

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор высшей школы  
программной инженерии  
А.В. Малеев**

<b>по дисциплине:</b>	<b>Рабочая программа дисциплины (модуля)</b> Теория отказоустойчивых распределенных систем
<b>по направлению:</b>	Программная инженерия
<b>профиль подготовки:</b>	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
<b>курс:</b>	3
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 48 час.

Всего часов: 108, всего зач. ед.: 3

Программу составил: А.В. Созыкин, канд. техн. наук, доцент

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 28.04.2023

## Аннотация

Курс посвящен теории, лежащей в основе современных промышленных распределенных систем: файловых систем, очередей сообщений, key/value хранилищ, баз данных. Эти системы хранят десятки и сотни петабайт данных, обслуживают многие тысячи запросов в секунду и масштабируются до сотен и тысяч машин, переживая при этом отказы дисков и питания, дрейф часов, задержки и нарушения связности сети, а потому устроены невероятно сложно.

Но если посмотреть сквозь все инженерные детали и сотни тысяч строк кода, то окажется, что сложность, связанную с распределенностью, можно заключить в относительно простые модели и задачи: как узлам договориться о порядке доставки сообщений в асинхронной сети, как выбрать лидера среди равноправных машин, как добавить в систему еще один сервер или обнаружить сбойную машину. Именно от решения этих задач в конечном итоге будут зависеть важнейшие характеристики всей системы: границы ее отказоустойчивости, доступность при нестабильном поведении сети и модель согласованности данных.

В курсе мы рассмотрим эти задачи, исследуем ограничения, которые накладывает на них модель сети и сбоев, и потрогаем практические алгоритмы, которые применяются в известных промышленных распределенных системах.

## 1. Цели и задачи

### Цель дисциплины

- научить студента видеть за распределенными системами ряд фундаментальных задач, которые определяют ключевые характеристики этих систем: отказоустойчивость, масштабируемость, доступность;
- изучить различные модели сети и сбоев, исследовать ограничения, которые они накладывают на решения этих задач;
- изучить ключевые алгоритмы, которые используются в промышленных распределенных системах;
- научить студента ориентироваться в научной области, познакомиться с ключевыми академическими работами.

### Задачи дисциплины

- знает теоретические модели, ключевые задачи и результаты о невозможности (Atomic Broadcast, Consensus);
- знает алгоритмы, которые используются в промышленных распределенных системах (Multi-Paxos, RAFT, распределенные транзакции, PBFT, Bitcoin);
- знает подходы к верификации распределенных систем, владеет формальными методами верификации;
- умеет программировать изученные алгоритмы с применением современных инструментов асинхронного программирования (файберы, фьючи и т.д.).

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.1 Способен устанавливать разные виды коммуникации (учебную, научную, деловую, неформальную и др.)
	УК-3.2 Взаимодействует с другими членами команды для достижения поставленной задачи
УК-6 Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	УК-6.2 Способен планировать самостоятельную деятельность в решении профессиональных задач; подвергать критическому анализу проделанную работу; находить и творчески использовать имеющийся опыт в соответствии с задачами саморазвития

ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.2 Владеет навыками создания программного обеспечения для ЭВМ и систем различной архитектуры
ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.1 Обладает навыками разработки архитектуры программных систем и компонентов с учетом требований к производительности, надежности и безопасности
	ОПК-3.3 Знает основы информационной безопасности и методы защиты программного обеспечения от угроз и атак
ОПК-8 Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-8.2 Умеет применять технологии машинного обучения в различных прикладных областях
ПК-1 Способен самостоятельно или в качестве члена малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-1.2 Способен проводить научные исследования самостоятельно или в качестве члена малого научного коллектива
ПК-2 Способен формализовать и алгоритмизировать поставленную задачу	ПК-2.1 Способен формализовать и алгоритмизировать поставленную задачу

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- алгоритмы, решающие задачи вычислительной геометрии;
- методики распараллеливания алгоритмов, способы синхронизации потоков, разделения доступа к данным и контроля исполнения подзадач;
- оценки сложности стандартных алгоритмов.

уметь:

- реализовывать алгоритмы, решающие задачи вычислительной геометрии;
- реализовывать параллельные алгоритмы различной, выполнять синхронизацию потоков и доступа к данным.

владеть:

- средствами стандартной библиотеки C++ для создания многопоточных приложений;
- методами декомпозиции задач в области информационных технологий и построения единого решения с использованием изученных алгоритмов.

### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

#### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Модель распределенной системы	4	4		4
2	Репликация и модели согласованности	4	4		4
3	Atomic Broadcast и State Machine Replication	2	4		2

4	Распределенный консенсус	2	2		2
5	Алгоритм Single Decree Paxos	2	2		4
6	Алгоритм Multi-Paxos	2	2		4
7	Paxos Made Live	2	2		4
8	Распределенные транзакции	2	2		4
9	Формальные методы для верификации распределенных систем	2	2		4
10	Византийские отказы	2	2		4
11	Алгоритм Practical Byzantine Fault Tolerance (PBFT)	2	2		6
12	Bitcoin. Алгоритм HotStuff	4	2		6
Итого часов		30	30		48
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

## 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 6 (Весенний)

### 1. Модель распределенной системы

Модель распределенной системы: снаружи – внутри – узлы и модель передачи сообщений, снаружи – конкурентный атомарный объект. Моделирование узлов, сети, отказов, часов.

Время, виды часов (кварцевые, атомные), дрейф. Невозможность синхронизации часов без дрейфа в синхронной сети с неопределенностью при доставке сообщений. GPS и синхронизация часов, применение GPS и атомных часов в распределенных системах: TrueTime.

Семинар:

Время в промышленных системах: монотонные и wall time часы, стандартные временные оси, NTP и монотонность, високосные секунды и leap smearing.

Устройство датацентра: стойки и ToR коммутаторы, коммутационная файбрика, отказоустойчивость и задержки внутри ДЦ, охлаждение и питание. Магистральные кабели: задержки, поломки. Google private network и партишены.

Понятие failure domains

### 2. Репликация и модели согласованности

Мотивирующий пример: K/V хранилище, репликация, модели согласованности, линейаризуемость.

Задача репликации регистра, алгоритм ABD. Наивные кворумные операции и нарушение линейаризуемости. Случай одного писателя, двухфазное чтение, аналогия со свободой от блокировок в многопоточных алгоритмах, док-во линейаризуемости. Случай многих писателей, согласованный выбор временных меток, физическое время / двухфазная запись, док-во линейаризуемости. Выбор монотонных меток с помощью TrueTime. Переконфигурация набора реплик, рестарты реплик, возможность обобщения на более сложные операции.

Семинар:

Устройство отказоустойчивого локального хранилища для key/value хранилища / базы данных. Выбор API, выбор гарантий надежности: atomicity, durability. Выбор модели оценки сложности.

Физические устройства: HDD, SSD. HDD: время поворота блина и время seek-а, паттерны доступа и их стоимости. Характеристики современных HDD. SSD: API флэш памяти, страницы и блоки, изнашивание, FTL, паттерны доступа

Структуры данных: B+-деревья и LSM-деревья, read/write amplification

### 3. Atomic Broadcast и State Machine Replication

Примитив Atomic (Totally Ordered) Broadcast, свойства. АВ как транспорт команд, алгоритм репликации произвольного автомата (RSM), доказательство линейризуемости. Примеры применения RSM в индустрии.

Большие автоматы, шардирование и транзакции, недетерминизм. Таймауты на клиенте и семантика exactly-once. Недетерминизм. Read-only операции. Параллелизм.

Семинар:

Распределенные файловые системы (DFS). Мотивация. Выбор API, Append/Write.

Дизайн локальной файловой системы: абстракция блочного устройства, слой данных (блоки файлов) и метаданных (namespace, inode-ы).

Разделение DFS на Meta store и Chunk store. Иммуабельность чанков, выбор API для Chunk store, выражение операций DFS через Meta/Chunk store. Реализация Meta store: один узел -> RSM. Реализация chunk store, eventual consistency.

Сравнение с GFS. Операция перезаписи, лизы и primary чанков, гарантии согласованности данных. Гарантии атомарности при пересечении границы чанков.

LSM over DFS.

### 4. Распределенный консенсус

Atomic Broadcast как Reliable Broadcast + Consensus, эквивалентность задач АВ и Consensus.

Невозможность консенсуса в асинхронной системе: 1) граница  $n > 2f$  2) теорема FLP о невозможности консенсуса в асинхронной сети со сбоями для детерминированных процессов. Практические следствия.

Семинар:

Асинхронность в программировании: корутины / фиберы / фьючи

### 5. Алгоритм Single Decree Paxos

История алгоритма: статья Part Time Parliament, греки и репликация, статья Paxos Made Simple. Общая идея алгоритма, протокол, разбор сценариев. Интуиция для фазы Prepare. Понятие выбора, контрпример для большинства аксепторов с одним значением. Доказательство корректности. FLP и сценарий лайвлока – dueling proposers. Извлечение выбранного значения. Оптимизации.

Семинар:

Программная симуляция распределенной системы.

### 6. Алгоритм Multi-Paxos

Эффективная реализация Atomic Broadcast, общая схема RSM, репликация лога команд. Примеры применения RSM в реальных системах.

Multi-Paxos: независимые инстансы консенсуса в для каждого слота лога. Выбор лидера, алгоритм Лэмпорта и его недостатки, ортогональность выбора лидера и репликации, ситуация двух лидеров. Пайплайнинг, ускорение протокола на быстром пути до одного RTT. Масштабирование фазы Prepare на суффикс лога, понятие эпохи. Правила коммита команды лога.

Семинар:

Алгоритм RAFT. Роли, термы, фазы выбора лидера и репликации. Алгоритм выбора лидера в терме. Нетривиальные сценарии, правила коммита и правила голосования. Сравнение RAFT и Multi-Paxos. Разбор промышленной реализации RAFT.

## 7. Paxos Made Live

Применение Multi-Paxos в промышленной системе. Выбор числа реплик. Расположение реплик, failure domains. Задача переконфигурации: наивный подход, служебная команда в протоколе Multi-Paxos,  $\alpha$ -метод. Read-only операции, кворумное подтверждение, использование часов и leader leases. Групповой коммит. Компактификация лога и снимки состояния. Снимки: персистентность, CoW и fork, fuzzy snapshots в ZK. Устройство лога команд: сегментирование, чексуммы, преаллокация и fsync.

Семинар:

Консенсус как сервис: Google Chubby и Apache ZooKeeper. Применение ZK в промышленных распределенных системах.

Crash consistency и файловые системы

## 8. Распределенные транзакции

Транзакции, ACID, изоляция транзакций, сериализуемость. consistency models и гарантии изоляции ANSI, аномалии.

Конфликтная сериализуемость, механизм двухфазных блокировок.

Мультиверсионность и снапшоты, изоляция снапшотов, Google Percolator - протокол транзакций поверх BigTable.

Двухфазный коммит(2PC) в Google Spanner, применение TrueTime.

Детерминированные транзакции (Calvin)

## 9. Формальные методы для верификации распределенных систем

Проблемы дизайна и верификации распределенных систем, стандартные подходы к верификации. Формальная спецификация и explicit model checking. Граф конфигураций для распределенной системы в асинхронной модели. Масштаб моделей для практической проверки и почему такого масштаба достаточно. Свойства safety и liveness для распределенных систем, линейная темпоральная логика (LTL), выражение типичных свойств для конкурентных / распределенных алгоритмов / объектов в LTL. Язык TLA+.

Разбор спецификаций TLA+ для Single Decree Paxos, RAFT. Техники моделирования распределенных алгоритмов и систем на TLA+.

Язык PlusCal для моделирования многопоточных алгоритмов. Трансляция PlusCal в TLA+.

Fault Injection на примере фреймворка Jepsen: инструменты для внедрения сбоев сети / времени, различные сценарии партишенов, тестирование линейности.

## 10. Византийские отказы

Византийская модель сбоев. Причины и примеры византийского поведения. Почему промышленные системы не учитывают византийские сбои. Аутентификация и цифровые подписи. Граница  $n > 3f$  для задачи консенсуса, переход через границу в византийской модели и модели с отказами узлов. Рандомизированный алгоритм Ben-Or, кворумы для византийских алгоритмов.

Семинар:

Криптографические инструменты: хэш-функции, цифровые подписи, сертификаты, TLS

## 11. Алгоритм Practical Byzantine Fault Tolerance (PBFT)

Репликация автомата в византийском окружении. Public Key Infrastructure. Получение ответа от византийской системы. Варианты византийского поведения primary. Фазы Pre-Prepare и Prepare. Rotating primary, протокол перехода через эпоху, кворумные сертификаты. Локальное знание и фаза Commit. Снимки состояния автомата. Цифровые подписи и коды аутентификации сообщений.

Семинар:  
Разбор реализации PBFT.

## 12. Bitcoin. Алгоритм HotStuff

Общая схема электронных денег, граф транзакций и цифровые подписи. Проблема double spending и лог транзакций, задача репликация лога в византийском окружении. Децентрализация и публичность, анонимность и псевдонимность, динамический набор реплик и публичные ключи в качестве адресов. Блокчейн, блоки, транзакции, gossiping. Децентрализованная лотерея – PoW, форки, стабилизация, атака 51%, finality. Мотивация майнеров и эмиссия монет. Блокчейн через линзы классических алгоритмов репликации, сравнение с PBFT.

Семинар:  
Selfish mining, шардирование и транзакции между блокчейнами (atomic swaps), анонимность и доказательства с нулевым разглашением

## 5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютерами для каждого студента.

## 6. Перечень рекомендуемой литературы

### Основная литература

1. Самопроверяемые устройства и отказоустойчивые системы [Текст]/Е. С. Согомонян, Е. В. Слабаков, -М., Радио и связь, 1989

Литература, рекомендованная для самостоятельного изучения:

- Diego Ongaro, & John Ousterhout. (n.d.). search of an understandable consensus algorithm (extended version). <http://ramcloud.stanford.edu/raft.pdf>.
- Leslie Lamport, John Matthews, Mark Tuttle, & Yuan Yu. (2002). Specifying and Verifying Systems with TLA+.
- Leslie Lamport. (2000). The part-time parliament.
- Michael J Fischer, Nancy A Lynch, Michael S Paterson, & Coventry England. (1985). Impossibility of distributed consensus with one faulty process.

### Дополнительная литература

Литература, рекомендованная для самостоятельного изучения:

- Barbara Liskov. (n.d.). Chapter 7 From Viewstamped Replication to Byzantine Fault Tolerance.
- Herlihy, M. (2018). Atomic Cross-Chain Swaps.

## 7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

T. Chandra, S. Toueg / Unreliable Failure Detectors for Reliable Distributed Systems / 1996 / <https://dl.acm.org/doi/10.1145/226643.226647>

M. Fischer, N. Lynch, M. Paterson / Impossibility of Distributed Consensus with One Faulty Process / 1985 / <https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf>

L. Lamport / The Part-Time Parliament / 1998 / <https://lamport.azurewebsites.net/pubs/lamport-paxos.pdf>

D. Ongaro and J. Ousterhout / In Search of an Understandable Consensus Algorithm / 2014 / <https://raft.github.io/raft.pdf>

**8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

ПО для разработки и отладки программ на языке программирования C++.

**9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Выполнение домашнего задания является одной общей работой студента. На занятиях особое внимание обращается на современные способы разработки программных продуктов. Необходимо требовать от студентов качественного и понятного программного кода на ряду и в равной степени с правильной работой программы. На занятиях стоит постоянно разбирать ошибки, которые допускали студенты в своих программах, а так же логические ошибки.



**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Программная инженерия
<b>профиль подготовки:</b>	Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии
<b>курс:</b>	3
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

**Разработчик:** А.В. Созыкин, канд. техн. наук, доцент

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.1 Способен устанавливать разные виды коммуникации (учебную, научную, деловую, неформальную и др.)
	УК-3.2 Взаимодействует с другими членами команды для достижения поставленной задачи
УК-6 Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	УК-6.2 Способен планировать самостоятельную деятельность в решении профессиональных задач; подвергать критическому анализу проделанную работу; находить и творчески использовать имеющийся опыт в соответствии с задачами саморазвития
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.2 Владеет навыками создания программного обеспечения для ЭВМ и систем различной архитектуры
ОПК-3 Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	ОПК-3.1 Обладает навыками разработки архитектуры программных систем и компонентов с учетом требований к производительности, надежности и безопасности
	ОПК-3.3 Знает основы информационной безопасности и методы защиты программного обеспечения от угроз и атак
ОПК-8 Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий	ОПК-8.2 Умеет применять технологии машинного обучения в различных прикладных областях
ПК-1 Способен самостоятельно или в качестве члена малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-1.2 Способен проводить научные исследования самостоятельно или в качестве члена малого научного коллектива
ПК-2 Способен формализовать и алгоритмизировать поставленную задачу	ПК-2.1 Способен формализовать и алгоритмизировать поставленную задачу

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Теория отказоустойчивых распределенных систем» обучающийся должен:

### знать:

- алгоритмы, решающие задачи вычислительной геометрии;
- методики распараллеливания алгоритмов, способы синхронизации потоков, разделения доступа к данным и контроля исполнения подзадач;
- оценки сложности стандартных алгоритмов.

### уметь:

- реализовывать алгоритмы, решающие задачи вычислительной геометрии;
- реализовывать параллельные алгоритмы различной, выполнять синхронизацию потоков и доступа к данным.

### владеть:

- средствами стандартной библиотеки C++ для создания многопоточных приложений;
- методами декомпозиции задач в области информационных технологий и построения единого решения с использованием изученных алгоритмов.

### 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

#### 1. Доказательство корректности алгоритма RAFT.

Докажите Leader Completeness Property – пусть в терме  $t$  лидер закомитил в лог некоторую команду, тогда эта команда будет присутствовать в логе лидера во всех последующих термах.

#### 2. Написание спецификаций для алгоритмов репликации регистра.

Напишите спецификации для двух алгоритмов репликации регистра для случая 1 писателя.

#### 3. Анализ кода системы Apache ZooKeeper.

- Найдите код автомата, который реплицирует ZooKeeper
- Найдите код, который выполняет сегментирование лога транзакций и преаллокацию сегментов
- Найдите код, который пишет снимок дерева. Как он синхронизирован с конкурирующими апдейтами?

#### 4. Анализ статьи про Google BigTable.

Прочтите и разберите статью Bigtable: A Distributed Storage System for Structured Data.

Некоторые контрольные вопросы по статье:

- Что такое таблет в BigTable?
- Какие данные таблета хранятся в оперативной памяти обслуживающего его таблет-сервера?
- Зачем в LSM нужен WAL? Почему таблет-сервер ведет сразу два журнала?

### 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Модель распределенной системы. Время
2. Репликация, отказоустойчивый атомарный регистр
3. State Machine Replication, Atomic Broadcast и Consensus
4. Невозможность консенсуса
5. Single-Decree Paxos
6. Multi-Paxos, RAFT
7. Paxos Made Live
8. Распределенные транзакции
9. Верификация, формальные методы
10. Византийские сбои
11. Practical Byzantine Fault-Tolerance
12. Bitcoin и блокчейны

#### Критерии оценивания

отлично

10 всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

9 систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

8 глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

хорошо

7 твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

6 знает материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

5 знает основной материал, грамотно излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач неточности;

удовлетворительно

4 фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

3 характер знаний достаточен для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

неудовлетворительно

2 не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет правильно использовать полученные знания при решении типовых практических задач.

1 не знает формулировок основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

## **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Дифференцированный зачет выставляется на основе работы на семинаре и выполнения домашних работ, либо, в случае пересдачи комиссии, выполнения задания и его защиты комиссии.

Оценка за дифференцированный зачет выставляется из соотношения: 30% за теоретическую часть при ответе на билет, и 70% - за практическую по итогам выполненных домашних заданий.