

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

УТВЕРЖДАЮ

Заведующий кафедрой
Программирования и информационных технологий

 _____ проф. Махортов С.Д.,
05.03.2025

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.В.04 Построение и анализ алгоритмов

1. Код и наименование направления подготовки/специальности:

09.04.04 Программная инженерия

2. Профиль подготовки/специализация:

Системное программирование

3. Квалификация (степень) выпускника: магистр

4. Форма обучения: Очная

5. Кафедра, отвечающая за реализацию дисциплины: Программирования и
информационных технологий

6. Составители программы:

ст. преподаватель каф. ПиИТ Соломатин Дмитрий Иванович
e-mail: solomatin@cs.vsu.ru факультет: Компьютерных наук
кафедра: Программирования и информационных технологий

7. Рекомендована:

НМС ф-та компьютерных наук, протокол № 5 от 05.03.2025

8. Учебный год: 2025-2026

Семестр(ы): 1

9. Цели и задачи учебной дисциплины:

Изучение различных подходов к построению эффективных алгоритмов (жадные алгоритмы, динамическое программирование, эвристические алгоритмы, алгоритмы поисковой оптимизации и т.п.), овладение практическими навыками построения и анализа алгоритмов для решения практических задач.

10. Место учебной дисциплины в структуре ООП: Дисциплина относится к части блока Б1, формируемой участниками образовательных отношений.

Для успешного освоения дисциплины необходимы знания из дискретной математики, базовых структур данных и алгоритмов, а также уверенные навыки практического программирования.

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
ПК-4	Способен выполнять экспертную поддержку разработки прототипов ИС	ПК-4.1	Умеет вырабатывать варианты реализации прототипов ИС на основе накопленного опыта	Знать: классические переборные и оптимизационные алгоритмы, подходы к построению эвристических алгоритмов Уметь: применять изученные алгоритмы к прикладным задачам для реализации прототипов ИС Владеть: навыками оценки целесообразности и эффективности применения тех или иных алгоритмов при реализации прототипа ИС
		ПК-4.2	Умеет выполнять экспертную оценку предложенного варианта реализации прототипа ИС	

12. Объем дисциплины в зачетных единицах/час. (в соответствии с уч. планом) – 3 / 108.

Форма промежуточной аттестации – Зачёт с оценкой

13. Виды учебной работы

Вид учебной работы		Трудоемкость			
		Всего	По семестрам		
			1 сем.	–	–
Аудиторные занятия		36	36	–	–
в том числе:	лекции	18	18	–	–
	практические	–	–	–	–
	лабораторные	18	18	–	–
Самостоятельная работа		72	72	–	–
в том числе: курсовая работа (проект)		–	–	–	–
Форма промежуточной аттестации (зачет – 0 час. / экзамен – 36 час.)		–	–	–	–
Итого:		108	108	–	–

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
1.1	Введение в предмет	Цели и задачи изучения дисциплины, связь алгоритмов и структур данных.	
1.2	Понятие сложности алгоритма	Понятие алгоритма, свойства алгоритма, задача оценки и сравнения эффективности алгоритмов; временная и пространственная сложность, асимптотические оценки сложности алгоритмов для различных случаев и их свойства; практические принципы оценки сложности алгоритмов в O-нотации, примеры алгоритмов с различной сложностью.	
1.3	Алгоритмы сортировки	Свойства алгоритмов сортировки: время (временная сложность), память (пространственная сложность), устойчивость и естественность поведения, внутренние и внешние сортировки. Классические «быстрые» ($O(n \cdot \log(n))$) сортировки со сравнением элементом: сортировка слиянием (merge sort), быстрая сортировка (quick sort), пирамидальная сортировка (heap sort); сортировки «без сравнений»: сортировка подсчетом (counting sort), блочная сортировка (bucket sort).	
1.4	Базовые структуры данных и их свойства	Классификация структур данных: связные списки, стеки и очереди, очереди с приоритетом, деревья, хеш-таблицы. Основные методы и свойства структур данных, варианты реализации.	
1.5	Алгоритмы на графах	Варианты реализации графов. Обход и поиск на графах, поиск кратчайших путей (алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Флойда-Уоршелла). Поиск минимального остовного дерева (алгоритм Прима, алгоритм Краскала). Потоки в сетях и паросочетание в двудольных графах. Применение перечисленных алгоритмов и подходов к решению практических задач.	
1.6	Комбинаторный поиск и эвристические алгоритмы	Перебор с возвратом (генерация всех подмножеств, генерация перестановок, генерация всех путей в графе). Отсечение вариантов поиска. Эвристические методы перебора (произвольная выборка, локальный поиск, имитация отжига).	
1.7	Жадные алгоритмы	Применимость жадных алгоритмов. Рекурсивная и итеративная реализация. Коды Хаффмана и другие примеры практических задач. Жадные алгоритмы и матроиды.	
1.8	Динамическое программирование	Одномерные и двумерные задачи динамического программирования. Классические задачи: о наибольшей общей подпоследовательности, поиска наибольшей увеличивающейся подпоследовательности, редакционное расстояние (расстояние Левенштейна), задача о рюкзаке и др. Разбор практических примеров, доказательство корректности.	
1.9	Генетические алгоритмы	Генетические алгоритмы как модель эволюции живых существ. Принципы генетических алгоритмов: генетическая рекомбинация, мутации и естественный отбор. Особенности применения генетических алгоритмов и их преимущества перед традиционным	

		подходом в ряде задач	
1.1 0	Муравьиные алгоритмы	Модель поведения колонии муравьев при поиске пищи. Ферменты, их размещение муравьем при движении и испарение фермента. Сравнение алгоритма движения муравьев с классическим алгоритмом ветвей и границ задачи коммивояжера.	
1.1 1	Обзор других подходов при построении эвристических алгоритмов	Гармонический поиск (Harmony Search), искусственные иммунные системы (Artificial Immune Systems), гравитационный поиск (Gravitational Search), разбросанный поиск (Scatter Search), метод перекрестной энтропии (Cross-Entropy Method) и др.	
2. Практические занятия			
2.1	нет		
3. Лабораторные работы			
3.1	Понятие сложности алгоритма	Понятие алгоритма, свойства алгоритма, задача оценки и сравнения эффективности алгоритмов; временная и пространственная сложность, асимптотические оценки сложности алгоритмов для различных случаев и их свойства; практические принципы оценки сложности алгоритмов в O-нотации, примеры алгоритмов с различной сложностью.	
3.2	Алгоритмы сортировки	Свойства алгоритмов сортировки: время (временная сложность), память (пространственная сложность), устойчивость и естественность поведения, внутренние и внешние сортировки. Классические «быстрые» ($O(n \cdot \log(n))$) сортировки со сравнением элементом: сортировка слиянием (merge sort), быстрая сортировка (quick sort), пирамидальная сортировка (heap sort); сортировки «без сравнений»: сортировка подсчетом (counting sort), блочная сортировка (bucket sort).	
3.3	Базовые структуры данных и их свойства	Классификация структур данных: связные списки, стеки и очереди, очереди с приоритетом, деревья, хеш-таблицы. Основные методы и свойства структур данных, варианты реализации.	
3.4	Алгоритмы на графах	Варианты реализации графов. Обход и поиск на графах, поиск кратчайших путей (алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Флойда-Уоршелла). Поиск минимального остовного дерева (алгоритм Прима, алгоритм Краскала). Потоки в сетях и паросочетание в двудольных графах. Применение перечисленных алгоритмов и подходов к решению практических задач.	
3.5	Комбинаторный поиск и эвристические алгоритмы	Перебор с возвратом (генерация всех подмножеств, генерация перестановок, генерация всех путей в графе). Отсечение вариантов поиска. Эвристические методы перебора (произвольная выборка, локальный поиск, имитация отжига).	
3.6	Жадные алгоритмы	Применимость жадных алгоритмов. Рекурсивная и итеративная реализация. Коды Хаффмана и другие примеры практических задач. Жадные алгоритмы и матроиды.	
3.7	Динамическое программирование	Одномерные и двумерные задачи динамического программирования. Классические задачи: о наибольшей общей подпоследовательности, поиска наибольшей увеличивающейся подпоследовательности, редакционное расстояние (расстояние Левенштейна), задача о рюкзаке и др. Разбор практических примеров, доказательство корректности.	

3.8	Генетические алгоритмы	Генетические алгоритмы как модель эволюции живых существ. Принципы генетических алгоритмов: генетическая рекомбинация, мутации и естественный отбор. Особенности применения генетических алгоритмов и их преимущества перед традиционным подходом в ряде задач	
3.9	Муравьиные алгоритмы	Модель поведения колонии муравьев при поиске пищи. Ферменты, их размещение муравьем при движении и испарение фермента. Сравнение алгоритма движения муравьев с классическим алгоритмом ветвей и границ задачи коммивояжера.	
3.10	Обзор других подходов при построении эвристических алгоритмов	Гармонический поиск (Harmony Search), искусственные иммунные системы (Artificial Immune Systems), гравитационный поиск (Gravitational Search), разбросанный поиск (Scatter Search), метод перекрестной энтропии (Cross-Entropy Method) и др.	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Введение в предмет	1	–	–	4	5
2	Понятие сложности алгоритма	1	–	2	4	7
3	Алгоритмы сортировки	1	–	1	10	12
4	Базовые структуры данных и их свойства	1	–	2	8	11
5	Алгоритмы на графах	2	–	1	8	11
6	Комбинаторный поиск и эвристические алгоритмы	2	–	2	8	12
7	Жадные алгоритмы	2	–	2	8	12
8	Динамическое программирование	2	–	2	6	10
9	Генетические алгоритмы	2	–	2	6	10
10	Муравьиные алгоритмы	2	–	2	6	10
11	Обзор других подходов при построении эвристических алгоритмов	2	–	2	4	8
	Итого:	18	–	18	72	108

14. Методические указания для обучающихся по освоению дисциплины

Рекомендуется работа с конспектами лекций, презентационным материалом, выполнение всех лабораторных и контрольных работ, заданий текущей аттестации. Учебные и методические материалы по дисциплине размещены на сетевом диске, доступным на любом компьютере в локальной сети ФКН.

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

а) основная литература:

№ п/п	Источник
1	Скиена С. Алгоритмы. Руководство по разработке / Стивен С. Скиена. — 2-е изд. — СПб.: БХВПетербург, 2011. — 720 с.
2	Кормен, Томас. Алгоритмы : Построение и анализ : [Учебник] / Т. Кормен, Ч. Лейзерсон, Р. Ривест ; Пер. с англ. К. Белов и др.; Науч. ред. А. Шень .— М. : МЦНМО, 2002. — 955 с. : ил. — (Классические учебники: computer science) .— ISBN 5-900916-37-5.

3	Седжвик, Роберт. Фундаментальные алгоритмы на Java : Пер. с англ. / Р. Седжвик .— М. : DiaSoft, 2003. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2003 .— 680 с. : ил. — Библиогр. в конце глав. - Предм. указ.: с. 680-687 .— Парал. тит. л. англ. — ISBN 966-7992-22-5.
4	Ахо, Альфред В. Структуры данных и алгоритмы : [Учебное пособие] / Альфред В. Ахо, Джон Э. Хопкрофт, Джеффри Д. Ульман; Пер. с англ. и ред. А.А. Минько .— М. и др. : Вильямс, 2003 .— 382 с. : ил., табл. — Парал. тит. л. англ. — Библиогр.: с.369-374 .— Предм. указ.: с. 375-382 .— ISBN 5-8459-0122-7.

б) дополнительная литература:

№ п/п	Источник
5	Седжвик, Роберт. Фундаментальные алгоритмы на C++ / Р. Седжвик .— М. и др. : DiaSoft, 2002. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2002 .— 687 с. : ил. — ISBN 5-93772047-4.
6	Седжвик, Роберт. Фундаментальные алгоритмы на C++ / Р. Седжвик .— М. и др. : DiaSoft, 2002. — Ч.5: Алгоритмы на графах .— 2002 .— 484 с. : ил. — Парал. тит. л. англ. — ISBN 5-93772-054-7 .— ISBN 0-201-36118-3.
7	Седжвик, Роберт. Фундаментальные алгоритмы на C : Пер. с англ. / Р. Седжвик ; Принстонский ун-т .— 3-е изд. — СПб. и др. : DiaSoft, 2003. — Ч.1-4: Анализ. Структуры данных. Сортировка. Поиск .— 2003 .— 670 с. : ил. — Парал. тит. л. англ. — ISBN 5-93772-081-4.
8	Седжвик, Роберт. Фундаментальные алгоритмы на C : Пер. с англ. / Р. Седжвик ; Принстонский ун-т .— 3-е изд. — СПб., и др. : DiaSoft, 2003. — Ч.5: Алгоритмы на графах .— 2003 .— С.6611127 : ил .— Парал. тит. л. англ. — ISBN 5-93772-082-2.

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

№ п/п	Источник
9	MAXimal::algo [Электронный ресурс] : — Режим доступа: http://e-maxx.ru/algo/
10	Алгоритмы: теория и практика. Методы [Электронный ресурс] : — Режим доступа: https://stepik.org/course/217/promo
11	Алгоритмы программирования и структуры данных [Электронный ресурс] : — Режим доступа: https://openedu.ru/course/ITMOUniversity/PADS/

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Гладков, Леонид Анатольевич. Генетические алгоритмы : учебное пособие для студ. вузов, обуч. по направлениям "Информатика и вычислительная техника" и "Информационные системы" / Л.А. Гладков, В.В. Курейчик, В.М. Курейчик ; под ред. В.М. Курейчика .— Изд. 2-е, испр. и доп. — М. : Физматлит, 2006 .— 319 с. : ил. — Библиогр. в конце разд. — ISBN 5-9221-0510-8.
2	Рутковская, Данута. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский ; пер. с пол. И.Д. Рудинского .— М. : Горячая линияТелеком, 2007 .— 383 с. : ил .— Библиогр. в конце гл. — Предм. указ. : с.381-383 .— ISBN 593517-103-1.

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

№ п/п	Наименование
1	Python версии 3.5 или выше с установленными дополнительными библиотеками (возможен вариант в виде дистрибутива Anaconda) - бесплатен
2	Среда разработки PyCharm (академическая лицензия или версия Community) - бесплатна
3	OpenJDK - бесплатен
4	Среда разработки NetBeans или IntelliJ IDEA (академическая лицензия или версия Community) - бесплатны

18. Материально-техническое обеспечение дисциплины:

№ п/п	Наименование
1	Компьютерный класс с проектором (корп. 1а, ауд. № 382-385 или другие подходящие): ПК-Intel-i3 16 шт., специализированная мебель: доска маркерная 1 шт., столы и стулья в количестве, достаточном для размещения академической группы (подгруппы) студентов; выход в Интернет, доступ к фондам учебно-методической документации и электронным изданиям.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Введение в предмет	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
2	Понятие сложности алгоритма	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
3	Алгоритмы сортировки	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
4	Базовые структуры данных и их свойства	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
5	Алгоритмы на графах	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
6	Комбинаторный поиск и эвристические алгоритмы	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
7	Жадные алгоритмы	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
8	Динамическое программирование	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
9	Генетические алгоритмы	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
10	Муравьиные алгоритмы	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
11	Обзор других подходов при построении эвристических алгоритмов	ПК-4	ПК-4.1, ПК-4.2	Обязательные практические задания из пункта 20.1 (контроль и оценка выполнения)
Промежуточная аттестация форма контроля – экзамен				Перечень вопросов к экзамену из пункта 20.2

20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1. Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью контроля выполнения обязательных практических заданий.

Перечень заданий обсуждается с каждым студентом индивидуально.

20.2. Промежуточная аттестация Для оценивания результатов обучения на зачете используются следующие содержательные показатели (формулируется с учетом конкретных требований дисциплины):

- 1) знание теоретических основ учебного материала, основных определений, понятий и используемой терминологии;
- 2) умение проводить обоснование и представление основных теоретических и практических результатов (теорем, алгоритмов, методик) с использованием математических выкладок, блоксхем, структурных схем и стандартных описаний к ним;
- 3) умение связывать теорию с практикой, иллюстрировать ответ примерами, в том числе, собственными, умение выявлять и анализировать основные закономерности, полученные, в том числе, в ходе выполнения лабораторно-практических заданий;
- 4) умение обосновывать свои суждения и профессиональную позицию по излагаемому вопросу;
- 5) владение навыками программирования и экспериментирования в рамках выполняемых лабораторных заданий;

Различные комбинации перечисленных показателей определяют критерии оценивания результатов обучения (сформированности компетенций) на зачете:

- высокий (углубленный) уровень сформированности компетенций; –
- повышенный (продвинутый) уровень сформированности компетенций;
- пороговый (базовый) уровень сформированности компетенций.

Для оценивания результатов обучения на зачете с оценкой используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Соотношение показателей, критериев и шкалы оценивания результатов обучения на государственном экзамене представлено в следующей таблице.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Студент владеет основными понятиями учебной дисциплины, может пояснить большинство принципов на примерах; вовремя сдал все практические задания, которые выполнены на высоком уровне, без явных ошибок.	Повышенный уровень	Отлично
Студент владеет основными понятиями учебной дисциплины, однако в ответах на некоторые вопросы допускает неточности; сдал все практические задания, однако к некоторым решениям студента у преподавателя есть замечания.	Базовый уровень	Хорошо
Студент знает основные определения из учебной дисциплины, однако пояснить многие понятия на примерах затрудняется; сдал большую часть практических заданий, однако продемонстрированные решения содержат существенные	Пороговый уровень	Удовлетворительно

ошибки.		
Студент путается в основных понятиях учебной дисциплины, не может привести примеры; не сдал большую часть практических заданий.	–	Неудовлетворительно

Перечень вопросов к экзамену (зачету):

№ п/п	Вопрос
1	Понятие сложности алгоритма, практические приемы оценки сложности алгоритмов
2	Алгоритмы сортировки
3	Базовые структуры данных и их свойства
4	Реализация структур данных
5	Графы: основные понятия и определения, способы задания (хранения) графов, варианты реализации графов
6	Обход и поиск на графах, поиск кратчайших путей (алгоритм Дейкстры, алгоритм Беллмана-Форда, алгоритм Флойда-Уоршелла)
7	Поиск минимального остовного дерева (алгоритм Прима, алгоритм Краскала)
8	Комбинаторный поиск (генерация всех подмножеств, генерация перестановок, генерация всех путей в графе), отсечение вариантов поиска
9	Эвристические методы перебора
10	Жадные алгоритмы
11	Динамическое программирование
12	Генетические алгоритмы
13	Муравьиные алгоритмы
14	Подходы при построении эвристических алгоритмов