

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.

8.06.2018 г.

## **РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.В.09 Java

**1. Код и наименование направления подготовки/специальности:**

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:** Курбатова И.В., к.ф.-м.н., доцент кафедры программного обеспечения и администрирования информационных систем

**7. Рекомендована:** НМС факультет ПММ № 10 от 18.06.2018

**8. Учебный год:** 2018/2019

**Семестр(ы):** 5,6

**9. Цели и задачи учебной дисциплины:** приобретение базовых знаний и навыков в области разработки приложений на языке Java.

В результате изучения дисциплины студент должен:

Знать:

- структуру приложений в кроссплатформенных языках на примере Java;
- структуры данных языка, шаблоны, типы массивов и векторов;
- базовые классы swing;
- классы обработки исключений;
- методы сетевого взаимодействия;
- понятия о принципах работы сборщика мусора;
- архитектуру MVC;
- принципы многопоточности и разделения ресурсов.

Уметь:

- реализовывать простейшие программы из нескольких исходных файлов (классов);
- применять объектно-ориентированную технологию для проектирования решения задач;
- генерировать (throw) и перехватывать (catch) исключения (Exceptions)
- работать с основными файловыми потоками в Java.

**10. Место учебной дисциплины в структуре ООП:** Дисциплина «Java» входит в вариативную часть программы бакалавриата, изучается в 5 и 6 семестрах и не требует предварительной подготовки.

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

Компетенция		Планируемые результаты обучения
Код	Название	
ОПК-7	Способность использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений	<p>знать: основы объектно-ориентированного проектирования и программирования, основные структуры данных и алгоритмы, имеющиеся в стандартной библиотеке;</p> <p>уметь: применять полученные знания для решения практических задач; пользоваться документацией к стандартной библиотеке и осуществлять поиск в ней; отлаживать программы в случае обнаружения нештатного поведения;</p> <p>владеть (иметь навык(и)): написания программ различного типа (вычислительных, сетевых и т.д.) с использованием различных типов и структур данных.</p>
ОПК-8	Способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными	<p>знать: основные функции утилит JDK, а также интегрированной среды разработки;</p> <p>уметь: выполнять запуск, отладку, сборку программы с использованием интегрированной среды разработки;</p> <p>владеть (иметь навык(и)): запуска, сборки и отладки программ различной сложности и размера.</p>

	средствами, поддерживающими создание программного обеспечения	
ОПК-11	Готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	<p>знать: основные шаблоны проектирования; методы оценки сложности алгоритмов.;</p> <p>уметь: проектировать основные модули программы и выбирать оптимальные алгоритмы;</p> <p>владеть (иметь навык(и)): оценки алгоритмической сложности и потребления памяти программы.</p>

**12. Объем дисциплины в зачетных единицах/час.**(в соответствии с учебным планом) — 6 / 216.

**Форма промежуточной аттестации** (зачет/экзамен) зачет (5 сем), экзамен (6 сем).

### 13. Виды учебной работы

Вид учебной работы	Трудоемкость		
	Всего	По семестрам	
		Сем. 5	Сем. 6
Аудиторные занятия	96	32	64
в том числе: лекции	32	0	32
семинары	16	16	0
лабораторные	48	16	32
Самостоятельная работа	84	49	35
Форма промежуточной аттестации (зачет – 0 час. / экзамен – 36час.)	36	зачет	экзамен
Итого:	216	81	135

#### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
<b>1. Семинарские занятия</b>		
1.1	Предварительные сведения	Первое знакомство с языком Java. Знакомство с интегрированной средой разработки IntelliJ Idea 14.1.1. Сравнение Java и других ООП языков
1.2	Знакомство со средой разработки и самим языком.	Написание первой программы на языке Java
1.3	Более детальное знакомство с языком Java и его ООП возможностями	Написание первого проекта, состоящего из нескольких классов, в Idea
1.4	Структуризация программы.	Структуризация программы. Знакомство с интерфейсом и внутренней частью
1.5	Знакомство с проектированием.	Проектирование задачи. Самостоятельное разделение на интерфейс, внутреннюю часть, классы и методы.
<b>2. Лекции</b>		
2.1	Основы Синтаксиса	Историческая справка. История создания языка Java. Краткий

		<p>обзор предыдущих версий.  Классификация платформ Java. Краткая терминология, принятая в Java.  Основы синтаксиса. Определение понятий объект, пакет, класс, метод, переменные экземпляра и переменные метода.  Правила именования сущностей в Java.  Правила написания комментариев в Java.  Методы. Структура метода. Определение возвращаемого типа, идентификатора метода, тела метода, параметров метода. Пример простой программы.  Переменные. Пример инициализации переменных.  Пакеты и организация Java-кода.  Примитивные типы и объектные типы. Изучение всех примитивных типов, их диапазона, размера в битах, особенностей хранения.  Правила приведения примитивных типов. Различия между примитивными и объектными типами.  Основные арифметические операции и особенности их использования в Java. Операторы сравнения. Логические операторы. Побитовые логические операции. Операторы битового сдвига. Операторы присваивания. Тернарный оператор. Операторы выбора.  Организация цикла в Java. Циклы while, do while, for. Примеры кода с этими циклами. Особенности использования различных циклов.</p>
2.2	Основы ООП	<p>Классы и объекты. Примеры кода. Примеры классов из стандартной библиотеки и их структура. Создание собственных классов.  Модификаторы видимости. Ключевые слова private, public, protected.  Инкапсуляция. Определение инкапсуляции. Пример кода. Хороший тон при оформлении программы.  Конструкторы. Особенности создания объектов в Java. Конструкторы по умолчанию. Вызов родительского конструктора, внутри дочернего конструктора. Инициализация полей при конструировании класса.  Перегрузка методов. Особенности работы перегрузки. Пример кода.  Наследование. Отношение предок-потомок. Плюсы и минусы наследования. Примеры кода. Особенности конструирования классов при наследовании.  Полиморфизм. Различия между полиморфизмом и наследованием. Плюсы и минусы использования.  Приведение объектных типов. Сравнение в приведении примитивных типов. Особенности и примеры.  Абстрактные классы. Плюсы использования абстрактных классов. Примеры кода.  Интерфейсы. Отличие интерфейсов от абстрактных классов. Отношения между объектами. Особенности композиции, агрегации, ассоциации и делегирования. Плюсы и минусы. Примеры.</p>
	Дополнительные возможности синтаксиса	<p>Модификатор static. Особенности использование статических методов и статических полей. Примеры. Статические блоки инициализации. Особенности применения.  Модификатор final. Особенности применения. Final поля, методы, классы. Одновременное использование static и final.  Базовый класс Object. Особенности наследование от базового родительского класса. Стандартные методы класса Object. Необходимость переопределения методов в дочерних классах.  Класс String. Особенности хранения и составления строк. Примитивный класс char и его обертка. Вычисление длины строки. Объединение строк. Форматирование строк. Вспомогательные классы StringBuilder и StringBuffer для формирования и преобразования строк. Основные методы класса String.  Массивы. Работы с массивами. Плюсы и минусы. Примеры. Методы с переменным количеством аргументов. Утилиты для работы с массивами. Двумерные массивы. Особенности</p>

		<p>реализации многомерных массивов в Java. Примеры. Перечисления. Удобство использование перечислений. Примеры создания своих перечислений. Методы перечислений. Оберточные типы. Необходимость использования оберточного типа для примитивных типов. Примеры.</p> <p>Внутренние классы. Особенности работы в внутренними классами. Примеры использования. Локальные внутренние классы. Анонимные классы. Вложенные классы. Вложенные интерфейсы. Исключения. Механизм обработки исключений. Создание собственных исключений. Иерархия и виды исключений.</p> <p>Параметризованные типы. Специфика параметризации в Java. Параметризованные классы. Параметризованные интерфейсы. Параметризованные методы. Примеры. Ограничения. Метасимволы. Примеры использования. Стирание типов.</p>
	Углубленное изучение	<p>Коллекции. Виды коллекций в Java. Интерфейс Collection. Списки: примеры и методы, особенности использования, реализация. Множества: примеры и методы, особенности использования, реализация. Сортировка коллекций. Ассоциативные массивы. Реализация ассоциативных массивов. Очередь и стек. Итераторы. Стандартные библиотеки для работы с числами. Библиотека Math. Примеры работы. Классы BigInteger и BigDecimal. Области применения, плюсы и минусы. Класс Random.</p> <p>Система ввода/вывода. Класс File: методы и специфика применения. Поточные классы, отвечающие за ввод и вывод. Чтение из файла. Запись в файл.</p> <p>Сериализация. Особенности и область применения. Рефлексия. Механизм рефлексии. Особенности применения. Аннотации. Примеры. Необходимость использования аннотации.</p> <p>4Шаблоны проектирования. Порождающие, структурные и поведенческие шаблоны. Примеры шаблонов.</p> <p>Сборщик мусора. Особенности сборки мусора в Java. Отличие в завершении работы по сравнению с другими языками программирования.</p> <p>Загрузка классов в Java. Специфика работы. Многопоточное программирование. Понятие процесса и потока. Методы потока. Синхронизация потоков. Атомарный доступ. Система сборки Maven. Независимость операционной системы. Управление зависимостями. Интеграция со средами разработки.</p>

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

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Семинарские	Лабораторные	Самостоятельная работа	Всего
1	Предварительные сведения		2	2	9	13
2	Знакомство со средой разработки и самим языком.		4	4	10	18
3	Более детальное знакомство с языком Java и его ООП возможностями		2	2	10	14
4	Структуризация программы.		4	4	10	18
5	Знакомство с проектированием.		4	4	10	18
6	Основы Синтаксиса	8		8	8	24
7	Основы ООП	8		8	8	24
8	Дополнительные возможности синтаксиса	8		8	8	24
9	Углубленное изучение	8		8	11	27
	Итого:	32	16	48	84	180

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

(рекомендации обучающимся по освоению дисциплины: работа с конспектами лекций, презентационным материалом, выполнение практических заданий, тестов, заданий текущей аттестации и т.д.)

Работа с конспектами лекций, выполнение практических заданий для самостоятельной работы, выполнение лабораторных работ, использование рекомендованной литературы. Выполнение контрольных работ.

#### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины (список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)

а) основная литература:

№ п/п	Источник
1	Эккель Б. Философия Java/ Эккель Б.; - 4 издание: - Питер: Классика Computer science, 2017 – 1168с.— ISBN 978-5-496-01127-3, 978-0131872486

б) дополнительная литература:

№ п/п	Источник
2	Шильдт Г. Java 8. Руководство для начинающих / Шильдт Г.; - Вильямс: Руководство для начинающих, 2018. – 720 с.-ISBN 978-5-8459-1955-7
3	Вирт Н. Алгоритмы и структуры данных / Вирт Н. ДМК Пресс: классика программирования, 2014. – 272 с. – ISBN 978-5-97060-230-0, 978-013022005-9
4	Решение комбинаторных задач на языке Java. Учебно-методическое пособие для вузов / И.В. Курбатова. – электр, 2018. – 41 с.

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

№ п/п	Ресурс
5	Электронный каталог Научной библиотеки Воронежского государственного университета. – <a href="http://www.lib.vsu.ru/">http://www.lib.vsu.ru/</a>
6	ЭБС «Издательство Лань» <a href="http://e.lanbook.com/">http://e.lanbook.com/</a>
7	Обучающий портал Java Tutorials <a href="https://docs.oracle.com/javase/tutorial/java/">https://docs.oracle.com/javase/tutorial/java/</a>

\* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы

#### 16. Перечень учебно-методического обеспечения для самостоятельной работы (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных) работ и др.)

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

ОС Windows, IntelliJ Idea 14.1.1 или выше

#### 18. Материально-техническое обеспечение дисциплины:

(при использовании лабораторного оборудования указывать полный перечень, при большом количестве оборудования можно вынести данный раздел в приложение к рабочей программе)

Семинарские занятия: аудитория, доска, проектор.

Лекционные занятия: аудитория, доска, проектор.

## 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 способность использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений	знать: основы объектно-ориентированного проектирования и программирования, основные структуры данных и алгоритмы, имеющиеся в стандартной библиотеке;	Раздел 1 Предварительные сведения	Практическое задание
	уметь: применять полученные знания для решения практических задач; пользоваться документацией к стандартной библиотеке и осуществлять поиск в ней; отлаживать программы в случае обнаружения нештатного поведения;	Раздел 2 Знакомство со средой разработки и самим языком.	Практическое задание
	владеть (иметь навык(и)): написания программ различного типа (вычислительных, сетевых и т.д.) с использованием различных типов и структур данных.	Раздел 3 Более детальное знакомство с языком Java и его ООП Возможностями	Практическое задание
ОПК-8 способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание ПО	знать: основные функции утилит JDK, а также интегрированной среды разработки;	Раздел 4 Структуризация программы.	Практическое задание
	уметь: выполнять запуск, отладку, сборку программы с использованием интегрированной среды разработки;	Раздел 5 Знакомство с проектированием.	Практическое задание
	владеть (иметь навык(и)): запуска, сборки и отладки программ различной сложности и размера.	Раздел 6 Основа Синтаксиса	Контрольная работа
ОПК-11 готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для	знать: основные шаблоны проектирования; методы оценки сложности алгоритмов.;	Раздел 7 Основа ООП	Практическое задание
	уметь: проектировать основные модули программы и выбирать оптимальные алгоритмы;	Раздел 8 Дополнительные возможности синтаксиса	Практическое задание
	владеть (иметь навык(и)): оценки алгоритмической сложности и потребления памяти программы.	Раздел 9 Углубленное изучение	Практическое задание

решения задач в различных предметных областях			Контрольная работа
<b>Промежуточная аттестация</b>			Зачет, экзамен

\* В графе «ФОС» в обязательном порядке перечисляются оценочные средства текущей и промежуточной аттестаций.

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

Для оценивания результатов обучения на зачете используются следующие показатели:

- 1) теоретические знания технологий и составляющих Java;
- 2) практические навыки выбора и использования технологий Java для реализации лабораторных задач.

Для оценивания результатов обучения на зачете с оценкой 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, правильные ответы на тестовые вопросы. Должны быть выполнены ВСЕ лабораторные работы.	<i>Повышенный уровень</i>	<i>Отлично</i>
Хорошее знание теоретического материала НО: допускает незначительные ошибки.	<i>Базовый уровень</i>	<i>Хорошо</i>
Недостаточное знание теоретического материала. ИЛИ: выполнены все показатели базового уровня, но не зачтена одна из лабораторных задач.	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
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 экзамену:

1. Краткая Java-терминология. Основы синтаксиса языка Java
2. Правила именования. Правила написания комментариев.
3. Методы. Структура методов. Переменные. Объявление переменной.
4. Пакеты и организация кода. Примитивные типы. Целочисленные типы. Числа с плавающей точкой. Примитивные и объектные типы. Приведение примитивных типов.
5. Основные операторы. Арифметические операторы. Операторы сравнения. Логические операторы. Побитовые логические операторы. Операторы побитового сдвига. Операторы присваивания. Тернарный оператор. Оператор выбора. Операторы цикла. Операторы перехода.
6. Классы и объекты. Модификаторы видимости. Инкапсуляция. Конструкторы. Конструктор по умолчанию. Инициализация полей.



7. Перегрузка.
8. Наследование. Особенности конструирования классов при наследовании.
9. Полиморфизм. Приведение объектных типов.
10. Абстрактные классы
11. Интерфейсы
12. Отношения. Композиция. Агрегация. Ассоциация. Делегирование.
13. Коллекции. Виды коллекций. Интерфейс Collection.
14. Списки. Интерфейс List. Реализации списков.
15. Множества. Интерфейс Set. Реализация множеств.
16. Сортировка коллекций.
17. Ассоциативные массивы. Интерфейс Map. Реализация ассоциативных массивов.
18. Очередь и стек. Интерфейсы Queue и Deque.
19. Итераторы
20. Стандартные библиотеки для работы с числами. Math и BigInteger,
21. Стандартные библиотеки для работы с числами. BigDecimal и Random.
22. Система ввода и вывода. Класс File.
23. Система ввода и вывода. Потоки ввода и вывода.
24. Система ввода и вывода. Чтение из файла.
25. Система ввода и вывода. Запись в файл.
26. Сериализация.
27. Рефлексия.
28. Аннотации.
29. Шаблоны проектирования.
30. Сборщик мусора.
31. Загрузка классов.
32. Многопоточное программирование.
33. Система сборки Maven.

### 19.3.2 Перечень заданий для лабораторной работы

**Задание 1** Дана строка J, где перечисляются типы «ценных» камней. В J гарантированно все буквы разные. Дана строка S – те, камни, что фактически есть у нас. Посчитать сколько у нас ценных камней. Пример входных данных: J=aA; S=aAAbbbb. Ответ 3.

**Задание 2** Собственными делителями числа называются все делители этого числа, кроме него самого. Например, собственными делителями числа 12 являются 1, 2, 3, 4 и 6. Сумма делителей числа 12 равна 16.

Оказывается, что сумма всех собственных делителей числа 220 равна 284, а сумма всех собственных делителей числа 284 равна 220. Числа 220 и 284 образуют цепочку из двух чисел. Такие числа как 220 и 284 называются парой дружественных чисел.

Возможны цепочки и большей длины. Например, начиная с числа 12496, образуется цепочка из 5 чисел:

12496 -> 14288 -> 15472 -> 14536 -> 14264 -> 12496 ...

Эта цепочка оканчивается тем же числом, которым она начиналась, ее называют цепочкой дружественных чисел. Известно, что с какого бы элемента мы не начали строить цепочку, в какой-то момент она «заиклится», но не обязательно на том элементе, с которого начиналось построение. Другими словами, хотя заикленная цепочка получится всегда, но не обязательно начальное число попадает в эту цепочку.

Найдите наименьший член самой длинной цепочки дружественных чисел, ни один элемент которой не превышает 1 000 000.

**Задание 3** Рассмотрим следующие последовательности чисел (слева - формула, справа - несколько первых членов):

Треугольные:  $T_n = n(n+1)/2$ ,  $T_n = 1, 3, 6, 10, 15, \dots$

Пятиугольные:  $P_n = n(3n-1)/2$ ,  $P_n = 1, 5, 12, 22, 35, \dots$

Шестиугольные:  $H_n = n(2n-1)$ ,  $H_n = 1, 6, 15, 28, 45, \dots$

Эти числа называют многоугольными, поскольку они равны числу целочисленных точек в соответствующем многоугольнике.

Легко проверяется, что

$$T_{285} = P_{165} = H_{143} = 40755.$$

Найдите следующее треугольное число, являющееся одновременно пятиугольным и шестиугольным.

**Задание 4.** Рассмотрим следующие последовательности чисел:

Треугольные:  $T_n = n(n+1)/2$ ,  $T_n = 1, 3, 6, 10, 15, \dots$

Квадратные:  $S_n = n^2$ ,  $S_n = 1, 4, 9, 16, 25, \dots$

Пятиугольные:  $P_n = n(3n-1)/2$ ,  $P_n = 1, 5, 12, 22, 35, \dots$

Шестиугольные:  $H_n = n(2n-1)$ ,  $H_n = 1, 6, 15, 28, 45, \dots$

Семиугольные:  $E_n = n(5n-3)/2$ ,  $E_n = 1, 7, 18, 34, 55, \dots$

Восьмиугольные:  $O_n = n(3n-2)$ ,  $O_n = 1, 8, 21, 40, 65, \dots$

Рассмотрим упорядоченное множество из трех четырехзначных чисел:

$$8128, 2882, 8281.$$

Оно обладает тремя интересными свойствами.

Множество является циклическим: последние две цифры каждого числа совпадают с первыми двумя цифрами следующего (это справедливо также для последнего и первого чисел).

Каждый тип многоугольника — треугольник ( $T_{127}=8128$ ), квадрат ( $S_{91}=8281$ ) и пятиугольник ( $P_{44}=2882$ ) - являются различными числами данного множества (но не по порядку, а именно, конец треугольного числа совпадает с началом пятиугольного, а не четырехугольного числа).

Рассмотренное нами множество - единственное множество из трех четырехзначных чисел, обладающее указанными свойствами.

Найдите единственное упорядоченное множество из шести циклических четырехзначных чисел, в котором каждый тип многоугольников — треугольник, квадрат, пятиугольник, шестиугольник, семиугольник и восьмиугольник — представлены различными числами этого множества.

**Задание 5** Дробь называется правильной, если ее числитель меньше знаменателя. Если же числитель больше знаменателя или равен ему, то дробь называется неправильной.

Например,  $2/3$ ,  $3/5$  и  $5/7$  - правильные дроби, а  $4/3$ ,  $8/5$  и  $9/5$  - неправильные дроби.

Приведенной называется дробь, которую невозможно сократить, т.е. наименьшее общее кратное числителя и знаменателя (GCD) равно единице.

Рассмотрим дробь  $n/d$ , где  $n$  и  $d$  - натуральные числа. Если  $n < d$  и  $GCD(n,d)=1$ , т.е. дробь является одновременно приведенной и правильной, то такая дробь называется приведенной правильной дробью.

Множество всех приведенных правильных дробей со знаменателем, не превышающем 8, выписанное в порядке возрастания, выглядит так:

$$1/8, 1/7, 1/6, 1/5, 1/4, 2/7, 1/3, 3/8, 2/5, 3/7, 1/2, 4/7, 3/5, 5/8, 2/3, 5/7, 3/4, 4/5, 5/6, 6/7, 7/8$$

Отметим, что здесь дробь  $2/5$  расположена непосредственно слева от дроби  $3/7$ .

Расположив множество всех приведенных правильных дробей, со знаменателем не превышающим 1 000 000, в порядке возрастания, найдите числитель дроби, расположенной непосредственно слева от  $3/7$ .

**Задание 6** Напомним, что простым называется число, у которого нет других делителей, кроме единицы и его самого.

Число 10 можно записать в виде суммы простых чисел ровно пятью различными способами:

$$10 = 7 + 3,$$

$$10 = 5 + 5,$$

$$10 = 5 + 3 + 2,$$

$$10 = 3 + 3 + 2 + 2,$$

$$10 = 2 + 2 + 2 + 2 + 2.$$

Какое наименьшее число можно записать в виде суммы простых чисел не меньше, чем пятью тысячами различными способами?

**Задание 7** Известно, что квадратный корень из натурального числа является или целым числом, или - иррациональным числом. Представление таких квадратных корней в виде десятичных дробей бесконечно и не имеет периода.

Например, корень из  $\sqrt{2}=1.41421356237309504880\dots$ , при этом сумма первых ста цифр в десятичном представлении равна 475.

Чему равна сумма первых ста цифр всех иррациональных квадратных корней среди первых ста натуральных чисел?

**Задание 8** На гранях двух кубиков нанесены разные неповторяющиеся цифры (от 0 до 9). Располагая два кубика рядом и поворачивая разными гранями вверх, можно получить различные двузначные числа.

Цифры обоих кубиков можно выбрать так, чтобы получить все двузначные квадраты: 01, 04, 09, 16, 25, 36, 49, 64 и 81.

Например, на грани одного кубика нанести цифры {0, 5, 6, 7, 8, 9}, а на грани второго - цифры {1, 2, 3, 4, 8, 9}.

Разрешается переворачивать 6 и 9, чтобы они заменяли друг друга.

Нас интересуют только цифры на гранях каждого из кубиков, а не их порядок следования.

Например, {1, 2, 3, 4, 5, 6} равносильно {3, 6, 4, 1, 2, 5}, а {1, 2, 3, 4, 5, 6} отличается от {1, 2, 3, 4, 5, 9}.

В то же время, оба множества из предыдущего примера эквивалентны по своим возможностям расширенному множеству {1, 2, 3, 4, 5, 6, 9} для получения двузначных чисел.

Сколько различных комбинаций нанесения цифр на кубики дают возможность получения всех квадратов?

#### 19.3.4 Перечень заданий для контрольных работ

**Задание 1** Известно, что квадратный корень из натурального числа является или целым числом, или - иррациональным числом. Представление таких квадратных корней в виде десятичных дробей бесконечно и не имеет периода.

Например, корень из  $\sqrt{2}=1.41421356237309504880\dots$ , при этом сумма первых ста цифр в десятичном представлении равна 475.

Чему равна сумма первых ста цифр всех иррациональных квадратных корней среди первых ста натуральных чисел?

**Задание 2** На гранях двух кубиков нанесены разные неповторяющиеся цифры (от 0 до 9). Располагая два кубика рядом и поворачивая разными гранями вверх, можно получить различные двузначные числа.

Цифры обоих кубиков можно выбрать так, чтобы получить все двузначные квадраты: 01, 04, 09, 16, 25, 36, 49, 64 и 81.

Например, на грани одного кубика нанести цифры {0, 5, 6, 7, 8, 9}, а на грани второго - цифры {1, 2, 3, 4, 8, 9}.

Разрешается переворачивать 6 и 9, чтобы они заменяли друг друга.

Нас интересуют только цифры на гранях каждого из кубиков, а не их порядок следования.

Например, {1, 2, 3, 4, 5, 6} равносильно {3, 6, 4, 1, 2, 5}, а {1, 2, 3, 4, 5, 6} отличается от {1, 2, 3, 4, 5, 9}.

В то же время, оба множества из предыдущего примера эквивалентны по своим возможностям расширенному множеству {1, 2, 3, 4, 5, 6, 9} для получения двузначных чисел.

Сколько различных комбинаций нанесения цифр на кубики дают возможность получения всех квадратов?

#### 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 теоретические вопросы, позволяющие оценить степень сформированности умений и навыков. Практические навыки определяются в ходе проверки выполнения лабораторных работ.

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