать компоненты для программ с использованием алгоритмов сортировки и	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия	Задания лабораторных работ, КИМ

сложности;	поиска данных, рекурсивных алгоритмов, алгоритмов с возвратом, применяемых в комбинаторных алгоритмах и алгоритмах на графах.	5. Алгоритмы на деревьях 6. Алгоритмы на графах	
	Владеть (иметь навык(и)): навыками разработки компонент для программ с использованием алгоритмов сортировки и поиска данных, рекурсивных алгоритмов и алгоритмов с возвратом, применяемых в комбинаторных алгоритмах и алгоритмах на графах.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-13.12. Умеет формализовать поставленную задачу	Знать: основы современной технологии разработки программ.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: формализовать поставленную задачу для реализации алгоритмов и программ.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Владеть (иметь навык(и)): формализации поставленной задачи для разработки алгоритмов и программ.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-13.13. Умеет разрабатывать эффективные алгоритмы и программы	Знать: основы современной технологии разработки программ.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: разрабатывать алгоритмы для поставленной задачи.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Владеть (иметь навык(и)): разработки алгоритмов для поставленной задачи.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-13.14. Умеет проводить оценку вычислительной сложности алгоритма	Знать: методику оценки вычислительной сложности алгоритма.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: проводить оценку вычислительной сложности алгоритма.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях	Задания лабораторных работ, КИМ

		6. Алгоритмы на графах	
	Владеть (иметь навык(и)): оценки вычислительной сложности алгоритма.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-13.15. Умеет планировать разработку сложного программного обеспечения	Знать: основы современной технологии разработки программ.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: планировать разработку сложного программного обеспечения.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Владеть (иметь навык(и)): планирования разработки сложного программного обеспечения.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-13.16. Владеет методами оценки качества готового программного обеспечения	Знать: методы тестирования и отладки программного обеспечения.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: проводить тестирование и отладку программного обеспечения.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Владеть (иметь навык(и)): тестирования и отладки программного обеспечения.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
ОПК-13.17. Владеет навыками разработки алгоритмов для решения типовых профессиональных задач	Знать: основы современной технологии разработки программ.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Уметь: разрабатывать алгоритмы для решения сложных задач.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
	Владеть (иметь навык(и)): разработки алгоритмов для решения сложных задач.	1. Сортировки и поиск 2. Динамические структуры данных 3. Списки, стеки, очереди 4. Рекурсия 5. Алгоритмы на деревьях 6. Алгоритмы на графах	Задания лабораторных работ, КИМ
Промежуточная аттестация - экзамен			КИМ

19.2 Описание критериев и шкалы оценивания компетенций (результатов обучения) при промежуточной аттестации

Для оценивания результатов обучения на экзамене используются следующие показатели (ЗУНы из 19.1): владение теоретическими основами дисциплины, способностью иллюстрировать ответ примерами, применять теоретические знания для решения практических задач.

Для оценивания результатов обучения на экзамене используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<i>Обучающийся в полной мере владеет теоретическими основами дисциплины, способен иллюстрировать ответ примерами, фактами, данными научных исследований, применять теоретические знания для решения практических задач.</i>	<i>Повышенный уровень</i>	<i>Отлично</i>
<i>Обучающийся владеет теоретическими основами дисциплины, способен выполнить несложные лабораторные задания, допускает ошибки при устном ответе.</i>	<i>Базовый уровень</i>	<i>Хорошо</i>
<i>Обучающийся владеет частично теоретическими основами дисциплины, фрагментарно способен ответить на вопросы, не умеет применять теоретические знания, но может выполнить несложные лабораторные задачи.</i>	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
<i>Результаты аттестации не соответствуют трем из перечисленных показателей. Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки.</i>	<i>–</i>	<i>Неудовлетворительно</i>

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

19.3.1. Перечень вопросов к экзамену

1. Классы ввода и вывода.
2. Класс FileStream.
3. Позиция в файле.
4. Текстовые классы.
5. Класс StreamWriter.
6. Запись в текстовый файл.
7. Класс StreamReader.
8. Чтение из текстового файла.
9. Частотный словарь.
10. Файлы прямого доступа.
11. Запись в двоичный файл.
12. Чтение двоичного файла.
13. Запись строки в файл прямого доступа.
14. Чтение строки из файла прямого доступа.
15. Запись и чтение массива.
16. Запись и чтение типизированного файла.
17. Другие классы для работы с файлами. Классы File и Directory.
18. Класс FileInfo.
19. Класс DirectoryInfo
20. Сортировка простым выбором.
21. Сортировка включениями.
22. Сортировка простыми включениями.
23. Сортировка бинарными включениями.

24. Сортировка обменом.
25. Сортировка простым обменом.
26. Шейкер-сортировка.
27. Сортировка Шелла.
28. Сортировка подсчетом.
29. Поиск в неупорядоченном массиве.
30. Поиск с барьером.
31. Поиск в упорядоченном массиве.
32. Бинарный поиск.
33. Класс Hashtable.
34. Множества. Определения. Операции. Свойства.
35. Реализация класса MySet с использованием класса HashTable
36. Рекурсивные определения и рекурсивные алгоритмы.
37. "Ханойские башни".
38. Быстрая сортировка.
39. Алгоритмы с возвратом. Расстановка ферзей.
40. Задача оптимального выбора
41. Основные операции с бинарными деревьями.
42. Упорядоченные деревья.
43. Поиск по дереву с включением.
44. Удаление из упорядоченного дерева.
45. Турнирная сортировка.
46. Основы работы интерпретатора.
47. Интерпретатор
48. Проект для алгоритмов на графах.
49. Структура данных для графа.
50. Изображение графов.
51. Запись и чтение графов.
52. Поиск в графах. Поиск в глубину.
53. Поиск в ширину.
54. Остов графа.
55. Кратчайшие пути.
56. Волновой алгоритм.
57. Алгоритм Дейкстры
58. Алгоритм Форда-Мура-Беллмана.
59. Алгоритм Флойда
60. Циклы на графах. Эйлеровы циклы.
61. Гамильтонов цикл.
62. Алгоритмы с возвратом.
63. Гамильтоновы циклы и задача коммивояжера.
64. Метод перебора.
65. Кратчайший незамкнутый путь.
66. Комбинаторные задачи на графах.
67. Минимальная раскраска графа.
68. Переборный алгоритм.
69. Приближенные алгоритмы раскраски графа, основанные на понятии соцветных вершин.
70. Алгоритмы о связности графа.
71. Топологическая сортировка.
72. Остовное дерево наименьшей стоимости.
73. Алгоритмы Прима и Краскала.
74. Выделение компонент связности

19.3.2 Перечень тем лабораторных работ

1. Линейные алгоритмы
2. Логическое выражение
3. Условный оператор
4. Циклы
5. Последовательности чисел

6. Строки
7. Одномерные массивы
8. Процедуры и функции
9. Матрицы
10. Перечислимый тип
11. Рекурсия
12. Списки, стеки, очереди
13. Множества
14. Сортировки
15. Разбор выражений
16. Деревья
17. Графы
18. Файлы

19.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины осуществляется в ходе текущей и промежуточной аттестаций.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в форме устного, оценки лабораторных работ. Критерии оценивания приведены выше.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Контрольно-измерительные материалы промежуточной аттестации включают в себя теоретические вопросы, позволяющие оценить уровень полученных знаний. Лабораторные работы позволяют оценить степень сформированности умений и навыков программирования. При оценивании используются качественные шкалы оценок. Критерии оценивания приведены выше.