

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.
02.04.2024г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.В.04 Программирование на языке Java

1. Код и наименование направления подготовки/специальности:

02.03.03 Математическое обеспечение и администрирование информационных систем

2. Профиль подготовки/специализация:

Проектирование и разработка информационных систем

3. Квалификация (степень) выпускника:

Бакалавриат

4. Форма обучения:

Очная

5. Кафедра, отвечающая за реализацию дисциплины:

Кафедра программного обеспечения и администрирования информационных систем

6. Составители программы:

преп. Кондусов Н.И.

7. Рекомендована: НМС факультета ПММ, протокол №5 от 22.03.2024

Учебный год: 2024-2025

Семестр(ы)/Триместр(ы): 5, 6

9. Цели и задачи учебной дисциплины:

Цель изучения дисциплины: приобретение знаний и навыков в области разработки приложений на языке Java, достаточных для уровня трудоустройства на должность стажера/младшего разработчика.

Задачи учебной дисциплины: знакомство со структурой приложений в кроссплатформенных языках на примере Java, изучение структур данных языка, шаблонов, типов, принципов многопоточности и разделения ресурсов, основного индустриального фреймворка – Spring.

10. Место учебной дисциплины в структуре ООП:

Учебная дисциплина относится к формируемой участниками образовательных отношений части Блока 1.

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

Код и название компетенции	Код и название индикатора компетенции	Знания, умения, навыки
ОПК-3 Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	ОПК-3.1 Знает современные информационные технологии и применяет их при создании программных продуктов ОПК-3.2 Реализует различные новые программные продукты с использованием современных информационных технологий	знать: основы объектно-ориентированного проектирования и программирования, основные структуры данных и алгоритмы, имеющиеся в стандартной библиотеке; 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; владеть (иметь навык(и)): написания программ различного типа (вычислительных, сетевых и т.д.) с использованием различных типов и структур данных.

<p>ПКВ-5 Способен проектировать, разрабатывать и верифицировать программное обеспечение информационных систем</p>	<p>ПКВ-5.1. Знает методы и способы формализации и алгоритмизации поставленных задач, способы анализа требований к ПО, методологии проектирования ПО ПКВ-5.2. Способен создавать программный код с использованием языков программирования, определения и манипулирования данными и оформлять его в соответствии с установленными требованиями, проверять и отлаживать его ПКВ-5.3. Имеет практический опыт работы с системой контроля версий, разработки и рефакторинга программного кода.</p>	<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>
---	---	--

12. Объем дисциплины в зачетных единицах/час:

6/216

Форма промежуточной аттестации:

Зачет, Экзамен

13. Виды учебной работы

Вид учебной работы	Семестр 5	Семестр 6	Всего
Аудиторные занятия	48	48	96
Лекционные занятия	32	32	64
Практические занятия			0
Лабораторные занятия	16	16	32
Самостоятельная работа	42	42	84
Курсовая работа			0
Промежуточная аттестация	0	36	36
Часы на контроль		36	36
Всего	90	126	216

Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1	Основы Синтаксиса	<p>Историческая справка. История создания языка Java. Краткий обзор предыдущих версий.</p> <p>Классификация платформ Java. Краткая терминология, принятая в Java.</p> <p>Основы синтаксиса. Определение понятий объект, пакет, класс, метод, переменные экземпляра и переменные метода.</p> <p>Правила именования сущностей в Java.</p> <p>Правила написания комментариев в Java.</p> <p>Методы. Структура метода. Определение возвращаемого типа, идентификатора метода, тела метода, параметров метода. Пример простой программы.</p> <p>Переменные. Пример инициализации переменных.</p> <p>Пакеты и организация Java-кода.</p> <p>Примитивные типы и объектные типы. Изучение всех примитивных типов, их диапазона, размера в битах, особенностей хранения. Правила приведение примитивных типов. Различия между примитивными и объектными типами.</p> <p>Основные арифметические операции и особенности их использования в Java. Операторы сравнения.</p> <p>Логические операторы. Побитовые логические операции. Операторы битового сдвига. Операторы присваивание. Тернарный оператор. Операторы выбора.</p> <p>Организация цикла в Java. Циклы while, do while, for. Примеры кода с этими циклами. Особенности использования различных циклов.</p>	https://edu.vsu.ru/course/view.php?id=5172
п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
2	Основы ООП	<p>Классы и объекты. Примеры кода. Примеры классов из стандартной библиотеки и их структура.</p> <p>Создание собственных классов.</p> <p>Модификаторы видимости. Ключевые слова private, public, protected.</p> <p>Инкапсуляция. Определение инкапсуляции. Пример кода. Хороший тон при оформлении программы.</p> <p>Конструкторы. Особенности создания объектов в Java. Конструкторы по умолчанию. Вызов родительского конструктора, внутри дочернего конструктора. Инициализация полей при конструировании класса.</p> <p>Перегрузка методов. Особенности работы перегрузки. Пример кода.</p> <p>Наследование. Отношение предок-потомок. Плюсы и минусы наследования. Примеры кода. Особенности конструирования классов при наследовании.</p> <p>Полиморфизм. Различия между полиморфизмом и наследованием. Плюсы и минусы использования.</p> <p>Приведение объектных типов. Сравнение в приведением примитивных типов. Особенности и примеры.</p> <p>Абстрактные классы. Плюсы использования абстрактных классов. Примеры кода.</p> <p>Интерфейсы. Отличие интерфейсов от абстрактных классов.</p> <p>Отношения между объектами. Особенности композиции, агрегации, ассоциации и делегирование. Плюсы и минусы. Примеры.</p>	https://edu.vsu.ru/course/view.php?id=5172

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
3	Дополнительные возможности синтаксиса	<p>Модификатор <code>static</code>. Особенности использование статических методов и статических полей. Примеры. Статические блоки инициализации. Особенности применения.</p> <p>Модификатор <code>final</code>. Особенности применения. <code>Final</code> поля, методы, классы. Одновременное использование <code>static</code> и <code>final</code>.</p> <p>Базовый класс <code>Object</code>. Особенности наследование от базового родительского класса. Стандартные методы класса <code>Object</code>. Необходимость переопределения методов в дочерних классах.</p> <p>Класс <code>String</code>. Особенности хранения и составления строк. Примитивный класс <code>char</code> и его обертка.</p> <p>Вычисление длины строки. Объединение строк. Форматирование строк. Вспомогательные классы <code>StringBuilder</code> и <code>StringBuffer</code> для формирования и преобразования строк. Основные методы класса <code>String</code>.</p> <p>Массивы. Работы с массивами. Плюсы и минусы. Примеры. Методы с переменным количеством аргументов. Утилиты для работы с массивами.</p> <p>Двумерные массивы. Особенности реализации многомерных массивов в <code>Java</code>. Примеры.</p> <p>Перечисления. Удобство использование перечислений. Примеры создания своих перечислений. Методы перечислений.</p> <p>Оберточные типы. Необходимость использования оберточного типа для примитивных типов. Примеры.</p> <p>Внутренние классы. Особенности работы в внутренними классами. Примеры использования.</p> <p>Локальные внутренние классы. Анонимные классы.</p> <p>Вложенные классы. Вложенные интерфейсы.</p> <p>Исключения. Механизм обработки исключений. Создание собственных исключений. Иерархия и виды исключений.</p> <p>Параметризованные типы. Специфика параметризации в <code>Java</code>. Параметризованные классы.</p> <p>Параметризованные интерфейсы.</p> <p>Параметризованные методы. Примеры.</p> <p>Ограничения.</p> <p>Метасимволы. Примеры использования.</p> <p>Стирание типов.</p>	https://edu.vsu.ru/course/view.php?id=5172

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
4	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.</p> <p>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. Поточковые классы, отвечающие за ввод и вывод. Чтение из файла. Запись в файл.</p> <p>Сериализация. Особенности и область применения. Рефлексия. Механизм рефлексии. Особенности применения.</p> <p>Аннотации. Примеры. Необходимость использования аннотации.</p> <p>Шаблоны 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.</p> <p>Система сборки Maven. Независимость операционной системы. Управление зависимостями. Интеграция со средами разработки. Фреймворк Spring. Tomcat. Servlet api.</p>	https://edu.vsu.ru/course/view.php?id=5172

13.1. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела)	Лекционные занятия	Практические занятия	Лабораторные занятия	Самостоятельная работа	Всего
1	Основы Синтаксиса	16		8	22	46
2	Основы ООП	16		8	20	44
3	Дополнительные возможности синтаксиса	16		8	22	46
4	Углубленное изучение	16		8	20	44
		64	0	32	84	180

14. Методические указания для обучающихся по освоению дисциплины

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

№ 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
1	Шильдт Г. Java 8. Руководство для начинающих / Шильдт Г.; - Вильямс: Руководство для начинающих, 2018. - 720 с.-ISBN 978-5-8459-1955-7
2	Курбатова, Ирина Витальевна. Язык программирования JAVA : учебное пособие. Ч. 1. Основы синтаксиса и его применение в объектно-ориентированном программировании / И. В. Курбатова, А. В. Печкуров .— Воронеж : Издательский дом ВГУ, 2019 .— 78 с. — Тираж 50. 4,9 п.л. — ISBN 978-5-9273-2790-1.
3	Курбатова, Ирина Витальевна. Язык программирования JAVA : учебное пособие. Ч. 2. Специальные возможности синтаксиса / И. В. Курбатова, А. В. Печкуров .— Воронеж : Издательский дом ВГУ, 2019 .— 92 с. — Тираж 50. 5,8 п.л. — ISBN 978-5-9273-2792-8.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
1	Электронный каталог Научной библиотеки Воронежского государственного университета. - http://www.lib.vsu.ru/
2	ЭБС «Издательство Лань» http://e.lanbook.com/
3	Обучающий портал Java Tutorials https://docs.oracle.com/javase/tutorial/java/

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Решение комбинаторных задач на языке Java. Учебно-методическое пособие для вузов / И.В. Курбатова. - электр, 2018. - 41 с.

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

IntelliJ Idea IDE

Материально-техническое обеспечение дисциплины:

Учебная мультимедийная аудитория (ауд. 226): доска меловая, доска маркерная, компьютер (монитор NEC, системный блок Intel Pentium Dual CPU E 2160@ 1.80 GHz, ОЗУ 1 Гб), мультимедиа-проектор Acer, источник бесперебойного питания Ippon Smart Power Pro 1000, экран для проектора Draper Star Projection Screen

Лабораторный класс с проектором (ауд 20): компьютеры (мониторы Acer V226HQL, системные блоки Intel Core i3-4160 CPU@ 3.60GHz, ОЗУ 4 Гб) (14 шт.), мультимедиа-проектор Acer, коммутатор HP ProCurve Switch 1400-24G

Аудитория для самостоятельной работы:

Лабораторный класс с проектором (ауд 15): доска маркерная, мультимедиа-проектор Acer, терминальные рабочие станции SunRay 2 (мониторы и терминалы) (15 шт.)

18. Оценочные средства для проведения текущей и промежуточной аттестаций

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

№ п/п	Разделы дисциплины (модули)	Код компетенции	Код индикатора	Оценочные средства для текущей аттестации
1	1-4	ОПК-3	ОПК-3.1, ОПК-3.2	КИМ
2	1-4	ПКВ-5	ПКВ-5.1, ПКВ-5.2, ПКВ-5.3.	КИМ

Промежуточная аттестация

Форма контроля - Зачет, Экзамен

Оценочные средства для промежуточной аттестации

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
свободно владеет материалом, отвечает на вопросы; умеет рассуждать; в случае незнания небольшой части материала способен выстроить собственную логическую цепочку рассуждений и получить ответ	<i>Повышенный уровень</i>	<i>Отлично</i>
знает материал, допускает неточности, не полностью отвечает на вопросы;	<i>Базовый уровень</i>	<i>Хорошо</i>
знание минимально допустимого объема материала; нежелание рассуждать;	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
незнание материала	-	<i>Неудовлетворительно</i>

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

контрольная работа

20.2 Промежуточная аттестация

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.
34. Паттерн инъекция зависимостей.
35. Servlet - контейнер tomcat
36. Фреймворк Spring.

Перечень заданий для лабораторной работы

Задание 1 Посчитать \cos/\sin через ряд тейлора двумя способами – а) с использованием фиксированного количества итераций через цикл `for` б) С определенной точностью с использованием цикла `while`.

Задание 2 Спроектировать систему классов с использованием базовых принципов ооп (наследование, полиморфизм, инкапсуляцию). Переопределить методы `toString`, `hashCode`, `equals`. Продемонстрировать работу сравнение объектов и полиморфных методов, вывести результаты в консоль.

Задание 3

Написать тесты производительности коллекций для следующих случаев

- Добавление элементов в `ArrayList`, `LinkedList`, `HashSet`, `TreeSet`
- Добавление по фиксированному индексу (2 теста, в начало и в конец) в `ArrayList`, `LinkedList`.
- Заполнить `HashSet` и `TreeSet` целыми числами. Проверить разницу в скорости по методу `contains` (вызывать контейнс в цикле, а не один раз)
- Сравнить скорость `contains` для `HashSet`, `ArrayList`, `LinkedList`
- Написать комментарий, почему отличается производительность в том или ином случае

Задание 4.

Дан файл со следующей структурой

1563328060

КОНДУСОВ НИКОЛАЙ ИГОРЕВИЧ;28;+79191987421;MALE;Воронеж, Площадь ленина 3;

ПЕНЯЕВ СЕРГЕЙ ПАВЛОВИЧ;12;+79191987421;MALE;Самара, улица Космоновтов 2;

Анна Евгеньевна;58;+22321321;FEMALE;Ул Остужева;

Первая строка - контрольная сумма файла.

Далее идут строки со следующей структурой -

ФИО(Строка);Возраст(Целое число);Телефон(Строка);ПОЛ(MALE или FEMALE);Адрес;

В строке с помощью символа ";" отделены поля. Все поля обязательные

Контрольная сумма - результат работы хеш функции над содержимым файла (в примере - просто `hashCode`). Вы можете выбрать хеш функцию самостоятельно

Программа при старте должна запускать консольный интерфейс и предоставлять следующие функции

`LOADFILE`, запрашивает имя файла на вход, загружает файл в память программы и проверить контрольную сумму. Пример:

```
-----  
LOADFILE  
"Введите путь к файлу"  
C:/test.dat  
"Файл успешно загружен."  
-----
```

При загрузке файла могут произойти следующие ошибки:

1) Если совпала хеш сумма - логируем ошибку, не открываем файл. Для обработки используем

исключение.

2) Формат файла поврежден - (например нет какого либо поля) - логируем ошибку, не открываем файл. Для обработки используем исключение.

3) Не получилось прочитать файл или он не найден - логируем ошибку. Для обработки используем исключение.

4) Повторяющееся ФИО. логируем ошибку. Для обработки НЕ используем исключение. Используем последнее найденное значение.

SEARCH, ищет пользователя по ФИО, не чувствительно к регистру, выводит информацию о нем.
Пример:

SEARCH

"Введите ФИО"

КондУсОв НИКОЛАЙ ИГоревич

Найден пользователь:

ФИО='КОНДУСОВ НИКОЛАЙ ИГОРЕВИЧ', ВОЗРАСТ=28, ТЕЛЕФОН='+79191987421', ПОЛ=MALE,
АДРЕС='Воронеж, Площадь ленина 3'

Возможные ошибки

1) Пользователь не найден - пишем в консоль соответствующее сообщение. Логируем ошибку. Для обработки НЕ используем исключение.

ADDUSER - Добавляет новую строку в данные в памяти. Пример

ADDUSER

"Введите имя"

Василий Теркин

"Введите возраст"

18

"Введите телефон"

+7111111111

"Введите пол: MALE или FEMALE"

MALE

"Введите адрес"

Улица Танкистов

"Пользователь создан"

Возможные ошибки

1) Пользователь с таким именем уже существует - логируем ошибку, пишем сообщение в консоль, не перезаписываем пользователя. Не используем исключение.

2) Неверный формат данных (например возраст < 0 или не введено поле) - логируем ошибку. Для обработки используем исключение. Поля не могут содержать символ ";"

REMOVEUSER- Удаляет строку из данные в памяти. Пример

REMOVEUSER

"Введите имя"

Василий Теркин

Возможные ошибки

1) Пользователя с таким именем не существует - логируем ошибку, пишем сообщение в консоль. Не используем исключение.

SAVEFILE - перезаписывает исходный файл, обновляет содержимое и контрольную сумму. Пример:

```
-----  
SAVEFILE  
"Файл успешно сохранен"  
-----
```

SAVEFILEAS -сохраняет файл с новым именем, обновляет содержимое и контрольную сумму. Пример:

```
-----  
SAVEFILEAS  
"Введите путь к файлу"  
C:/test22.dat  
"Файл успешно загружен."  
-----
```

Возможные ошибки

1) Не получилось прочитать файл - логируем ошибку. Для обработки используем исключение.

EXIT - выход из программы.

При попытке пользователя ввести несуществующую команду, например TESTUSER - генерируем исключение, обрабатываем, увеличиваем счетчик неверных команд.

Если неверная команда была введена больше 3х раз - прекращаем работу программы.

Задание 5.

Реализовать паттерн produce/consumer двумя способами 1) wait/notify 2) с использованием коллекций из java.util.concurrent

Задание 6 Реализовать приложение с использованием servlet api, jdbc, jsp, реализующее регистрацию пользователя, аутентификацию, возможность добавлять/удалять/редактировать выбранные студентом сущности (напр. список домашних животных).

Задание 7.

Реализовать приложение с использованием spring framework, реализующее возможность добавлять/удалять/редактировать выбранные студентом сущности (напр. список домашних животных).

Управление сущностями должно быть реализовано по концепциям Rest, в качестве интерфейса использовать swagger. Использовать Spring Security для разграничение доступа

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?

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.