

**Федеральное государственное бюджетное образовательное
учреждение высшего образования
«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

ФАКУЛЬТЕТ ФИНАНСОВ И БАНКОВСКОГО ДЕЛА

(наименование структурного подразделения (института/факультета/филиала))

Кафедра «Фондовые рынки и финансовый инжиниринг»

(наименование кафедры)

УТВЕРЖДЕНА

Кафедрой «Фондовые рынки и
финансовый инжиниринг»
Факультета финансов и банковского дела
Протокол от «04» сентября 2020 г.
№5

ПЕРЕУТВЕРЖДЕНА

Кафедрой «Фондовые рынки и
финансовый инжиниринг»
Факультета финансов и банковского дела
Протокол от «20» апреля 2021 г.
№5

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.В.03 «Технология работы с большими данными»

(индекс, наименование дисциплины (модуля), в соответствии с учебным планом)

38.04.08 «Финансы и кредит»

(код, наименование направления подготовки (специальности))

«Денежно-кредитное и финансовое регулирование экономики»

(направленность(и) (профиль (и)/специализация(ии))

Магистр

(квалификация)

Очная/очно-заочная/заочная

(форма(ы) обучения)

Год набора: 2021

Москва, 2021 г.

Автор–составитель:

Гусев А.И., преподаватель кафедры «Финансы, денежное обращение и кредит»

Заведующий кафедрой

«Фондовые рынки и финансовый инжиниринг» д.э.н., проф. Корищенко К.Н.

СОДЕРЖАНИЕ

1.Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения программы	4
2. Объем и место дисциплины (модуля) в структуре ОП ВО	4
3. Содержание и структура дисциплины	6
4. Материалы текущего контроля успеваемости обучающихся и фонд оценочных средств промежуточной аттестации по дисциплине	11
5. Методические указания для обучающихся по освоению дисциплины	30
6. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", включая перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)	32
6.1. Основная литература.....	32
6.2. Дополнительная литература.....	32
6.3. Учебно-методическое обеспечение самостоятельной работы.....	33
6.4. Нормативные правовые документы.	33
6.5. Интернет-ресурсы.	33
7. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы.....	35

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения программы

1.1. Дисциплина Б1.В.03 «Технология работы с большими данными» обеспечивает овладение следующими компетенциями:

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции
ПКс-5	Способен использовать современные информационные технологии и программные средства для решения прикладных финансово-экономических задач.	ПКс-5.1	Способен оценивать влияние технологий больших данных на результаты решений исследовательских и практических задач.

1.2. В результате освоения дисциплины у студентов должны быть сформированы:

ОТФ/ТФ (при наличии профстандарта)	Код этапа освоения компетенции	Результаты обучения
Использование трудовых функций для данной профессиональной компетенции, установленной самостоятельно, не предусмотрено.	ПКс-5.1	На уровне знания: Знает: 1. Архитектуру, устройство и функционирование вычислительных систем; 2. Инфраструктуру информационных технологий; 3. Программные средства по автоматизации профессиональной деятельности.
		На уровне умения: Умеет: 1. Применять при решении прикладных задач основные методы, способы и средства получения, хранения и преобразования информации; анализировать большие данные.
		На уровне навыков: Владеет навыками: 1. Использования современных информационных технологий и программных средств для решения прикладных финансово-экономических задач.

2. Объем и место дисциплины (модуля) в структуре ОП ВО

Объем дисциплины

Дисциплина Б1.В.03 «Технология работы с большими данными» составляет 3 зачетных единиц, т.е. 108 академических часов.

Для студентов очной и очно-заочной формы обучения на контактную работу с преподавателем выделено 24 часа, из них 8 часов лекций и 14 часов практических занятий, в том числе 2 часа на консультацию по промежуточной аттестации; на самостоятельную работу обучающихся выделено 48 часов. Для студентов заочной формы обучения на контактную работу с преподавателем выделено 12 часов, из них 4 часа лекций и 8 часов практических занятий, на самостоятельную работу обучающихся выделено 90 часов.

Место дисциплины в структуре ОП ВО

Дисциплина Б1.В.03 «Технология работы с большими данными» изучается на 1 курсе, во 2 семестре студентами очной и очно-заочной формы обучения; студентами заочной формы обучения изучается на 2 курсе.

Дисциплина Б1.В.03 «Технология работы с большими данными» реализуется после изучения дисциплин бакалавриата.

Форма промежуточной аттестации в соответствии с учебным планом – экзамен.

3. Содержание и структура дисциплины Очная форма обучения

№ п/п	Наименование тем (разделов)	Объем дисциплины , час.						Форма текущего контроля успеваемости* , промежуточно й аттестации***
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий				СР	
			Л	ЛР	ПЗ	КСР		
Тема 1	Обработка, агрегация и визуализация данных	16	2		2		12	ПР
Тема 2	Анализ изображений	18	2		4		12	ПР
Тема 3	Методы машинного обучения	18	2		4		12	ПР
Тема 4	Технологии хранения и обработки больших данных	18	2		4		12	ПР
Консультация на промежуточную аттестацию		2						
Промежуточная аттестация		36						
Всего по курсу:		108	8		14		48	Экз.

Очно-заочная форма обучения

№ п/п	Наименование тем (разделов)	Объем дисциплины , час.						Форма текущего контроля успеваемости* , промежуточной аттестации***
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий				СР	
			Л	ЛР	ПЗ	КСР		
Тема 1	Обработка, агрегация и визуализация данных	16	2		2		12	ПР
Тема 2	Анализ изображений	18	2		4		12	ПР
Тема 3	Методы машинного обучения	18	2		4		12	ПР
Тема 4	Технологии хранения и обработки	18	2		4		12	ПР

	больших данных							
Консультация на промежуточную аттестацию	2							
Промежуточная аттестация	36							
Всего по курсу:	108	8		14		48		<i>Экз.</i>

Заочная форма обучения

№ п/п	Наименование тем (разделов)	Объем дисциплины , час.					Форма текущего контроля успеваемости **, промежуточной аттестации***	
		Всего	Контактная работа обучающихся с преподавателем по видам учебных занятий			СР		
			Л	ЛР	ПЗ			КСР
Тема 1	Обработка, агрегация и визуализация данных	24	1		1		22	ПР
Тема 2	Анализ изображений	26	1		3		22	ПР
Тема 3	Методы машинного обучения	24	1		1		22	ПР
Тема 4	Технологии хранения и обработки больших данных	28	1		3		24	ПР
Промежуточная аттестация		6						
Всего по курсу:		108	4		8		90	Экз.

Примечание:

** – формы текущего контроля успеваемости: практическая работа (ПР).

*** - формы промежуточной аттестации: экзамен (Экз.).

Содержание дисциплины

Тема 1. Обработка, агрегация и визуализация данных

Процесс интеллектуального анализа данных. Основные определения и термины, подходы в машинном обучении. Роль машинного обучения в современных прикладных задачах. Основные этапы работы аналитика с методами машинного обучения. Data-driven подход. Основы работы с инструментальными средствами машинного обучения.

Извлечение данных. Основные задачи извлечения данных. Работа со структурированными файлами, подключение к реляционным БД, написание SQL-запросов.

Трансформация данных. Обогащение данных. Очистка, восстановление пропущенных значений, обработка аномалий. Кодирование и биннинг. Понятие качества данных и методы его проверки.

Визуализация данных. Работа с простыми визуализациями. Столбиковые диаграммы, круговые диаграммы, диаграммы рассеяния, тепловые карты, «ящик с усами».

Тема 2. Анализ изображений

Методы улучшения контраста изображений. Понятие цифрового изображения. Пространственное и яркостное разрешение изображения. Статистические характеристики изображения. Понятия динамического диапазона и контраста изображения. Количественные меры контраста. Локальный контраст. Гистограмма яркости. Отношение сигнал-шум. Методы улучшения контраста изображений. Масштабирование яркости. Выравнивание гистограммы яркостей. Выравнивания яркости по заданной гистограмме. Методы выделения и улучшения границ. Понятие границы. Градиент яркости. Использование линейных фильтров для оценки градиента. Операция свертки. Градиентные операторы. Сглаживание изображений. Улучшение резкости изображения. Нерезкое маскирование.

Методы устранения шумов. Источники шума на изображениях. Методы устранения шума. Устранение шума линейными фильтрами. Фильтр простого скользящего среднего. Свойство сепарабельности линейного фильтра. Биномиальный фильтр. Медианная фильтрация. Билатеральный фильтр. Управляемые фильтры.

Выделение и анализ объектов. Задачи высокоуровневого анализа изображений. Пороговое выделение объектов. Понятия точек интереса и углов на изображении. Методы выделения углов. Понятия ключевых точек и дескрипторов. Виды дескрипторов. Использование дескрипторов для выделения объектов. Оценка качества выделения объектов на изображении.

Нейронные сети для анализа изображений. Архитектуры нейронных сетей, используемые для обработки и анализа изображений. Сверточные нейронные сети. Перенос знаний в сверточных нейронных сетях. Сегментация и классификация изображений с помощью нейронных сетей. Сеть U-Net. Сети AlexNet, VGG, ResNet.

Тема 3. Методы машинного обучения

Сокращение размерности. Методы факторного анализа. Метод главных компонент. Метод сингулярного разложения. Методы определения оптимального числа факторов. Вращение факторов. Сокращение размерности переменных для визуализации многомерных данных. Кластеризация. Методы классификации без учителя. Метод k-средних, иерархическая агломерация. Метрики расстояния между объектами. Метрики расстояния между кластерами. Понятие нечеткой кластеризации. Кластеризация документов.

Классификация. Методы классификации с учителем и без учителя. Понятие обучающей выборки. Метод ближайших соседей. Наивный байесовский подход. Деревья решений. Обучение и использование моделей. Распознавание спама.

Прогнозирование. Методы прогнозирования временных рядов. Регрессионные модели. Линейная и логистическая регрессия. Проблема мультиколлинеарности. Предсказание курса акций.

Качество моделей машинного обучения. Основные метрики качества моделей машинного обучения. Матрица запутанности (ошибок). Точность, полнота, MSE, MAE и др. метрики. Оценка качества моделей. Выбор наилучшей модели.

Тема 4. Технологии хранения и обработки больших данных

Современные BigData-решения и архитектуры. Виды задач, решаемых BigData-системами. Зависимость способа решения задачи от ее разновидности. Составные части BigData-систем. Проблема оценки качества BigData-систем. Функциональные и нефункциональные требования. Виды обработки: синхронная и асинхронная, пакетная и поточная. Преимущества и недостатки. Соотношение видов обработки с пользовательским опытом. Системы оперативного и долгосрочного хранения данных.

Системы разработки, сборки и доставки кода. Качества языка программирования, влияющие на производительность. Сравнение языков программирования на примере реализации однотипной BigData-задачи. Системы контроля версий. Git, SVN. операции. Проблема управления версиями программных модулей и публикации артефактов. Системы сборки Maven и Gradle. Репозиторий артефактов программного кода.

Специфика развертывания распределенных высокопроизводительных приложений. Введение в облачные сервисы (IaaS, PaaS, SaaS). Философия DevOps, непрерывная интеграция и доставка кода.

Технология обработки и хранения данных Hadoop. Спектр решаемых задач. Примеры использования в крупных компаниях. Составные части Hadoop-кластера. Экосистема Hadoop-проектов. Распределенное выполнение программ. Операции отображения и свертки. Примеры. Возможность переопределять части MapReduce программы в каркасе Hadoop. Возможности Hadoop Streaming API. Примеры реализации MapReduce-программ на языке python и запуска из консоли bash.

Организация хранения данных в BigData-системах. Определение Базы Данных и Системы Управления Базами Данных. Организация хранения информации в классических СУБД на примере PostgreSQL. Вопросы обеспечения целостности данных. Требования ACID. Подходы к хранению данных. Специфика хранения файлов. Использование распределенных файловых систем в задачах обработки больших объемов данных.

Специфика хранения данных по записям. Проблематика обеспечения согласованности данных. Архитектура NoSQL (not only SQL) на примере Apache Cassandra и их отличие от классических СУБД.

Системы пакетной обработки данных. Состав и назначение Spark-кластера. Спектр решаемых задач. Примеры работы с источниками данных. План выполнения вычислений. Работа с основной и внешней памятью. Использование программного интерфейса Spark RDD. Состав и назначение кластера Hive. Связь с технологией Hadoop. Проблема импорта данных в системы пакетной обработки. Импорт данных в HDFS при помощи Sqoop. Настройка количества операций отображения..

Системы поточной обработки данных. Архитектура событийно-ориентированных BigData-систем. Задача брокера сообщений. Типовые архитектуры систем передачи сообщений. Брокер сообщений Apache Kafka. Архитектура. Производители, потребители, брокеры. Вопросы масштабирования. Принципы организации долговременного хранения сообщений. Настройка дедупликации сообщений.

Архитектура Flume. Агент, канал передачи, селектор, слив. Варианты конфигурирования Apache Spark Streaming. Принцип микропакетной обработки. Плавающее окно

Архитектура облачных BigData-приложений. Технология Docker.. Состав технологии: файлы докер, реестр образов, контейнеры, демон Docker. Слои доступа к файлам. Кластер Docker-хостов. Создание кластера. Масштабирование контейнеров. Маршрутизация входящих соединений. Оркестрация сервисов масштабных приложений. Проблемы и вызовы. Состав кластера Kubernetes. Основные элементы развертывания (Pod, Service, Route). Создание динамических сетей маршрутизации данных между сервисами. Отделение слоя доставки данных и авторизации от бизнес-логики.

Способы повышения производительности BigData-систем. Понятие и назначение кеша данных. Ограничения. Стратегии замещения данных в кеш, стратегии предвыборки. Популярные системы кеширования для BigData-задач. Технологии вычислений в памяти и хранения данных в памяти. Кластер Ignite. Возможности Ignite для организации кеширования, вычислений, передачи сообщений и машинного обучения на основе больших данных. Адаптация ресурсов приложения в зависимости от нагрузки. Способы мониторинга нагрузки. Встроенная поддержка правил автоматического масштабирования в Kubernetes.

4. Материалы текущего контроля успеваемости обучающихся и фонд оценочных средств промежуточной аттестации по дисциплине

4.1. Текущий контроль успеваемости

4.1.1. Формы текущего контроля успеваемости

	Тема (раздел)	Методы текущего контроля успеваемости
Тема 1	Обработка, агрегация и визуализация данных	Практическая работа
Тема 2	Анализ изображений	Практическая работа
Тема 3	Методы машинного обучения	Практическая работа
Тема 4	Технологии хранения и обработки больших данных	Практическая работа

Форма и средства (методы) проведения текущего контроля успеваемости

Текущий контроль успеваемости (Практическая работа) проводится в письменной форме в конце последнего занятия по каждой из тем, соответственно, подводя итог усвоения пройденной темы обучающимися. При написании Практической работы обучающийся письменно отвечает на вопрос по теме учебной дисциплины (подготавливает эссе по тематике заданного вопроса), объемом 1-2 страницы текста, не более.

Текущий контроль успеваемости проводится преподавателем, ведущим лекционные занятия по данной дисциплине. Инвалиды и лица с ограниченными возможностями здоровья, имеющие нарушения опорно-двигательного аппарата, допускаются на аттестационные испытания в сопровождении ассистентов-сопровождающих.

Во время текущего контроля успеваемости обучающиеся могут пользоваться программой учебной дисциплины, а также с разрешения преподавателя калькуляторами. В конце занятия, на котором проводится текущий контроль успеваемости, каждый обучающийся получает от преподавателя вопрос в рамках пройденного материала по теме. Время подготовки ответа не должно составлять не менее 15 минут (по желанию обучающегося ответ может быть досрочным).

Практическая работа оценивается следующим образом:

- очевидна логика решения, ее соответствие изученным теоретическим основам, методам и инструментам анализа – 2 балла,
- правильно прописаны все этапы и использована соответствующая методология – 1 балл,
- продемонстрировано знание нормативной базы – 1 балл,
- выводы соответствуют полученным результатам, обоснованы и аргументированы – 1 балл.

Неудовлетворительная оценка за практическую работу выставляется, если обучающийся набрал менее трех баллов. Переписывание практической работы, за которую обучающийся получил неудовлетворительную оценку, не допускается. При двух и более неудовлетворительных оценках за практические работы обучающийся не допускается к промежуточной аттестации.

Оценка результатов текущего контроля, после проверки преподавателем, объявляется обучающимся в день его проведения, письмом от преподавателя на электронную почту обучающихся.

4.1.2. Материалы текущего контроля успеваемости.

Типовые оценочные материалы по теме 1. Обработка, агрегация и визуализация данных

Примерные темы практических работ:

1. Процесс интеллектуального анализа данных.
2. Основные определения и термины, подходы в машинном обучении.
3. Роль машинного обучения в современных прикладных задачах.
4. Основные этапы работы аналитика с методами машинного обучения.
5. Data-driven подход.
6. Основы работы с инструментальными средствами машинного обучения.
7. Извлечение данных. Основные задачи извлечения данных.
8. Трансформация данных.
9. Обогащение данных.
10. Очистка, восстановление пропущенных значений, обработка аномалий.
11. Кодирование и биннинг.
12. Понятие качества данных и методы его проверки.
13. Визуализация данных.

Типовые оценочные материалы по теме 2. Анализ изображений

Примерные темы практических работ:

1. Методы улучшения контраста изображений.
2. Понятие цифрового изображения.

3. Пространственное и яркостное разрешение изображения.
4. Статистические характеристики изображения.
5. Понятия динамического диапазона и контраста изображения.
6. Количественные меры контраста.
7. Локальный контраст.
8. Гистограмма яркости.
9. Отношение сигнал-шум.
10. Методы улучшения контраста изображений.
11. Масштабирование яркости.
12. Выравнивание гистограммы яркостей.
13. Выравнивания яркости по заданной гистограмме.
14. Методы выделения и улучшения границ.
15. Градиент яркости.
16. Использование линейных фильтров для оценки градиента.
17. Градиентные операторы.
18. Сглаживание изображений.
19. Улучшение резкости изображения.
20. Нерезкое маскирование.
21. Источники шума на изображениях.
22. Устранение шума линейными фильтрами.
23. Фильтр простого скользящего среднего.
24. Свойство сепарабельности линейного фильтра.
25. Биномиальный фильтр.
26. Медианная фильтрация.
27. Билатеральный фильтр.
28. Управляемые фильтры.
29. Выделение и анализ объектов.
30. Пороговое выделение объектов.
31. Понятия точек интереса и углов на изображении.
32. Понятия ключевых точек и дескрипторов.
33. Использование дескрипторов для выделения объектов.
34. Нейронные сети для анализа изображений.
35. Архитектуры нейронных сетей, используемые для обработки и анализа изображений.

36. Сверточные нейронные сети.
37. Сегментация и классификация изображений с помощью нейронных сетей.

Типовые оценочные материалы по теме 3. Методы машинного обучения

Примерные темы практических работ:

1. Методы факторного анализа.
2. Метод главных компонент.
3. Метод сингулярного разложения.
4. Методы определения оптимального числа факторов.
5. Сокращение размерности переменных для визуализации многомерных данных.
6. Кластеризация.
7. Методы классификации без учителя.
8. Метод k-средних, иерархическая агломерация.
9. Метрики расстояния между объектами.
10. Метрики расстояния между кластерами.
11. Понятие нечеткой кластеризации.
12. Кластеризация документов.
13. Методы классификации с учителем и без учителя.
14. Понятие обучающей выборки.
15. Метод ближайших соседей.
16. Прогнозирование.
17. Регрессионные модели.
18. Линейная и логистическая регрессия.
19. Проблема мультиколлинеарности.
20. Предсказание курса акций.
21. Качество моделей машинного обучения.
22. Основные метрики качества моделей машинного обучения.

Типовые оценочные материалы по теме 4. Технологии хранения и обработки больших данных

Примерные темы практических работ:

1. Современные BigData-решения и архитектуры.
2. Виды задач, решаемых BigData-системами.
3. Зависимость способа решения задачи от ее разновидности.

4. Составные части BigData-систем.
5. Проблема оценки качества BigData-систем.
6. Виды обработки: синхронная и асинхронная, пакетная и поточная.
7. Соотношение видов обработки с пользовательским опытом.
8. Системы оперативного и долгосрочного хранения данных.
9. Качества языка программирования, влияющие на производительность.
10. Проблема управления версиями программных модулей и публикации артефактов.
11. Системы сборки Maven и Gradle.
12. Репозиторий артефактов программного кода.
13. Введение в облачные сервисы (IaaS, PaaS, SaaS).
14. Философия DevOps, непрерывная интеграция и доставка кода.
15. Технология обработки и хранения данных Hadoop.
16. Составные части Hadoop-кластера.
17. Экосистема Hadoop-проектов.
18. Распределенное выполнение программ.
19. Операции отображения и свертки.
20. Возможность переопределять части MapReduce программы в каркасе Hadoop.
21. Возможности Hadoop Streaming API.
22. Примеры реализации MapReduce-программ на языке python и запуска из консоли bash.
23. Организация хранения данных в BigData-системах.
24. Организация хранения информации в классических СУБД на примере PostgreSQL.
25. Вопросы обеспечения целостности данных.
26. Требования ACID.
27. Использование распределенных файловых систем в задачах обработки больших объемов данных.
28. Архитектура NoSQL (not only SQL) на примере Apache Cassandra и их отличие от классических СУБД.
29. Системы пакетной обработки данных.
30. Состав и назначение Spark-кластера.
31. Использование программного интерфейса Spark RDD.
32. Состав и назначение кластера Hive.
33. Проблема импорта данных в системы пакетной обработки.
34. Импорт данных в HDFS при помощи Sqoop.

35. Системы поточной обработки данных.
36. Архитектура событийно-ориентированных BigData-систем.
37. Типовые архитектуры систем передачи сообщений.
38. Брокер сообщений Apache Kafka.
39. Архитектура Flume.
40. Варианты конфигурирования Apache Spark Streaming.
41. Принцип микропакетной обработки.
42. Архитектура облачных BigData-приложений.
43. Состав технологии Docker: файлы докер, реестр образов, контейнеры, демон Docker.
44. Состав кластера Kubernetes.
45. Способы повышения производительности BigData-систем.
46. Стратегии замещения данных в кеш, стратегии предвыборки.
47. Популярные системы кеширования для BigData-задач.
48. Технологии вычислений в памяти и хранения данных в памяти.
49. Кластер Ignite.
50. Возможности Ignite для организации кеширования, вычислений, передачи сообщений и машинного обучения на основе больших данных.
51. Встроенная поддержка правил автоматического масштабирования в Kubernetes.

Образец ответа при проведении текущего контроля успеваемости

Практическая работа по теме 4. «Технологии хранения и обработки больших данных»

Вопрос №14 Философия DevOps, непрерывная интеграция и доставка кода

Идея и культура DevOps

Методология DevOps получила широкое распространение после организованной бельгийским разработчиком Патриком Дебуа в 2009 году конференции DevOpsDays.

Главная идея DevOps заключается в том, чтобы устранить перекалывание ответственности на других членов команды в больших коллективах. Взаимозависимость между созданием и эксплуатацией программного обеспечения преследовала цель привить команде новую культуру разработки продукта.

Культура DevOps предполагает, что каждый из членов команды ответственен за конечный результат. Базируется она на нескольких основных положениях:

Регулярное сотрудничество и общение. Команда должна работать слаженно, понимать потребности и ожидания всех ее членов.

Постепенное развертывание. Внедрение постепенного развертывания позволяет группам доставки выпускать продукт, имея возможность вносить обновления и делать откат, если что-то пойдет не так.

Общая ответственность. Все члены команды должны двигаться к единой цели и отвечать за проект в равной степени.

Решение проблем на ранних этапах. Методология DevOps требует, чтобы в жизненном цикле проекта задачи выполнялись как можно быстрее. Это помогает оперативно решать возникшие проблемы.

Принципы DevOps

В 2010 году Дэймоном Эдвардсом и Джоном Уиллисом была разработана модель CAMS, ключевые идеи которой стали принципами DevOps. Согласно ей, развитие DevOps идет в трех направлениях: люди, процессы и инструменты. При этом важна поддержка каждого пункта на всех этапах развития.

Аббревиатура CAMS расшифровывается следующим образом:

культура (culture);

автоматизация (automation);

измерение (measurement);

обмен (sharing).

Культура

Классические бизнес-модели в IT разделяют специалистов по разработке и эксплуатации на две отдельные группы. До появления DevOps они общались на разных языках, ведь перед разработчиками стояла задача быстро внедрять инновации, а операционный персонал отвечал за поддержание стабильной среды и инфраструктуры.

Конкурирующие рабочие цели создавали между специалистами по разработке и эксплуатации недопонимание, поэтому основная задача DevOps – изменить бизнес-культуру, разделить ответственность двух групп и объединить их профессиональные навыки.

Автоматизация

Следуя пути DevOps, код требуется переводить из стадии разработки в производство непрерывно в автоматическом режиме, поэтому автоматизацию можно считать синонимом DevOps.

В идеале автоматизировать нужно почти все:

инфраструктуру;

выпуски программного обеспечения (software releases);

тестирование;

развертывание;

основные задачи по безопасности;

политику соглашений;

задачи управления конфигурацией.

Автоматизация упрощает рабочие процессы, сокращает количество сбоев и отказов, уменьшает количество ошибок, которые возникают при ручной настройке. Повышение эффективности, улучшение производительности и польза для конечного потребителя – главные преимущества автоматизации.

Измерение

Измерение необходимо для постоянного предложения ценности и улучшений. В DevOps важно отслеживать ключевые показатели, которые зависят от целей проекта.

Измерять нужно показатели следующих процессов:

мониторинга и отслеживания производительности на протяжении всего жизненного цикла разработки программного обеспечения;

сбора, анализа и предоставления способов реагирования на обратную связь;

анализа ошибок и способов их предотвращения;

оказания помощи командам в работе над общими целями.

DevOps способствует развитию только в том случае, если конкретные показатели собираются и анализируются непрерывно.

Обмен

DevOps подразумевает тесное сотрудничество специалистов по разработке и эксплуатации. Большое внимание уделяется прозрачности и открытости в коллективе. Чем больше знаний распространяется между сотрудниками, тем больше обратной связи они получают – это помогает улучшить их работу в целом.

DevOps и Agile

DevOps является естественным продолжением гибких подходов и подходов к непрерывной доставке. DevOps и Agile могут дополнять друг друга и применяться в тандеме, но сравнивать эти методологии не стоит.

По сути DevOps объединяет две разрозненные команды (разработку и эксплуатацию), чтобы обеспечить быстрые выпуски программного обеспечения. Agile ориентирован на сотрудничество небольших команд друг с другом для быстрого реагирования на изменчивые потребности пользователей.

Основные различия между DevOps и Agile:

Разработка, тестирование и развертывание программного обеспечения происходят как в DevOps, так и в Agile. Подход Agile характерен тем, что разработка завершается сразу после развертывания. DevOps же включает операции, которые происходят постоянно, например, мониторинг и модификации программного обеспечения;

В Agile разные специалисты несут ответственность за разработку, тестирование и развертывание программного обеспечения. В DevOps за все эти процессы отвечают специально обученные инженеры;

Agile выступает за поэтапное развертывание после каждого спринта. Для DevOps характерна непрерывная доставка (до нескольких раз в день).

Жизненный цикл DevOps

Выделим основные этапы проекта, следующего принципам DevOps:

Непрерывное развитие;

Непрерывная интеграция;

Непрерывное тестирование;

Непрерывное развертывание;

Непрерывный мониторинг;

Постоянная обратная связь.

Эти этапы гарантируют оптимизацию всех процессов разработки, от предложения до производства и поставки.

Непрерывное развитие

На первом этапе происходит планирование и кодирование программного обеспечения, а также формируется видение проекта.

Код может быть написан на любом языке, но поддерживается с помощью средств контроля версий. Процесс поддержки называется управлением исходным кодом (SCM). В нем используются следующие инструменты: Git, SVN, Mercurial, CVS и JIRA.

Непрерывная интеграция

На этом этапе чаще всего применяется Jenkins. Когда в репозитории Git происходят изменения, Jenkins извлекает обновленный код и готовит сборку. Упакованный код переходит к следующему этапу и пересылается либо на рабочий, либо на тестовый сервер.

Этап включает в себя не только компиляцию кода, но и проверку, модульное тестирование, интеграционное тестирование и упаковку. Непрерывная интеграция нового кода в существующий исходный код помогает отразить изменения для конечных пользователей.

Непрерывное тестирование

На этапе непрерывного тестирования разрабатываемое программное обеспечение проверяется на наличие ошибок. Автоматическое тестирование позволяет разработчикам экономить силы и время.

Для непрерывного тестирования используются следующие инструменты: Selenium, TestNG, JUnit и т.д. Они позволяют инженерам QA тестировать несколько баз кода параллельно, чтобы гарантировать отсутствие недостатков в функциональности. При этом тестируемое приложение часто запускается в виртуальной среде или контейнерах, например, с помощью Docker.

Автоматическое тестирование выполняет Selenium, а отчеты генерирует TestNG. Весь этап можно автоматизировать с помощью инструмента непрерывной интеграции Jenkins. В конце, протестированный код повторно отправляется на этап непрерывной интеграции для обновления исходного кода.

Непрерывное развертывание

Когда код развертывается на производственных серверах, важно получить корректный результат на всех серверах.

Управление конфигурацией – ключевой процесс на этом этапе. Он обеспечивает точное развертывание приложения и поддерживает согласованность конфигураций на всех серверах.

Ansible, Puppet и Chef – инструменты, которые чаще всего используются в DevOps для быстрого и непрерывного развертывания нового кода.

Не менее важную роль на данном этапе играют инструменты контейнеризации. Vagrant и Docker обеспечивают согласованность в различных средах – от разработки и тестирования, до подготовки и производства.

Непрерывный мониторинг

Важный этап жизненного цикла DevOps, на котором в реальном времени отслеживается производительность приложения. Для этого автоматически собираются определенные показатели телеметрии и метаданных, а также настраиваются оповещения об отклонениях в работе.

Непрерывный мониторинг помогает поддерживать доступность сервисов. Он также определяет угрозы и основные причины повторяющихся системных ошибок, а проблемы безопасности автоматически решаются в момент появления.

Активное участие в непрерывном мониторинге участвуют операционные группы. Они наблюдают за действиями пользователей, проверяют системы на предмет необычного поведения и отслеживают наличие ошибок.

Для этого используются следующие популярные инструменты: Prometheus, Splunk, ELK Stack, Nagios и другие. Они обеспечивают полный контроль над производительностью системы, рабочего сервера и приложения.

Постоянная обратная связь

Непрерывная обратная связь – особый этап, на котором анализируются улучшения, сделанные на этапах непрерывного тестирования и непрерывной интеграции.

Чтобы оценить результаты изменений в конечном продукте, необходимо получить обратную связь от клиентов. Процесс разработки приложения обновляется с учетом их отзывов – после этого разработчики начинают вносить изменения в продукт. Когда отзывы становятся положительными, открывается путь для выпуска новых версий, либо для поддержки приложения.

Студент:

Группа:

4.2. Промежуточная аттестация

4.2.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы. Показатели и критерии оценивания компетенций с учетом этапа их формирования

Код компетенции	Наименование компетенции	Код этапа освоения компетенции	Наименование этапа освоения компетенции
ПКс-5	Способен использовать современные информационные технологии и программные средства для решения прикладных финансово-экономических задач.	ПКс-5.1.	Способен оценивать влияние технологий больших данных на результаты решений исследовательских и практических задач.

Этап освоения компетенции	Критерий оценивания	Показатель оценивания
ПКс-5.1. Способен оценивать влияние технологий больших данных на результаты решений исследовательских и практических задач.	Использует технологии работы с большими данными при принятии организационных и экономических решений и обосновывает финансово-экономические показатели для решения прикладных финансово-экономических задач.	Использует технологию работы с большими данными при принятии организационных и экономических решений и обосновывает финансово-экономические показатели для решения прикладных финансово-экономических задач.

4.2.2. Форма и средства (методы) проведения промежуточной аттестации

Экзамен проводится с применением следующих методов (средств): устного опроса и письменного задания.

4.2.3. Типовые оценочные средства

Пример типовых оценочных средств (вопросов к экзамену)

1. Процесс интеллектуального анализа данных.
2. Основные определения и термины, подходы в машинном обучении.
3. Роль машинного обучения в современных прикладных задачах.
4. Основные этапы работы аналитика с методами машинного обучения.
5. Data-driven подход.
6. Основы работы с инструментальными средствами машинного обучения.
7. Извлечение данных. Основные задачи извлечения данных.
8. Работа со структурированными файлами, подключение к реляционным БД, написание SQL-запросов.
9. Трансформация данных.
10. Обогащение данных.
11. Очистка, восстановление пропущенных значений, обработка аномалий.
12. Кодирование и биннинг.
13. Понятие качества данных и методы его проверки.
14. Визуализация данных.
15. Работа с простыми визуализациями.

16. Столбиковые диаграммы, круговые диаграммы, диаграммы рассеяния, тепловые карты, «ящик с усами».
17. Методы улучшения контраста изображений.
18. Понятие цифрового изображения.
19. Пространственное и яркостное разрешение изображения.
20. Статистические характеристики изображения.
21. Понятия динамического диапазона и контраста изображения.
22. Количественные меры контраста.
23. Локальный контраст.
24. Гистограмма яркости.
25. Отношение сигнал-шум.
26. Методы улучшения контраста изображений.
27. Масштабирование яркости.
28. Выравнивание гистограммы яркостей.
29. Выравнивания яркости по заданной гистограмме.
30. Методы выделения и улучшения границ.
31. Понятие границы.
32. Градиент яркости.
33. Использование линейных фильтров для оценки градиента.
34. Операция свертки.
35. Градиентные операторы.
36. Сглаживание изображений.
37. Улучшение резкости изображения.
38. Нерезкое маскирование.
39. Методы устранения шумов.
40. Источники шума на изображениях.
41. Методы устранения шума.
42. Устранение шума линейными фильтрами.
43. Фильтр простого скользящего среднего.
44. Свойство сепарабельности линейного фильтра.
45. Биномиальный фильтр.
46. Медианная фильтрация.
47. Билатеральный фильтр.
48. Управляемые фильтры.

49. Выделение и анализ объектов.
50. Задачи высокоуровневого анализа изображений.
51. Пороговое выделение объектов.
52. Понятия точек интереса и углов на изображении.
53. Методы выделения углов.
54. Понятия ключевых точек и дескрипторов.
55. Виды дескрипторов.
56. Использование дескрипторов для выделения объектов.
57. Оценка качества выделения объектов на изображении.
58. Нейронные сети для анализа изображений.
59. Архитектуры нейронных сетей, используемые для обработки и анализа изображений.
60. Сверточные нейронные сети.
61. Перенос знаний в сверточных нейронных сетях.
62. Сегментация и классификация изображений с помощью нейронных сетей.
63. Методы факторного анализа.
64. Метод главных компонент.
65. Метод сингулярного разложения.
66. Методы определения оптимального числа факторов.
67. Вращение факторов.
68. Сокращение размерности переменных для визуализации многомерных данных.
69. Кластеризация.
70. Методы классификации без учителя.
71. Метод k-средних, иерархическая агломерация.
72. Метрики расстояния между объектами.
73. Метрики расстояния между кластерами.
74. Понятие нечеткой кластеризации.
75. Кластеризация документов.
76. Классификация.
77. Методы классификации с учителем и без учителя.
78. Понятие обучающей выборки.
79. Метод ближайших соседей.
80. Наивный байесовский подход.
81. Деревья решений.
82. Обучение и использование моделей.

83. Прогнозирование.
84. Методы прогнозирования временных рядов.
85. Регрессионные модели.
86. Линейная и логистическая регрессия.
87. Проблема мультиколлинеарности.
88. Предсказание курса акций.
89. Качество моделей машинного обучения.
90. Основные метрики качества моделей машинного обучения.
91. Матрица запутанности (ошибок).
92. Современные BigData-решения и архитектуры.
93. Виды задач, решаемых BigData-системами.
94. Зависимость способа решения задачи от ее разновидности.
95. Составные части BigData-систем.
96. Проблема оценки качества BigData-систем.
97. Функциональные и нефункциональные требования.
98. Виды обработки: синхронная и асинхронная, пакетная и поточная. Преимущества и недостатки.
99. Соотношение видов обработки с пользовательским опытом.
100. Системы оперативного и долгосрочного хранения данных.
101. Системы разработки, сборки и доставки кода.
102. Качества языка программирования, влияющие на производительность.
103. Системы контроля версий.
104. Проблема управления версиями программных модулей и публикации артефактов.
105. Системы сборки Maven и Gradle.
106. Репозиторий артефактов программного кода.
107. Специфика развертывания распределенных высокопроизводительных приложений.
108. Введение в облачные сервисы (IaaS, PaaS, SaaS).
109. Философия DevOps, непрерывная интеграция и доставка кода.
110. Технология обработки и хранения данных Hadoop.
111. Составные части Hadoop-кластера.
112. Экосистема Hadoop-проектов.
113. Распределенное выполнение программ.
114. Операции отображения и свертки.
115. Возможность переопределять части MapReduce программы в каркасе Hadoop.

116. Возможности Hadoop Streaming API.
117. Примеры реализации MapReduce-программ на языке python и запуска из консоли bash.
118. Организация хранения данных в BigData-системах.
119. Определение Базы Данных и Системы Управления Базами Данных.
120. Организация хранения информации в классических СУБД на примере PostgreSQL.
121. Вопросы обеспечения целостности данных.
122. Требования ACID.
123. Использование распределенных файловых систем в задачах обработки больших объемов данных.
124. Специфика хранения данных по записям.
125. Проблематика обеспечения согласованности данных.
126. Архитектура NoSQL (not only SQL) на примере Apache Cassandra и их отличие от классических СУБД.
127. Системы пакетной обработки данных.
128. Состав и назначение Spark-кластера. Спектр решаемых задач.
129. Работа с основной и внешней памятью.
130. Использование программного интерфейса Spark RDD.
131. Состав и назначение кластера Hive. Связь с технологией Hadoop.
132. Проблема импорта данных в системы пакетной обработки.
133. Импорт данных в HDFS при помощи Sqoop.
134. Настройка количества операций отображения.
135. Системы поточной обработки данных.
136. Архитектура событийно-ориентированных BigData-систем.
137. Типовые архитектуры систем передачи сообщений.
138. Брокер сообщений Apache Kafka.
139. Архитектура Flume.
140. Варианты конфигурирования Apache Spark Streaming.
141. Принцип микропакетной обработки.
142. Архитектура облачных BigData-приложений.
143. Состав технологии Docker: файлы докер, реестр образов, контейнеры, демон Docker.
144. Кластер Docker-хостов.
145. Состав кластера Kubernetes.
146. Способы повышения производительности BigData-систем.
147. Стратегии замещения данных в кеш, стратегии предвыборки.

148. Популярныe системы кеширования для BigData-задач.
149. Технологии вычислений в памяти и хранения данных в памяти.
150. Кластер Ignite.
151. Возможности Ignite для организации кеширования, вычислений, передачи сообщений и машинного обучения на основе больших данных.
152. Встроенная поддержка правил автоматического масштабирования в Kubernetes.

Образец ответа при проведении текущего промежуточной аттестации

Экзамен по дисциплине «Технология работы с большими данными»

Вопрос №8 Состав кластера Kubernetes

Компоненты Kubernetes

При развёртывании Kubernetes вы имеете дело с кластером.

Кластер Kubernetes cluster состоит из набор машин, так называемые узлы, которые запускают контейнеризированные приложения. Кластер имеет как минимум один рабочий узел.

В рабочих узлах размещены поды, являющиеся компонентами приложения. Плоскость управления управляет рабочими узлами и подами в кластере. В промышленных средах плоскость управления обычно запускается на нескольких компьютерах, а кластер, как правило, развёртывается на нескольких узлах, гарантируя отказоустойчивость и высокую надёжность.

На этой странице в общих чертах описывается различные компоненты, необходимые для работы кластера Kubernetes.

Ниже показана диаграмма кластера Kubernetes со всеми связанными компонентами.

Плоскость управления компонентами

Компоненты панели управления отвечают за основные операции кластера (например, планирование), а также обрабатывают события кластера (например, запускают новый под, когда поле replicas развёртывания не соответствует требуемому количеству реплик).

Компоненты панели управления могут быть запущены на любой машине в кластере. Однако для простоты сценарии настройки обычно запускают все компоненты панели управления на одном компьютере и в то же время не позволяют запускать пользовательские контейнеры на этом компьютере. Смотрите страницу Создание высоконадёжных кластеров для примера настройки нескольких ведущих виртуальных машин.

kube-apiserver

Сервер API — компонент Kubernetes панели управления, который представляет API Kubernetes. API-сервер — это клиентская часть панели управления Kubernetes

Основной реализацией API-сервера Kubernetes является kube-apiserver. kube-apiserver предназначен для горизонтального масштабирования, то есть развёртывание на несколько экземпляров. Вы можете запустить несколько экземпляров kube-apiserver и сбалансировать трафик между этими экземплярами.

etcd

Распределённое и высоконадёжное хранилище данных в формате "ключ-значение", которое используется как основное хранилище всех данных кластера в Kubernetes.

Если ваш кластер Kubernetes использует etcd в качестве основного хранилища, убедитесь, что у вас

настроено резервное копирование данных.

Вы можете найти подробную информацию о *etcd* в официальной документации.

kube-scheduler

Компонент плоскости управления, который отслеживает созданные поды без привязанного узла и выбирает узел, на котором они должны работать.

При планировании развёртывания подов на узлах учитываются множество факторов, включая требования к ресурсам, ограничения, связанные с аппаратными/программными политиками, принадлежности (*affinity*) и непринадлежности (*anti-affinity*) узлов/подов, местонахождения данных, предельных сроков.

kube-controller-manager

Компонент Control Plane запускает процессы контроллера.

Вполне логично, что каждый контроллер в свою очередь представляет собой отдельный процесс, и для упрощения все такие процессы скомпилированы в один двоичный файл и выполняются в одном процессе.

Эти контроллеры включают:

Контроллер узла (*Node Controller*): уведомляет и реагирует на сбои узла.

Контроллер репликации (*Replication Controller*): поддерживает правильное количество подов для каждого объекта контроллера репликации в системе.

Контроллер конечных точек (*Endpoints Controller*): заполняет объект конечных точек (*Endpoints*), то есть связывает сервисы (*Services*) и поды (*Pods*).

Контроллеры учетных записей и токенов (*Account & Token Controllers*): создают стандартные учетные записи и токены доступа API для новых пространств имен.

cloud-controller-manager

cloud-controller-manager запускает контроллеры, которые взаимодействуют с основными облачными провайдерами. Двоичный файл *cloud-controller-manager* — это альфа-функциональность, появившиеся в Kubernetes 1.6.

cloud-controller-manager запускает только циклы контроллера, относящиеся к облачному провайдеру. Вам нужно отключить эти циклы контроллера в *kube-controller-manager*. Вы можете отключить циклы контроллера, установив флаг *--cloud-provider* со значением *external* при запуске *kube-controller-manager*.

С помощью *cloud-controller-manager* код как облачных провайдеров, так и самого Kubernetes может разрабатываться независимо друг от друга. В предыдущих версиях код ядра Kubernetes зависел от кода, предназначенного для функциональности облачных провайдеров. В будущих выпусках код, специфичный для облачных провайдеров, должен поддерживаться самим облачным провайдером и компоноваться с *cloud-controller-manager* во время запуска Kubernetes.

Следующие контроллеры зависят от облачных провайдеров:

Контроллер узла (*Node Controller*): проверяет облачный провайдер, чтобы определить, был ли удален узел в облаке после того, как он перестал работать

Контроллер маршрутов (*Route Controller*): настраивает маршруты в основной инфраструктуре облака

Контроллер сервисов (*Service Controller*): создаёт, обновляет и удаляет балансировщики нагрузки облачного провайдера.

Контроллер тома (*Volume Controller*): создаёт, присоединяет и монтирует тома, а также взаимодействует с облачным провайдером для оркестрации томов.

Компоненты узла

Компоненты узла работают на каждом узле, поддерживая работу подов и среды выполнения Kubernetes.

kubelet

Агент, работающий на каждом узле в кластере. Он следит за тем, чтобы контейнеры были запущены в поде.

Утилита kubelet принимает набор PodSpecs, и гарантирует работоспособность и исправность определённых в них контейнеров. Агент kubelet не отвечает за контейнеры, не созданные Kubernetes.

kube-proxy

kube-proxy — сетевой прокси, работающий на каждом узле в кластере, и реализующий часть концепции сервис.

kube-proxy конфигурирует правила сети на узлах. При помощи них разрешаются сетевые подключения к вашим подам изнутри и снаружи кластера.

kube-proxy использует уровень фильтрации пакетов в операционной системы, если он доступен. В противном случае, kube-proxy сам обрабатывает передачу сетевого трафика.

Среда выполнения контейнера

Среда выполнения контейнера — это программа, предназначенная для выполнения контейнеров.

Kubernetes поддерживает несколько сред для запуска контейнеров: Docker, containerd, CRI-O, и любая реализация Kubernetes CRI (Container Runtime Interface).

Дополнения

Дополнения используют ресурсы Kubernetes (DaemonSet, Deployment и т.д.) для расширения функциональности кластера. Поскольку дополнения охватывают весь кластер, ресурсы относятся к пространству имен kube-system.

Некоторые из дополнений описаны ниже; более подробный список доступных расширений вы можете найти на странице Дополнения.

DNS

Хотя прочие дополнения не являются строго обязательными, однако при этом у всех Kubernetes-кластеров должен быть кластерный DNS, так как многие примеры предполагают его наличие.

Кластерный DNS — это DNS-сервер наряду с другими DNS-серверами в вашем окружении, который обновляет DNS-записи для сервисов Kubernetes.

Контейнеры, запущенные посредством Kubernetes, автоматически включают этот DNS-сервер в свои DNS.

Веб-интерфейс (Dashboard)

Dashboard — это универсальный веб-интерфейс для кластеров Kubernetes. С помощью этой панели, пользователи могут управлять и устранять неполадки кластера и приложений, работающих в кластере.

Мониторинг ресурсов контейнера

Мониторинг ресурсов контейнера записывает общие метрики о контейнерах в виде временных рядов в центральной базе данных и предлагает пользовательский интерфейс для просмотра этих данных.

Логирование кластера

Механизм логирования кластера отвечает за сохранение логов контейнера в централизованном хранилище логов с возможностью их поиска/просмотра

Студент:

Группа:

Шкала и критерии оценки при проведении промежуточной аттестации

Оценка 5 (отлично)	Использует технологию работы с большими данными при принятии организационных и экономических решений и обосновывает финансово-экономические показатели для решения прикладных финансово-экономических задач. При написании работы проявил творческий подход.
Оценка 4 (хорошо)	Использует технологию работы с большими данными при принятии организационных и экономических решений и обосновывает финансово-экономические показатели для решения прикладных финансово-экономических задач с недочетами. Работа написана на хорошем уровне.
Оценка 3 (удовлетворительно)	Использует технологию работы с большими данными при принятии организационных и экономических решений и обосновывает финансово-экономические показатели для решения прикладных финансово-экономических задач, допуская серьезные ошибки. Работа написана на низком уровне.
Оценка 2 (неудовлетворительно)	Не использует технологию работы с большими данными при принятии организационных и экономических решений и не обосновывает финансово-экономические показатели для решения прикладных финансово-экономических задач. Работа написана на неудовлетворительном уровне.

5. Методические указания для обучающихся по освоению дисциплины Самостоятельная работа

Самостоятельная работа – крайне важный элемент подготовки в процессе обучения. Получить всесторонние знания, ограничиваясь при этом только прослушиванием лекций и посещением семинарских занятий, невозможно.

Кроме того, понятийный аппарат курса разнообразен, объемен, что требует специальной работы для их усвоения. Важным условием успешной самостоятельной работы являются консультации преподавателя и тщательная подготовка к практическим занятиям.

Цель самостоятельной работы по изучению учебного материала – формирование навыков самостоятельного отбора и изучения учебной литературы, интернет-источников, материалов периодических изданий, их анализа и осмысления. В результате этой работы должны научиться понимать логику научного исследования, критически анализировать существующие в научной литературе точки зрения и на этой основе формировать собственную позицию по рассматриваемому вопросу.

Самостоятельная работа по дисциплине играет важную роль в ходе всего учебного процесса. Методические материалы и рекомендации готовятся преподавателем и выдаются обучающемуся.

Работа на семинарах и лекциях

Во время лекции обучающийся должен вести краткий конспект. Работа с конспектом лекций предполагает просмотр конспекта в тот же день после занятий. При этом необходимо пометить материалы конспекта, которые вызывают затруднения для понимания.

Обучающийся должен стараться найти ответы на затруднительные вопросы, используя рекомендуемую литературу. Если ему самостоятельно не удалось разобраться в материале, необходимо сформулировать вопросы и обратиться за помощью к преподавателю на консультации или ближайшей лекции. Обучающемуся необходимо регулярно отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.

При подготовке к практическому занятию необходимо помнить, что та или иная дисциплина тесно связана с ранее изучаемыми курсами. Более того, именно синтез полученных ранее знаний и текущего материала по курсу делает подготовку результативной и всесторонней.

На семинарских занятиях обучающийся должен уметь последовательно излагать свои мысли и аргументированно их отстаивать.

Для достижения этой цели необходимо:

- 1) ознакомиться с соответствующей темой программы дисциплины;
- 2) осмыслить круг изучаемых вопросов и логику их рассмотрения;
- 3) изучить рекомендованную литературу по данной теме;
- 4) тщательно изучить лекционный материал;
- 5) ознакомиться с вопросами очередного семинарского занятия.

Изучение вопросов очередной темы требует глубокого усвоения теоретических основ дисциплины, раскрытия сущности основных экономических категорий, проблемных аспектов темы и анализа фактического материала.

Методические рекомендации по подготовке к промежуточной аттестации

При подготовке к промежуточной аттестации ознакомьтесь со списком представленных вопросов. Формулируйте ответ с точки зрения применения различных методов анализа данных. Необходимо дать аргументированный ответ, подтверждающий уровень освоения компетенции.

6. Учебная литература и ресурсы информационно-телекоммуникационной сети "Интернет", включая перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

6.1. Основная литература.

1. Вайл, Питер Цифровая трансформация бизнеса: Изменение бизнес-модели для организации нового поколения / Питер Вайл, Стефани Ворнер ; перевод И. Окунькова. — Москва : Альпина Пабlishер, 2019. — 264 с. — ISBN 978-5-9614-2184-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/82656.html>
2. Миркин, Б. Г. Введение в анализ данных : учебник и практикум / Б. Г. Миркин. — Москва : Издательство Юрайт, 2020. — 174 с. — (Высшее образование). — ISBN 978-5-9916-5009-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450262>

6.2. Дополнительная литература.

1. Бессмертный, И. А. Системы искусственного интеллекта : учебное пособие для среднего профессионального образования / И. А. Бессмертный. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 157 с. — (Профессиональное образование). — ISBN 978-5-534-11361-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/455735> (дата обращения: 25.02.2021).
2. Глотова, М. Ю. Математическая обработка информации : учебник и практикум для вузов / М. Ю. Глотова, Е. А. Самохвалова. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 301 с. — (Высшее образование). — ISBN 978-5-534-13622-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/466129> (дата обращения: 25.02.2021).
3. Кремер, Н. Ш. Эконометрика : учебник и практикум для вузов / Н. Ш. Кремер, Б. А. Путко ; под редакцией Н. Ш. Кремера. — 4-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 308 с. — (Высшее образование). — ISBN 978-5-534-08710-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/449750> (дата обращения: 25.02.2021).
4. Мойзес, Б. Б. Статистические методы контроля качества и обработка экспериментальных данных : учебное пособие для вузов / Б. Б. Мойзес, И. В. Плотникова, Л. А. Редько. — 2-е изд. — Москва : Издательство Юрайт, 2020. — 118 с. — (Высшее

образование). — ISBN 978-5-534-11906-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/457092> (дата обращения: 25.02.2021).

5. Нестеров, С. А. Базы данных : учебник и практикум для вузов / С. А. Нестеров. — Москва : Издательство Юрайт, 2020. — 230 с. — (Высшее образование). — ISBN 978-5-534-00874-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450772> (дата обращения: 25.02.2021).

6. Парфенов, Ю. П. Постреляционные хранилища данных : учебное пособие для вузов / Ю. П. Парфенов ; под научной редакцией Н. В. Папуловской. — Москва : Издательство Юрайт, 2020. — 121 с. — (Высшее образование). — ISBN 978-5-534-09837-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/453758>

6.3. Учебно-методическое обеспечение самостоятельной работы.

Не предусмотрено.

6.4. Нормативные правовые документы.

1. ФЗ «О Центральном Банке Российской Федерации (Банке России)» от 10 июля 2002г. №86 - ФЗ. «Собрание законодательства РФ», 15.07.2002, №28, ст. 2790.

2. ФЗ «О банках и банковской деятельности» от 02 декабря 1990г. №395-1. «Собрание законодательства РФ», 05.02.1996, N 6, ст. 492.

3. ФЗ «О валютном регулировании и валютном контроле» от 10 декабря 2003 г. №173-ФЗ. «Собрание законодательства РФ», 15.12.2003, N 50, ст. 4859.

4. ФЗ «Об электронной подписи» от 6 апреля 2011 г. №63-ФЗ. «Собрание законодательства РФ», 11.04.2011, N 15, ст. 2036.

5. ФЗ «О национальной платежной системе» от 27 июня 2011 г. №161-3. «Собрание законодательства РФ», 04.07.2011, N 27, ст. 3872.

6. ФЗ «О персональных данных» от 27 июля 2006 года № 152-ФЗ. «Собрание законодательства РФ», 31.07.2006, N 31 (1 ч.), ст. 3451.

7. ФЗ «О противодействии легализации (отмыванию) доходов, полученных преступным путем, и финансированию терроризма» от 7 июля 2001 года № 115-ФЗ. «Собрание законодательства РФ», 13.08.2001, № 33 (часть I), ст. 3418.

6.5. Интернет-ресурсы.

1. [https:// habr.com](https://habr.com)

2. <https://rb.ru>

3. [https:// proglib.io](https://proglib.io)- Официальный сайт журнала ForkLog

4. [https:// vc.ru](https://vc.ru)

5. [https:// tproger.ru](https://tproger.ru)

6.6. Иные источники

1. Горбаченко, В. И. Интеллектуальные системы: нечеткие системы и сети : учебное пособие для вузов / В. И. Горбаченко, Б. С. Ахметов, О. Ю. Кузнецова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 105 с. — (Высшее образование). — ISBN 978-5-534-08359-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/453629> (дата обращения: 25.02.2021).
2. Загорулько, Ю. А. Искусственный интеллект. Инженерия знаний : учебное пособие для вузов / Ю. А. Загорулько, Г. Б. Загорулько. — Москва : Издательство Юрайт, 2020. — 93 с. — (Высшее образование). — ISBN 978-5-534-07198-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/455500> (дата обращения: 25.02.2021).
3. Кудрявцев, В. Б. Интеллектуальные системы : учебник и практикум для вузов / В. Б. Кудрявцев, Э. Э. Гасанов, А. С. Подколзин. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 165 с. — (Высшее образование). — ISBN 978-5-534-07779-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/452226> (дата обращения: 25.02.2021).
4. Маркин, А. В. Программирование на SQL : учебное пособие для среднего профессионального образования / А. В. Маркин. — Москва : Издательство Юрайт, 2020. — 435 с. — (Профессиональное образование). — ISBN 978-5-534-11093-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/456926> (дата обращения: 25.02.2021).
5. Новиков, Ф. А. Символический искусственный интеллект: математические основы представления знаний : учебное пособие для вузов / Ф. А. Новиков. — Москва : Издательство Юрайт, 2020. — 278 с. — (Высшее образование). — ISBN 978-5-534-00734-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/451447> (дата обращения: 25.02.2021).
6. Основы математической обработки информации : учебник и практикум для вузов / Н. Л. Стефанова, Н. В. Кочуренко, В. И. Снегурова, О. В. Харитонова ; под общей редакцией Н. Л. Стефановой. — Москва : Издательство Юрайт, 2020. — 218 с. — (Высшее образование). — ISBN 978-5-534-01267-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450842> (дата обращения: 25.02.2021).
7. Станкевич, Л. А. Интеллектуальные системы и технологии : учебник и практикум для вузов / Л. А. Станкевич. — Москва : Издательство Юрайт, 2020. — 397 с. — (Высшее образование).

- образование). — ISBN 978-5-534-02126-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450773> (дата обращения: 25.02.2021).
8. Третьяк, Л. Н. Основы теории и практики обработки экспериментальных данных : учебное пособие для вузов / Л. Н. Третьяк, А. Л. Воробьев ; под общей редакцией Л. Н. Третьяк. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 237 с. — (Высшее образование). — ISBN 978-5-534-08623-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454093> (дата обращения: 25.02.2021).
9. Черткова, Е. А. Статистика. Автоматизация обработки информации : учебное пособие для вузов / Е. А. Черткова ; под общей редакцией Е. А. Чертковой. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 195 с. — (Высшее образование). — ISBN 978-5-534-01429-7. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/452447> (дата обращения: 25.02.2021).

7. Материально-техническая база, информационные технологии, программное обеспечение и информационные справочные системы

Базы данных, информационно-справочные и поисковые системы

1. www.biblio-online.ru – Электронно-библиотечная система [ЭБС] Юрайт;
2. <http://www.iprbookshop.ru> – Электронно-библиотечная система [ЭБС] «Iprbooks»
3. <https://e.lanbook.com> - Электронно-библиотечная система [ЭБС] «Лань».
4. <http://elibrary.ru/> - Научная электронная библиотека Elibrary.ru.
5. <https://new.znaniy.com> Электронно-библиотечная система [ЭБС] «Znaniy.com».
6. <https://dlib.eastview.com> – Информационный сервис «East View».
7. <https://www.jstor.org> - Jstor. Полные тексты научных журналов и книг зарубежных издательств.
8. <https://elibrary.worldbank.org> - Электронная библиотека Всемирного Банка.
9. <https://link.springer.com> - Полнотекстовые политематические базы академических журналов и книг издательства Springer.
10. <https://ebookcentral.proquest.com> - Ebook Central. Полные тексты книг зарубежных научных издательств.
11. <https://www.oxfordhandbooks.com> - Доступ к полным текстам справочников Handbooks издательства Oxford по предметным областям: экономика и финансы, право, бизнес и управление.
12. <https://journals.sagepub.com> - Полнотекстовая база научных журналов академического

издательства Sage.

13. Справочно-правовая система «КонсультантПлюс».

14. Электронный периодический справочник «Гарант».

Программные, технические и электронные средства обучения и контроля знаний.

Для проведения занятий по дисциплине необходимо материально-техническое обеспечение учебных аудиторий (наглядными материалами, экраном, мультимедийным проектором с ноутбуками (ПК) для презентации учебного материала, выходом в сеть Интернет, программными продуктами Microsoft Office (Excel, Word, PowerPoint)) в зависимости от типа занятий: семинарского и лекционного типов, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Для самостоятельной работы обучающимся необходим доступ в читальные залы библиотеки и/или помещение, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет», доступ в электронную информационно-образовательную среду организации и ЭБС.