

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Баламирзоев Назим Лиодинович  
Должность: Ректор  
Дата подписания: 06.04.2026 13:18:12  
Уникальный программный ключ:  
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926

Приложение А

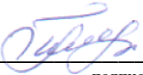
(обязательное к рабочей программе дисциплины)

Министерство науки и высшего образования Российской Федерации  
ФГБОУ ВО «Дагестанский государственный технический университет»


## ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Введение в программирование и алгоритмы»

Уровень образования	<u>специалитет</u> (бакалавриат/магистратура/специалитет)
Специальность	<u>10.05.03 Информационная безопасность автоматизированных систем</u> (код, наименование специальности)
Специализация	<u>Безопасность открытых информационных систем</u> (наименование)

Разработчик  Качаева Г.И.  
подпись (ФИО уч. степень, уч. звание)

Фонд оценочных средств обсужден на заседании кафедры ИБиПИ «15» октября 2025г., протокол № 2

Зав. кафедрой  Качаева Г.И.  
подпись (ФИО уч. степень, уч. звание)

г. Махачкала 2025

## СОДЕРЖАНИЕ

1. Область применения, цели и задачи фонда оценочных средств .....	3
2. Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля) .....	3
2.1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП .....	4
2.1.2. Этапы формирования компетенций .....	6
2.2. Показатели уровней сформированности компетенций на этапах их формирования, описание шкал оценивания .....	8
2.2.1. Показатели уровней сформированности компетенций на этапах их формирования	8
2.2.2. Описание шкал оценивания .....	10
3. Типовые контрольные задания, иные материалы и методические рекомендации, необходимые для оценки сформированности компетенций в процессе освоения ОПОП .....	11
3.1. Задания и вопросы для входного контроля .....	11
3.2. Оценочные средства и критерии сформированности компетенций .....	11
3.2.1. Аттестационная контрольная работа №1 .....	11
3.2.2. Аттестационная контрольная работа №2 .....	12
3.2.3. Аттестационная контрольная работа №3 .....	12
3.2.4. Список вопросов к зачету .....	13

## **1. Область применения, цели и задачи фонда оценочных средств**

Фонд оценочных средств (ФОС) является неотъемлемой частью рабочей программы дисциплины «Введение в программирование и алгоритмы» и предназначен для контроля и оценки образовательных достижений обучающихся (в т.ч. по самостоятельной работе студентов, далее – СРС), освоивших программу данной дисциплины.

Целью фонда оценочных средств является установление соответствия уровня подготовки обучающихся требованиям ФГОС ВО по специальности 10.05.03 Информационная безопасность автоматизированных систем.

Рабочей программой дисциплины «Введение в программирование и алгоритмы» предусмотрено формирование следующих компетенций:

ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности.

## **2. Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля)**

Описание показателей и критериев оценивания компетенций, формируемых в процессе освоения дисциплины (модуля), и используемые оценочные средства приведены в таблице 1.

*Перечень оценочных средств, рекомендуемых для заполнения таблицы 1 (в ФОС не приводится, используется только для заполнения таблицы)*

- *Эссе*
- *Устный опрос*
- *Вопросы для проведения зачета*

## 2.1. Перечень компетенций с указанием этапов их формирования в процессе освоения ОПОП

Таблица 1

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Критерии оценивания	Наименование контролируемых разделов и тем <sup>1</sup>
ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности	ОПК-2.1.1 знает состав, назначение функциональных компонентов и программного обеспечения персонального компьютера	<p>Знать: Состав и архитектуру вычислительной системы, включая процессор, память (иерархия, кэш), устройства ввода-вывода; принципы организации памяти (стек, куча, сегментация); классификацию и назначение программного обеспечения для разработки и анализа алгоритмов (компиляторы, интерпретаторы, IDE, профилировщики, отладчики); влияние аппаратных характеристик на производительность алгоритмов.</p> <p>Уметь: Оценивать потребление памяти и времени работы алгоритма с учётом особенностей аппаратного обеспечения (размер кэш-линии, частота процессора); выбирать оптимальные алгоритмы и структуры данных, исходя из доступных вычислительных ресурсов; использовать системные утилиты для мониторинга ресурсов при выполнении программ.</p> <p>Владеть: Навыками использования инструментов профилирования (cProfile, timeit) для анализа эффективности программ; методами оптимизации кода с учётом архитектуры компьютера; приёмами работы с низкоуровневыми аспектами языка Python для управления памятью</p>	№№ 1-17
	ОПК-2.1.2 знает формы и способы представления данных в персональном компьютере	<p>Знать: Способы представления основных структур данных в памяти компьютера (массивы, связанные списки, деревья, хэш-таблицы, графы); форматы хранения данных (битовые поля, записи, сериализация); понятие сложности по памяти; методы компактного хранения (сжатие, битовые карты); внутреннее представление базовых типов данных в Python.</p>	№№ 1-17

<sup>1</sup> Наименования разделов и тем должен соответствовать рабочей программе дисциплины.

		<p>Уметь: Реализовывать структуры данных на языке Python с учётом их внутреннего представления и эффективного использования памяти; анализировать потребление памяти различными структурами для конкретной задачи; выбирать структуру данных, обеспечивающую требуемую производительность при ограничениях по памяти.</p> <p>Владеть: Навыками реализации и использования структур данных (стек, очередь, дек, куча, деревья поиска, дерево отрезков, хэш-таблицы) в программном коде; методами оценки памяти, занимаемой структурами, с помощью встроенных средств (sys.getsizeof) и профилировщиков.</p>	
	<p><b>ОПК-2.2.1</b> умеет применять типовые программные средства сервисного назначения и пользоваться сетевыми средствами для обмена данными, в том числе с использованием глобальной информационной сети Интернет</p>	<p>Знать: Типовые программные средства для разработки, отладки и тестирования алгоритмов (IDE: PyCharm, VS Code; системы контроля версий: Git; профилировщики, отладчики); сетевые протоколы и форматы для обмена данными (HTTP, JSON, XML); возможности использования интернет-ресурсов (официальная документация, репозитории кода, форумы, Stack Overflow) для решения профессиональных задач.</p> <p>Уметь: Применять интегрированные среды разработки для написания, отладки и тестирования программ; использовать системы контроля версий для совместной работы и управления версиями кода; применять сетевые средства Python (модули requests, urllib, socket) для получения данных из интернета, взаимодействия с API, обмена данными между приложениями.</p> <p>Владеть: Навыками работы с Git (создание репозитория, коммиты, ветвление, слияние); методами поиска и использования сторонних библиотек и фреймворков для решения алгоритмических задач (NumPy, SciPy, стандартная библиотека); приёмами отладки (pdb, встроенный отладчик IDE) и профилирования кода с использованием встроенных и внешних инструментов.</p>	<p>№№ 1-17</p>

## 2.1.2. Этапы формирования компетенций

Сформированность компетенций по дисциплине «Введение в программирование и алгоритмы» определяется на следующих этапах:

1. **Этап текущих аттестаций** (Для проведения текущих аттестаций могут быть использованы оценочные средства, указанные в разделе 2)

2. **Этап промежуточных аттестаций** (Для проведения промежуточной аттестации могут быть использованы другие оценочные средства)

Таблица 2

Код и наименование формируемой компетенции	Код и наименование индикатора достижения формируемой компетенции	Этапы формирования компетенции					Этап промежуточной аттестации
		Этап текущих аттестаций					
		1-5 неделя	6-10 неделя	11-15 неделя	1-17 неделя		
		Текущая аттестация №1	Текущая аттестация №2	Текущая аттестация №3	СРС	КР/КП	
1		2	3	4	5	6	7
ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности	ОПК-2.1.1 знает состав, назначение функциональных компонентов и программного обеспечения персонального компьютера	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета
	ОПК-2.1.2 знает формы и способы представления данных в персональном компьютере	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета
	ОПК-2.2.1 умеет применять типовые программные средства сервисного назначения и пользоваться сетевыми средствами для обмена	Контрольная работа №1	Контрольная работа №2	Контрольная работа №3			Вопросы для проведения зачета

	данными, в том числе с использованием глобальной информационной сети Интернет						
--	---	--	--	--	--	--	--

**СРС** – самостоятельная работа студентов;

**КР** – курсовая работа;

**КП** – курсовой проект.

## 2.2. Показатели уровней сформированности компетенций на этапах их формирования, описание шкал оценивания

### 2.2.1. Показатели уровней сформированности компетенций на этапах их формирования

Результатом освоения дисциплины «Введение в программирование и алгоритмы» является установление одного из уровней сформированности компетенций: высокий, повышенный, базовый, низкий.

Таблица 3

Уровень	Универсальные компетенции	Общепрофессиональные/ профессиональные компетенции
Высокий (оценка «отлично», «зачтено»)	Сформированы четкие системные знания и представления по дисциплине. Ответы на вопросы оценочных средств полные и верные. Даны развернутые ответы на дополнительные вопросы. Обучающимся продемонстрирован высокий уровень освоения компетенции	Обучающимся усвоена взаимосвязь основных понятий дисциплины, в том числе для решения профессиональных задач. Ответы на вопросы оценочных средств самостоятельны, исчерпывающие, содержание вопроса/задания оценочного средства раскрыто полно, профессионально, грамотно. Даны ответы на дополнительные вопросы. Обучающимся продемонстрирован высокий уровень освоения компетенции
Повышенный (оценка «хорошо», «зачтено»)	Знания и представления по дисциплине сформированы на повышенном уровне. В ответах на вопросы/задания оценочных средств изложено понимание вопроса, дано достаточно подробное описание ответа, приведены и раскрыты в тезисной форме основные понятия. Ответ отражает полное знание материала, а также наличие, с незначительными пробелами, умений и навыков по изучаемой дисциплине. Допустимы единичные негрубые ошибки. Обучающимся продемонстрирован повышенный уровень освоения компетенции	Сформированы в целом системные знания и представления по дисциплине. Ответы на вопросы оценочных средств полные, грамотные. Продемонстрирован повышенный уровень владения практическими умениями и навыками. Допустимы единичные негрубые ошибки по ходу ответа, в применении умений и навыков
Базовый (оценка «удовлетворительно», «зачтено»)	Ответ отражает теоретические знания основного материала дисциплины в объеме, необходимом для дальнейшего освоения ОПОП. Обучающийся допускает неточности в ответе, но обладает необходимыми знаниями для их	Обучающийся владеет знаниями основного материал на базовом уровне. Ответы на вопросы оценочных средств неполные, допущены существенные ошибки. Продемонстрирован базовый уровень владения

Уровень	Универсальные компетенции	Общепрофессиональные/ профессиональные компетенции
	устранения. Обучающимся продемонстрирован базовый уровень освоения компетенции	практическими умениями и навыками, соответствующий минимально необходимому уровню для решения профессиональных задач
Низкий (оценка «неудовлетворительно», «не зачтено»)	Демонстрирует полное отсутствие теоретических знаний материала дисциплины, отсутствие практических умений и навыков	

Показатели уровней сформированности компетенций могут быть изменены, дополнены и адаптированы к конкретной рабочей программе дисциплины.

## 2.2.2. Описание шкал оценивания

В ФГБОУ ВО «ДГТУ» внедрена модульно-рейтинговая система оценки учебной деятельности студентов. В соответствии с этой системой применяются пятибалльная, двадцатибалльная и стобальная шкалы знаний, умений, навыков.

Шкалы оценивания			Критерии оценивания
пятибалльная	двадцатибалльная	стобальная	
«Отлично» - 5 баллов	«Отлично» - 18-20 баллов	«Отлично» - 85 – 100 баллов	Показывает высокий уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> <li>– продемонстрирует глубокое и прочное усвоение материала;</li> <li>– исчерпывающе, четко, последовательно, грамотно и логически стройно излагает теоретический материал;</li> <li>– правильно формирует определения;</li> <li>– демонстрирует умения самостоятельной работы с нормативно-правовой литературой;</li> <li>– умеет делать выводы по излагаемому материалу.</li> </ul>
«Хорошо» - 4 баллов	«Хорошо» - 15 - 17 баллов	«Хорошо» - 70 - 84 баллов	Показывает достаточный уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> <li>– демонстрирует достаточно полное знание материала, основных теоретических положений;</li> <li>– достаточно последовательно, грамотно логически стройно излагает материал;</li> <li>– демонстрирует умения ориентироваться в нормальной литературе;</li> <li>– умеет делать достаточно обоснованные выводы по излагаемому материалу.</li> </ul>
«Удовлетворительно» - 3 баллов	«Удовлетворительно» - 12 - 14 баллов	«Удовлетворительно» - 56 – 69 баллов	Показывает пороговый уровень сформированности компетенций, т.е.: <ul style="list-style-type: none"> <li>– демонстрирует общее знание изучаемого материала;</li> <li>– испытывает серьезные затруднения при ответах на дополнительные вопросы;</li> <li>– знает основную рекомендуемую литературу;</li> <li>– умеет строить ответ в соответствии со структурой излагаемого материала.</li> </ul>
«Неудовлетворительно» - 2 баллов	«Неудовлетворительно» - 1-11 баллов	«Неудовлетворительно» - 1-55 баллов	Ставится в случае: <ul style="list-style-type: none"> <li>– незнания значительной части программного материала;</li> <li>– не владения понятийным аппаратом дисциплины;</li> <li>– допущения существенных ошибок при изложении учебного материала;</li> <li>– неумение строить ответ в соответствии со структурой излагаемого вопроса;</li> <li>– неумение делать выводы по излагаемому материалу.</li> </ul>

### **3. Типовые контрольные задания, иные материалы и методические рекомендации, необходимые для оценки сформированности компетенций в процессе освоения ОПОП**

#### **3.1. Задания и вопросы для входного контроля**

1. Дайте определение алгоритма. Перечислите основные свойства алгоритмических конструкций.
2. Какие встроенные типы данных реализованы в Python? В чём отличие списка от кортежа?
3. Опишите синтаксис условного оператора if-elif-else. Приведите пример с несколькими условиями.
4. Напишите фрагмент кода, который с помощью цикла for выводит все чётные числа от 1 до 20.
5. Что такое функция в программировании? Как объявить функцию в Python и вернуть из неё значение?
6. Перечислите основные методы для работы со строками (разделение, соединение, замена, поиск).
7. Как выполнить чтение данных из текстового файла и запись в файл? Приведите пример с менеджером контекста with.
8. Что понимают под сложностью алгоритма? Чем характеризуются временная и ёмкостная сложности?
9. Объясните понятие рекурсии. Напишите рекурсивную функцию для вычисления факториала числа.
10. Какие структуры данных (контейнеры) доступны в Python «из коробки»? Для каких задач используется словарь?

#### **3.2. Оценочные средства и критерии сформированности компетенций**

##### **3.2.1. Аттестационная контрольная работа №1**

1. Дайте определения  $O$ -большого,  $o$ -малого,  $\Omega$ -большого,  $\omega$ -малого,  $\Theta$ -большого. Приведите примеры.
2. Сформулируйте мастер-теорему для решения рекуррентных соотношений вида  $T(n) = a \cdot T(n/b) + f(n)$ . Для каких случаев она применима?
3. Примените мастер-теорему к рекурренте  $T(n) = 2T(n/2) + O(n)$ . Какова асимптотика?
4. Реализуйте на Python класс Stack с использованием списка. Добавьте методы push, pop, peek, is\_empty.
5. Реализуйте очередь на Python, используя два стека (или список с кольцевым буфером). Оцените асимптотику операций.
6. Что такое дек (deque)? Какие операции он поддерживает? Приведите пример использования из модуля collections.
7. Опишите алгоритм поиска ближайшего большего справа для всех элементов массива за  $O(n)$ . Реализуйте его на Python.
8. Как организовать поиск минимума в стеке за  $O(1)$  дополнительно к основным операциям? Приведите реализацию.
9. Та же задача для очереди: как поддерживать минимум в очереди при добавлении и удалении элементов?
10. Что такое вектор (динамический массив)? Объясните метод бухгалтерского учёта для доказательства амортизированной сложности вставки в конец.
11. Оцените амортизированную стоимость последовательности из  $n$  операций вставки в динамический массив, если при переполнении размер увеличивается вдвое.
12. Напишите функцию, которая проверяет правильность скобочной последовательности с использованием стека.
13. Реализуйте алгоритм сортировки стека с использованием дополнительного стека.

14. В чём преимущества и недостатки реализации стека на основе списка Python?
15. Для массива целых чисел найдите все пары  $(i, j)$  такие, что  $i < j$  и  $a[i] < a[j]$  (по сути, подсчёт числа таких пар). Предложите эффективный алгоритм.

### 3.2.2. Аттестационная контрольная работа №2

1. Что такое стабильная сортировка? Приведите примеры стабильных и нестабильных алгоритмов.
2. Реализуйте сортировку слиянием (merge sort) на Python. Оцените её временную и пространственную сложность.
3. Как с помощью сортировки слиянием подсчитать число инверсий в массиве? Напишите код.
4. Опишите алгоритм быстрой сортировки (quick sort). Как выбор опорного элемента влияет на производительность?
5. Реализуйте детерминированный алгоритм поиска  $k$ -й порядковой статистики ( $k$ -го наименьшего элемента) с линейным ожидаемым временем.
6. Что такое сортировка подсчётом (counting sort)? В каких случаях она эффективна? Приведите пример.
7. Опишите алгоритм поразрядной сортировки (LSD) для целых чисел. Какая требуется дополнительная память?
8. Дайте определение кучи (heap). Какие основные операции поддерживает куча и какова их сложность?
9. Реализуйте двоичную кучу на Python с операциями sift\_up и sift\_down. Покажите, как выполнить извлечение минимума.
10. Что такое биномиальная куча? В чём её преимущества перед двоичной кучей?
11. Какие идеи лежат в основе Фибоначчиевой кучи? Как с помощью метода бухгалтерского учёта достигается амортизированная сложность операций?
12. Напишите функцию heapify, которая преобразует список в кучу за линейное время.
13. Используя кучу, реализуйте алгоритм сортировки (heapsort). Оцените его сложность.
14. Сравните производительность быстрой сортировки и сортировки кучей при разных входных данных.
15. Предложите алгоритм нахождения  $k$  наименьших элементов в массиве на основе кучи за  $O(n \log k)$ .

### 3.2.3. Аттестационная контрольная работа №3

1. Дайте определение дерева поиска. Какие операции должны поддерживаться и какова их сложность в сбалансированном дереве?
2. Опишите принцип балансировки в AVL-дереве. Что такое фактор баланса и когда выполняются повороты?
3. В чём особенность splay-деревя? Как обеспечивается амортизированная логарифмическая сложность?
4. Что такое декартово дерево (treap)? Какие свойства оно сочетает? Реализуйте операции split и merge.
5. Для чего используются B-деревья? Где они применяются на практике?
6. Опишите структуру дерева отрезков. Какие запросы (например, сумма на отрезке, минимум) можно выполнять за  $O(\log n)$ ? Реализуйте построение и запрос суммы.
7. Что такое отложенные операции (lazy propagation) в дереве отрезков? Когда они необходимы?
8. Опишите дерево Фенвика (двоичное индексированное дерево). Как вычислить сумму на префиксе и обновить элемент?
9. Сравните дерево отрезков и дерево Фенвика по сложности и применимости.
10. Что такое хэш-таблица? Опишите методы разрешения коллизий: цепочки и открытая адресация.

11. Что такое универсальное семейство хэш-функций? Как выбрать хорошую хэш-функцию?
12. Оцените вероятность коллизии при хэшировании. Как влияет размер таблицы и количество элементов?
13. Что такое фильтр Блума? Какие операции он поддерживает и какова вероятность ложноположительного срабатывания?
14. Реализуйте простой фильтр Блума на Python с использованием нескольких хэш-функций (можно использовать модуль hashlib).
15. Приведите примеры практического применения фильтров Блума

### 3.2.4. Список вопросов к зачету

1. Понятие асимптотической сложности. O-нотация,  $\Omega$ -нотация,  $\Theta$ -нотация. Примеры.
2. Мастер-теорема: формулировка и примеры применения.
3. Линейные структуры данных: стек, очередь, дек. Реализация на Python и анализ операций.
4. Поиск ближайшего большего справа в массиве за  $O(n)$ . Алгоритм и реализация.
5. Поддержка минимума в стеке и очереди. Идеи и реализация.
6. Динамический массив (вектор). Амортизированный анализ вставки.
7. Сортировка слиянием: алгоритм, сложность, стабильность, подсчёт инверсий.
8. Быстрая сортировка: алгоритм, выбор опорного элемента, анализ сложности в среднем и худшем случаях.
9. Сортировка подсчётом и поразрядная сортировка: условия применимости, сложность.
10. Поиск k-й порядковой статистики. Алгоритм с линейным ожидаемым временем.
11. Двоичная куча: определение, основные операции (sift\_up, sift\_down), построение кучи.
12. Сортировка кучей (heapsort). Реализация и анализ.
13. Биномиальные кучи: структура, операции, асимптотика.
14. Фибоначчиевы кучи: ключевые идеи, амортизированный анализ.
15. Деревья поиска: определение, основные операции. Виды сбалансированных деревьев.
16. AVL-деревья: балансировка, вращения, сложность операций.
17. Splay-деревья: принцип работы, амортизированный анализ.
18. Декартово дерево (treap): свойства, операции split и merge.
19. B-деревья: структура, применение, операции вставки и удаления.
20. Дерево отрезков: построение, запросы на отрезке, обновление элемента.
21. Отложенные операции (lazy propagation) в дереве отрезков.
22. Дерево Фенвика (двоичное индексированное дерево): принцип работы, реализация.
23. Сравнение дерева отрезков и дерева Фенвика.
24. Хэш-таблицы: основные понятия, хэш-функции, коллизии.
25. Методы разрешения коллизий: метод цепочек и открытая адресация.
26. Универсальное хэширование. Совершенное хэширование.
27. Фильтр Блума: устройство, операции, оценка вероятности ложного срабатывания.
28. Применение фильтров Блума на практике.
29. Анализ сложности алгоритмов: худший, средний, амортизированный случай.
30. Сравнительный анализ структур данных и алгоритмов для типовых задач

Зачеты и экзамены могут быть проведены в письменной форме, а также в письменной форме с устным дополнением ответа. Зачеты служат формой проверки качества выполнения студентами лабораторных работ, усвоения семестрового учебного материала по дисциплине (модулю), практических и семинарских занятий (при отсутствии экзамена по дисциплине).

По итогам зачета, соответствии с модульно – рейтинговой системой университета, выставляются баллы с последующим переходом по шкале баллы – оценки за зачет,

выставляемый как по наименованию «зачтено», «не зачтено», так и дифференцированно т.е. с выставлением отметки по схеме – «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», определяемое решением Ученого совета университета и прописываемого в учебном плане.

Экзамен по дисциплине (модулю) служит для оценки работы студента в течении семестра (года, всего срока обучения и др.) и призван выявить уровень, качество и систематичность полученных им теоретических и практических знаний, приобретения навыков самостоятельной работы, развития творческого мышления, умения синтезировать полученные знания и применять их в решении практических задач. По итогам экзамена, в соответствии с модульно – рейтинговой системой университета выставляются баллы, с последующим переходом по шкале оценок на оценки: «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно», свидетельствующие о приобретенных компетенциях или их отсутствии.

Критерии оценки уровня сформированности компетенций по результатам проведения зачета:

- оценка «зачтено»: обучающийся демонстрирует всестороннее, систематическое и глубокое знание материала, свободно выполняет задания, предусмотренные программой дисциплины, усвоивший основную и дополнительную литературу. Обучающийся выполняет задания, предусмотренные программой дисциплины, на уровне не ниже базового;

- оценка «не зачтено»: обучающийся демонстрирует незнание материала, не выполняет задания, предусмотренные программой дисциплины. Обучающийся не выполняет задания, предусмотренные программой дисциплины, на уровне ниже базового. Дальнейшее освоение ОПОП не возможно без дополнительного изучения материала и подготовки к зачету.

Критерии оценки уровня сформированности компетенций по результатам проведения дифференцированного зачёта (зачета с оценкой) / экзамена:

- оценка **«отлично»**: обучающийся дал полный, развернутый ответ на поставленный вопрос, проявил совокупность осознанных знаний об объекте, доказательно раскрыл основные положения темы. В ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий, явлений. Обучающийся подкрепляет теоретический ответ практическими примерами. Ответ сформулирован научным языком, обоснована авторская позиция обучающегося. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа или с помощью «наводящих» вопросов преподавателя. Обучающимся продемонстрирован высокий уровень владения компетенцией(-ями);

- оценка **«хорошо»**: обучающимся дан полный, развернутый ответ на поставленный вопрос, проявлено умение выделять существенные и несущественные признаки, причинно-следственные связи. Ответ четко структурирован, логичен, но есть недочеты в формулировании понятий, решении задач. При ответах на дополнительные вопросы допущены незначительные ошибки. Обучающимся продемонстрирован повышенный уровень владения компетенцией(-ями);

- оценка **«удовлетворительно»**: обучающимся дан неполный ответ на вопрос, логика и последовательность изложения имеют существенные нарушения. Допущены грубые ошибки при определении сущности раскрываемых понятий, явлений, нарушена логика ответа, не сделаны выводы. Речевое оформление требует коррекции. Обучающийся испытывает затруднение при ответе на дополнительные вопросы. Обучающимся продемонстрирован базовый уровень владения компетенцией(-ями);

- оценки **«неудовлетворительно»**: обучающийся испытывает значительные трудности в ответе на вопрос, допускает существенные ошибки, не владеет терминологией, не знает основных понятий, не может ответить на «наводящие» вопросы преподавателя. Обучающимся продемонстрирован низкий уровень владения компетенцией(-ями).