

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

**УТВЕРЖДАЮ**

Заведующий кафедрой программного обеспечения  
и администрирования информационных систем



Артемов М. А.

08.06.2018 г.

## **РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **Б1.В.ДВ.02.01 Комбинаторные алгоритмы**

**1. Шифр и наименование направления подготовки:**

02.03.03 Математическое обеспечение и администрирование информационных систем

**2. Профиль подготовки/специализация:** Информационные системы и базы данных

**3. Квалификация (степень) выпускника:** бакалавр

**4. Форма обучения:** очная

**5. Кафедра, отвечающая за реализацию дисциплины:**

Программного обеспечения и администрирования информационных систем

**6. Составители программы:**

Ефремов М.С., к.ф.-м.н., преп.

**7. Рекомендована:** НМС факультета ПММ протокол № 10 от 18.06.2018

**8. Учебный год:** 2018/2019      **Семестр:** пятый

**9. Цели и задачи учебной дисциплины:**

Цель курса - приобретение знаний и навыков в области комбинаторных алгоритмов и алгоритмов на графах, формирование алгоритмической подготовки высокого уровня.

Дисциплина знакомит студентов с фундаментальными алгоритмами, которые широко используются при разработке программного обеспечения. Целью курса является приобретение базовых знаний и навыков в области использования комбинаторных алгоритмов и алгоритмов на графах, приобретение опыта решения оптимизационных,

NP-полных задач, знакомство с основными проблемными задачами в курсе программирования, знакомство с вариантами решения задач искусственного интеллекта. Курс рассматривает различные общие подходы к решению задач по программированию и задач повышенной сложности, в частности.

В результате изучения дисциплины студент должен:

**Знать:** алгоритмы генерации перестановок, размещений и сочетаний; принципы оценки сложности алгоритмов; правила оптимизации программ; методы разработки алгоритмов; методы для обработки рекурсивных структур данных и рекурсивные алгоритмы; переборные алгоритмы и методы их упрощения; постановки оптимизационных задач на графах; эвристические методы решения задач.

**Уметь:** применять комбинаторные алгоритмы для решения задач методом перебора; оценивать сложность программ и производить их оптимизацию; использовать методы разработки алгоритмов; пользоваться рекурсивными структурами данных и рекурсивными алгоритмами обработки; использовать методы упрощения переборных алгоритмов; использовать эвристические алгоритмы для решения задач.

**Владеть:** навыками практического применения алгоритмов при решении конкретных задач.

#### 10. Место учебной дисциплины в структуре ООП:

Дисциплина относится к дисциплинам по выбору вариативной части учебного плана и изучается в 5-м семестре.

#### 11. Компетенции обучающегося, формируемые в результате освоения дисциплины:

Компетенция		Планируемые результаты обучения
Код	Название	
ОПК-2	способность применять в профессиональной деятельности знания математических основ информатики	<p><i>Знать:</i> алгоритмы генерации перестановок, размещений и сочетаний; принципы оценки сложности алгоритмов; правила оптимизации программ; методы разработки алгоритмов; методы для обработки рекурсивных структур данных и рекурсивные алгоритмы; переборные алгоритмы и методы их упрощения; постановки оптимизационных задач на графах; эвристические методы решения задач.</p> <p><i>Уметь:</i> применять комбинаторные алгоритмы для решения задач методом перебора; оценивать сложность программ и производить их оптимизацию; пользоваться рекурсивными структурами данных и рекурсивными алгоритмами обработки; использовать методы упрощения переборных алгоритмов; использовать эвристические алгоритмы для решения задач.</p> <p><i>Владеть:</i> методами разработки алгоритмов.</p>
ПК-3	готовность к разработке моделирующих алгоритмов и реализации их на базе языков и пакетов прикладных программ моделирования	<p><i>Знать:</i> алгоритмы генерации перестановок, размещений и сочетаний; принципы оценки сложности алгоритмов; правила оптимизации программ; методы разработки алгоритмов; методы для обработки рекурсивных структур данных и рекурсивные алгоритмы; переборные алгоритмы и методы их упрощения; постановки оптимизационных задач на графах; эвристические методы решения задач.</p> <p><i>Уметь:</i> применять комбинаторные алгоритмы для решения задач методом перебора; оценивать сложность программ и производить их оптимизацию;</p>

		использовать методы разработки алгоритмов; пользоваться рекурсивными структурами данных и рекурсивными алгоритмами обработки; использовать методы упрощения переборных алгоритмов; использовать эвристические алгоритмы для решения задач. <i>Владеть:</i> навыками практического применения алгоритмов при решении конкретных задач.
--	--	--

**12. Объем дисциплины в зачетных единицах/часах в соответствии с учебным планом — 3/108.**

**Форма промежуточной аттестации:** зачет с оценкой.

**13. Виды учебной работы**

Вид учебной работы	Трудоемкость (часы)
Аудиторные занятия	54
в том числе: лекции	
лабораторные	36
практические	18
Самостоятельная работа	54
Форма промежуточной аттестации	0
Итого:	108

**13.1. Содержание дисциплины**

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
<b>1. Практические занятия</b>		
1.1	Комбинаторные алгоритмы	Генерация перестановок (различный порядок генерации: лексикографический, с минимальным изменением, беспорядочные). Генерация размещений. Генерация сочетаний.  Обсуждение темы, разбор примеров.
1.2	Оценка производительности и оптимизация программ	Оценка производительности и сложности алгоритмов. Примеры программ. Правила оптимизации программ. Примеры оптимизации. Методы разработки алгоритмов. Метод «разделяй и властвуй». Итерационный метод.  Обсуждение темы, разбор примеров.
1.3	Рекурсивные структуры данных и алгоритмы	Деревья (двоичные, сбалансированные). «Красно-черные» деревья. Алгоритмы с возвратом (бэктрекинг).  Обсуждение темы, разбор примеров. Контрольная работа №1.
1.4	Перебор и методы его сокращения	Поиск в глубину. Поиск в ширину. Динамическое программирование. Примеры задач. «Жадные» алгоритмы. Метод ветвей и границ. Метод «решета».  Обсуждение темы, разбор примеров.

1.5	Алгоритмы на графах	Представление графа (матрицы и динамические структуры данных). Поиск в глубину. Поиск в ширину. Топологическая сортировка. Пример задачи с функциями. Минимальное остовное дерево. Определение связности. Поиск циклов. Нахождение кратчайшего пути. Задача о раскраске графа. Построение максимального потока в графе. Задача о пожарной команде и задача о рюкзаке. Задача коммивояжера, NP–полные задачи. Обсуждение темы, разбор примеров.
1.6	Эвристические алгоритмы	Эвристические алгоритмы. Случайные и генетические алгоритмы. Нейросетевые системы (обзор и пример алгоритмизации). Применение нечеткой логики. Обсуждение темы, разбор примеров.
<b>2. Лабораторные занятия</b>		
2.1	Комбинаторные алгоритмы	Решение задач по теме.
2.2	Оценка производительности и оптимизация программ	Решение задач по теме.
2.3	Рекурсивные структуры данных и алгоритмы	Решение задач по теме.
2.4	Перебор и методы его сокращения	Решение задач по теме.
2.5	Алгоритмы на графах	Решение задач по теме.
2.6	Эвристические алгоритмы	Решение задач по теме.

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Комбинаторные алгоритмы	0	8	4	10	22
2	Оценка производительности и оптимизация программ	0	4	2	8	14
3	Рекурсивные структуры данных и алгоритмы	0	4	2	12	18
4	Перебор и методы его сокращения	0	8	4	6	18
5	Алгоритмы на графах	0	4	2	12	18
6	Эвристические алгоритмы	0	8	4	6	18
	Итого:	0	36	18	54	108

### 14. Методические указания для обучающихся по освоению дисциплины

Выполнение практических заданий для самостоятельной работы, выполнение лабораторных работ, использование рекомендованной литературы и методических материалов, в том числе находящихся в личном кабинете. Выполнение контрольных работ.

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

а) основная литература:

№ п/п	Источник
1	Информатика: базовый курс: [учебное пособие для студ. вузов]; под ред. С.В. Симоновича. — 3-е изд. — СПб. [и др.]: Питер, 2012. — 637 с.

б) дополнительная литература:

№ п/п	Источник
2	Окулов, С. М. Программирование в алгоритмах [Электронный ресурс] / С. М. Окулов. - 5-е изд. (эл.). - М.: БИНОМ. Лаборатория знаний, 2014. - 383 с.: ил. <a href="http://znanium.com/catalog.php?bookinfo=502153">http://znanium.com/catalog.php?bookinfo=502153</a>
3	Окулов, С. М. Программирование в алгоритмах / С.М. Окулов. — 3-е изд. — М.: БИНОМ. Лаборатория знаний, 2007. — 383 с.
4	Бентли Дж. Жемчужины программирования / Джон Бентли; Пер. с англ. Д. Солнышков. — 2-е изд. — СПб. и др.: Питер, 2002. — 268 с. : ил.
5	МакКоннелл Дж. Анализ алгоритмов: Вводный курс / Дж. Макконелл; пер. с англ. С.К. Ландо. — М.: Техносфера, 2002. — 302 с. : ил.
6	Рейнгольд Э. Комбинаторные алгоритмы: Теория и практика / Э. Рейнгольд, Ю. Нивергельт, Н. Део; Пер. с англ. Е.П. Липатова; Под ред. В.Б. Алексеева. — М.: Мир, 1980. — 476 с. : ил.
7	Ахо А.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.
8	Новиков, Ф. А., Дискретная математика для программистов: [Учебник] / Ф. А. Новиков. — СПб, и др.: Питер, 2002. — 301с.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
19	Электронный каталог Научной библиотеки Воронежского государственного университета. — ( <a href="http://www.ru/lib.vsu/ru">http://www.ru/lib.vsu/ru</a> )

## 16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
-------	----------

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

ОС Windows, Delphi, Microsoft Visual Studio

## 18. Материально-техническое обеспечение дисциплины:

Аудитория, доска, ЭВМ

## 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)
<p>ОПК-2</p> <p>способность применять в профессиональной деятельности знания математических основ информатики</p>	<p>Знать:</p> <p>алгоритмы генерации перестановок, размещений и сочетаний;</p> <p>принципы оценки сложности алгоритмов; правила оптимизации программ;</p> <p>методы разработки алгоритмов; методы для обработки рекурсивных структур данных и рекурсивные алгоритмы;</p> <p>переборные алгоритмы и методы их упрощения;</p> <p>постановки оптимизационных задач на графах; эвристические методы решения задач.</p>	<p>Разделы 1.1 – 1.6.</p>	<p>Комплект КИМ №1-2</p>
	<p>Уметь:</p> <p>применять комбинаторные алгоритмы для решения задач методом перебора;</p> <p>оценивать сложность программ и производить их оптимизацию;</p> <p>использовать методы разработки алгоритмов;</p> <p>пользоваться рекурсивными структурами данных и рекурсивными алгоритмами обработки;</p> <p>использовать методы упрощения переборных алгоритмов;</p> <p>использовать эвристические алгоритмы для решения задач.</p>	<p>Разделы 2.1 – 2.6.</p>	<p>Контрольная работа №1</p>
<p>ПК-3</p> <p>готовность к разработке моделирующих алгоритмов и реализации их на базе языков и пакетов прикладных программ моделирования</p>	<p>Знать:</p> <p>алгоритмы генерации перестановок, размещений и сочетаний;</p> <p>принципы оценки сложности алгоритмов; правила оптимизации программ;</p> <p>методы разработки алгоритмов; методы для обработки рекурсивных структур данных и рекурсивные алгоритмы;</p>	<p>Разделы 1.1 – 1.6.</p>	<p>Комплект КИМ №1-2</p>

	переборные алгоритмы и методы их упрощения; постановки оптимизационных задач на графах; эвристические методы решения задач.		
	Уметь: применять комбинаторные алгоритмы для решения задач методом перебора; оценивать сложность программ и производить их оптимизацию; использовать методы разработки алгоритмов; пользоваться рекурсивными структурами данных и рекурсивными алгоритмами обработки; использовать методы упрощения переборных алгоритмов; использовать эвристические алгоритмы для решения задач.	Разделы 2.1 – 2.6.	Контрольная работа №1
	Владеть: навыками практического применения алгоритмов при решении конкретных задач.	Разделы 2.1 – 2.6.	Комплект КИМ №1-2 Контрольная работа №1
<b>Промежуточная аттестация</b>			Комплект КИМ №1 и №2

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

### Пример:

Для оценивания результатов обучения на экзамене/зачете используются следующие показатели:

- 1) владение навыками реализации программ;
- 2) знание методологии структурного программирования и применение ее на практике;
- 3) знание и умение использовать при решении задач базовые структуры данных;
- 4) умение выбирать и применять при реализации оптимальную структуру данных;
- 5) умение разрабатывать оптимальный алгоритм решения задачи и выполнять его реализацию;
- 4) знание теоретического материала

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

## Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Отличное знание теоретического материала, правильное и эффективное решение задачи. Должны быть выполнены ВСЕ лабораторные работы.	<i>Повышенный уровень</i>	<i>Отлично</i>
Хорошее знание теоретического материала, в целом правильное решение задачи. НО: допускает незначительные ошибки в решении задачи. ИЛИ: выполнены все показатели повышенного уровня, но не зачтена одна лабораторная работа, но студент продемонстрировал умение решать задачи по этой теме (это задача в КИМе)	<i>Базовый уровень</i>	<i>Хорошо</i>
Решение задачи не доведено до конца или недостаточное знание теоретического материала. Неоптимальное решение задачи и недостаточное владение теоретическим материалом. ИЛИ: выполнены все показатели базового уровня, но не зачтено более одной лабораторной работы.	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
Задача не решена или серьезные пробелы в знании теоретического материала.	–	<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 зачету с оценкой:

1. Алгоритм генерации перестановок в лексикографическом порядке, с минимальными изменениями, без порядка. Получение перестановки по номеру. Вычисление номера перестановки.
2. Генерация размещений без повторений. Получение размещения по номеру. Вычисление номера размещения.
3. Генерация сочетаний без повторений. Получение сочетания по номеру. Получение номера по сочетанию.
4. Оценка производительности и сложности алгоритмов. Правила оптимизации программ. Примеры оптимизации.
5. Метод «разделяй и властвуй».
6. Перебор с возвратом (бэктрекинг).
7. Перебор и методы его сокращения.
8. Динамическое программирование.
9. «Жадные» алгоритмы.
10. Волновой алгоритм.
11. Метод ветвей и границ.



12. Метод «решета».
13. Метод ветвей и границ.
14. Алгоритмы на графах. Представление графа (матрицы и динамические структуры данных). Поиск в глубину. Поиск в ширину.
15. Топологическая сортировка.
16. Минимальное остовное дерево.
17. Определение связности. Поиск циклов.
18. Нахождение кратчайшего пути.
19. Задача о раскраске графа.
20. Построение максимального потока в графе.
21. Задача коммивояжера, NP–полные задачи.
22. Эвристические алгоритмы. Случайные и генетические алгоритмы. Муравьиные алгоритмы.

### 19.3.2 Перечень практических заданий

Иллюстрируется на примере КИМ №1 и КИМ №2.

### 19.3.4 Перечень заданий для лабораторных работ

Иллюстрируется на примере КИМ №1 и КИМ №2

#### ПРИМЕРЫ

#### КИМ №1

<b>Вопрос</b>	Волновой алгоритм.
<b>Задача на практику</b>	Перечислить все разбиения натурального числа N на натуральные слагаемые (разбиения, отличающиеся лишь порядком слагаемых, считаются за одно) в следующих порядках (пример, при ЛГ=4):
<b>У</b>	<ul style="list-style-type: none"> <li>• 4, 3+1, 2+2, 2+1+1, 1+1+1+1;</li> <li>• 4, 2+2, 1+3, 1+1+2, 1+1+1+1;</li> <li>• 1+1+1+1, 1+1+2, 1+3, 2+2, 4.</li> </ul>
<b>Задача для лабораторной</b>	<p>Дан массив, состоящий из целых чисел. Размерность массива <math>1 \leq N \leq 100\,000</math>.</p> <p>Необходимо найти последовательность подряд идущих элементов с суммой наиболее близкой к нулю. Время работы программы не должно превышать 30 секунд.</p>

#### КИМ №2

<b>Вопрос</b>	Метод ветвей и границ.
<b>Задача на практику</b>	В массиве A (размерности N×M), заполненном нулями и единицами, найти квадратный блок максимального размера, состоящий из одних нулей.
<b>Задача для лабораторной</b>	<p>Коммивояжер. Задано N городов и известна матрица расстояний между городами. Торговец должен выйти из первого города, посетить по одному разу в неизвестном порядке города 2,3,4, ..., N и вернуться в первый город.</p> <p>Необходимо определить порядок посещения городов, который в итоге дает кратчайший из возможных путей. Предусмотреть вывод результата в текстовый файл.</p>

### 19.3.4 Перечень заданий для контрольных работ

Иллюстрируется на примерах заданий для контрольной работы №1

## Примеры заданий для контрольной работы № 1

**Примечание:** в задании №2 допускается использование любого процедурного языка программирования.

<b>Вариант № 1</b>
1. Сколько существует целочисленных решений уравнения $x_1+x_2+x_3+x_4=7$ ?
2. На множестве перестановок задан лексикографический порядок. Привести алгоритм генерации номера по заданной перестановке.
<b>Вариант № 2</b>
1. Сколько различных сообщений можно закодировать, меняя порядок 6 флажков: 2 красных, 3 синих и 1 зеленый?
2. На множестве сочетаний из N по M элементов задан антилексикографический порядок. Привести алгоритм генерации сочетания, следующего за заданным.

### Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно решены оба задания;
- оценка «хорошо» выставляется студенту, если решены оба задания, но имеются незначительные недочеты;
- оценка «удовлетворительно» выставляется студенту, если задание №1 решено верно, а при решении задания №2 допущены существенные ошибки;
- оценка «неудовлетворительно» выставляется студенту, если не выполняется ни один из вышеуказанных критериев оценки.

### 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 контрольных работ.

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

Контрольно-измерительные материалы промежуточной аттестации включают в себя практические задания, позволяющие оценить степень сформированности умений и навыков.

При оценивании используются количественные шкалы оценок. Критерии оценивания приведены выше.