

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных**

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: Zh0btSfg2O8J16S/Xx6ChgLDR4Em3jTs
Дата: 26.11.2025 г.

С.С. Маслов

Инженерия данных

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике
и финансах»

Профиль:

«Инженерия данных»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 03 от 16.12.2025 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 12 от 03.12.2025 г.)*

© Москва 2026

СОДЕРЖАНИЕ

1.	Наименование дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3.	Место дисциплины в структуре образовательной программы	8
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	8
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	9
5.1.	Содержание дисциплины	9
5.2.	Учебно-тематический план	12
5.3.	Содержание семинаров	13
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	16
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	16
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	18
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	20
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	31
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	32
10.	Методические указания для обучающихся по освоению дисциплины	33
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	34
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	35

1. Наименование дисциплины

«Инженерия данных».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПKN-3	Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней	Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов	знать: модели и форматы данных: реляционная модель, документная модель, колоночные форматы (Parquet, ORC), форматы для потоковой передачи (Avro). Принципы хранения данных: Data Lake (сырые/очищенные/витринные слои), Data Warehouse (схемы «звезда», «снежинка»); критерии выбора СУБД и систем хранения: OLTP vs OLAP, особенности row-oriented vs column-oriented хранилищ; методы оценки сложности и производительности операций в распределенных системах (шардирование, партиционирование, индексный поиск); уметь: анализировать шаблоны доступа к данным (частота чтения/записи, типы запросов) и на их основе выбирать подходящие системы хранения; проектировать схему хранения (партиционирование, кластеризация) для оптимизации выполнения типовых запросов; выбирать форматы сериализации и хранения данных (Parquet, Avro) исходя из требований к скорости чтения/записи и сжатию; оценивать производительность запросов и операций ETL, выявляя «узкие места» на уровне данных;
		Собирает, формулирует,	знать: специфические требования к

		систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе	системам данных: объем, скорость поступления (velocity), разнообразие (variety), задержка обработки (latency), согласованность (consistency), свежесть данных (freshness);архитектурные паттерны для данных: Lambda, Kappa, Data Lakehouse, Medallion Architecture;критерии выбора компонентов технологического стека: пакетная (Spark) vs потоковая (Flink) обработка, выбор оркестратора, системы мониторинга;принципы документирования архитектуры данных (Data Flow Diagrams, диаграммы пайплайнов); уметь: выявлять и формулировать требования к данным: источники, частота обновления, целевые потребители (аналитики, ML-модели), необходимые атрибуты качества;переводить бизнес-требования в технические спецификации для пайплайнов (SLA по времени выполнения, допустимый процент потерь); выбирать и обосновывать архитектурный паттерн (например, Kappa вместо Lambda) на основе требований к консистентности, простоте поддержки и задержке;проектировать высокоуровневую схему пайплайнов данных, включая компоненты для приема, обработки, хранения и сервировки данных;
		Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии	знать: принципы проектирования потоков данных: идемпотентность, повторная запускаяемость (replayability), обработка ошибок, стратегии повторных попыток (retry);практики обеспечения надежности: мониторинг, логирование, алертинг, версионирование данных и кода;принципы «Инфраструктуры как кода» (IaC) и контейнеризации применительно к задачам

			инженерии данных; методы декомпозиции сложного пайплайна на независимые, переиспользуемые этапы (tasks, jobs); уметь: проектировать DAG (Directed Acyclic Graph) в оркестраторе (например, Airflow), декомпозируя ETL-процесс на логические, перезапускаемые задачи; инкапсулировать логику трансформации данных в переиспользуемые модули (Python-пакеты, Docker-образы, dbt-модели); реализовывать механизмы обработки сбоев, проверки качества данных и уведомлений в пайплайне; конфигурировать пайплайны с помощью параметров и переменных окружения для обеспечения гибкости и переносимости между средами (dev/staging/prod);
ПКП-2	Способность разрабатывать, согласовывать и управлять исполнением технического задания и технического проекта с использованием технологий больших данных	Работает со стандартами, в том числе адаптирует стандарты для специфических требований больших данных	знать: ключевые российские и международные стандарты и нормативные документы в области разработки программного обеспечения и информационных систем (например, ГОСТ 34-й серии, ГОСТ Р ИСО/МЭК 12207, IEEE 830); специфику проектов больших данных, отличающую их от классических ИТ-проектов (масштабируемость, разнообразие данных, требования к производительности и времени обработки); принципы адаптации общих стандартов под особенности технологий больших данных (например, специфика требований к надежности и безопасности для распределенных систем типа Apache Spark или Kafka); уметь: анализировать требования проекта больших данных и определять применимые разделы стандартов; адаптировать шаблоны и структуру стандартной технической документации (ТЗ, ТП) под специфику задач big data (например,

			включение разделов по архитектуре данных, стратегиям масштабирования, выбору форматов хранения); обосновывать выбор и адаптацию стандартов при согласовании документации с заказчиком или смежными подразделениями;
		Разрабатывает технические задания и технические проекты для технологий больших данных	знать: структуру, содержание и логику построения документов «Техническое задание (ТЗ)» и «Технический проект (ТП)» в соответствии с установленными в организации/отрасли шаблонами; методы и техники выявления, анализа и формализации требований (функциональных, нефункциональных, бизнес-требований) к системам больших данных; принципы архитектурного проектирования систем big data, включая выбор компонентов (хранилища, процессоры, оркестраторы), проектирование потоков данных и схем взаимодействия; критерии приемки (Acceptance Criteria) и методы планирования испытаний для технологий больших данных; уметь: проводить интервью и workshops с заказчиками и стейкхолдерами для выявления и структурирования требований; разрабатывать ТЗ, включающее четкие цели, границы проекта, функциональные требования (источники данных, процессы трансформации, выходные витрины) и нефункциональные требования (производительность, доступность, безопасность); разрабатывать ТП, включающий обоснование выбора технологического стека, архитектурные схемы (логическую и физическую), план миграции данных и план тестирования; формулировать конкретные, измеримые и достижимые критерии успеха

			проекта и приемки результатов.
		Реализует управление рабочими проектами технологической инфраструктуры больших данных	знать: основы управления проектами (Project Management): процессы инициации, планирования, исполнения, контроля и завершения в контексте ИТ-проектов; основные методологии управления (Waterfall, Agile/Scrum, Kanban) и их применимость к проектам в области данных; методы оценки трудозатрат, сроков и рисков (Risk Management) при реализации проектов технологической инфраструктуры; принципы управления изменениями (Change Management) и конфигурациями в ходе выполнения проекта; уметь: разрабатывать и поддерживать план проекта (Project Plan), включающий календарный график работ, распределение ресурсов и бюджет; выявлять, анализировать и планировать реагирование на риски проекта (создавать Risk Register); организовывать и контролировать работу проектной команды, проводить регулярные статус-встречи (stand-ups), отслеживать прогресс по метрикам (KPI); управлять изменениями в проекте: обрабатывать запросы на изменение (Change Request), оценивать их влияние на сроки, бюджет и содержание проекта; формировать отчетность о ходе проекта для заказчика и руководства;

3. Место дисциплины в структуре образовательной программы

Дисциплина «Инженерия данных» относится к «Профилю "Инженерия данных"».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 5 (в часах)	Семестр 6 (в часах)
Общая трудоёмкость дисциплины	6/216	108	108
Контактная работа- Аудиторные занятия	100	50	50
Лекции	32	16	16
Семинары, практические занятия	68	34	34
Самостоятельная работа	116	58	58
В том числе курсовая работа/курсовой проект	24	-	-
Вид текущего контроля	Проектная работа	Проектная работа	
Вид промежуточной аттестации	Зачет, Экзамен	Зачет	Экзамен

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Введение в инженерии данных

Что такое инженерия данных? Роль в современном мире. Отличия от Data Science, анализа данных и программной инженерии. Жизненный цикл данных. Обзор современного стека технологий (Open Source vs. Cloud). Ключевые понятия: ETL, ELT, Data Warehouse, Data Lake, Pipeline.

Тема 2. Работа с данными на языке Python

Python для инженерии данных. Обзор ключевых библиотек (Pandas, NumPy, Requests). Паттерны обработки данных. Получение данных (Extract): API (REST, GraphQL), веб-скрапинг (BeautifulSoup, Scrapy), файлы (JSON, CSV, Parquet, Avro).

Преобразование данных (Transform): Очистка, валидация, агрегация, обработка пропусков и аномалий. Pandas в деталях. Производительность и оптимизация. Векторизация, эффективное использование датафреймов. Введение в Apache Arrow.

Тема 3. Базы данных и SQL для инженерии данных

Реляционные БД (PostgreSQL). Углубленный SQL: оконные функции, CTE, сложные джойны. Проектирование схемы для хранения сырых данных. Оптимизация запросов: индексы, планировщик запросов (EXPLAIN ANALYZE).

Нереляционные (NoSQL) БД. Обзор видов: документные (MongoDB), колоночные (ClickHouse), ключ-значение (Redis). Шаблоны применения.

Загрузка данных. Паттерны загрузки: полная, инкрементальная, CDC (Change Data Capture). Инструменты (обзорно).

Тема 4. Основы распределенных систем и введение в Big Data

Проблемы масштабирования. Почему одной БД недостаточно? Принципы распределенных систем: надежность, масштабируемость, отказоустойчивость. Модель MapReduce. Вычислительный кластер. Введение в Apache Spark: архитектура, RDD, Spark SQL.

Data Lakes. Концепция и архитектура. HDFS, S3. Форматы хранения в Data Lake (Parquet, ORC) и их важность. Data Warehouses. Концепция и отличия от Data Lake. Схемы "Звезда" и "Снежинка". Обзор современных DWH (Google BigQuery, Amazon Redshift, Snowflake и российские аналоги).

Тема 5. Комплексные потоки данных и введение в оркестрацию

Сборка end-to-end пайплайна. Логирование, обработка ошибок, мониторинг. Введение в оркестрацию. Зачем нужны планировщики? Обзор Apache Airflow.

Тема 6. Масштабирование потоков данных. Сложные вопросы оркестрации

Apache Airflow подробно. Ключевые концепции: DAG, Operator, Sensor, XCom. Лучшие практики использования. Шаблоны проектирования потоков данных. Обработка зависимостей, перезапуски, бэк-филлы. Мониторинг и оповещения в Airflow. Интеграция с мессенджерами. Альтернативы Airflow (обзорно Luigi, Prefect, Dagster).

Тема 7. Поточковая обработка данных (Streaming)

Введение в стриминг. Use cases. Отличия от пакетной обработки. Концепции: ивенты, сообщения, брокеры. Брокеры сообщений: Apache Kafka (архитектура, темы, партиции, репликация). Альтернативы (RabbitMQ, Pulsar).

Обработка потоков с помощью Apache Spark Streaming (Structured Streaming). Паттерны потоковой обработки (микробатчи, оконные агрегаты, соединение потоков).

Тема 8. Качество данных, тестирование и метрики

Data Quality. Архитектура системы мониторинга качества данных. Где и как встраивать проверки (в пайплайн vs отдельный сервис). Тестирование в инженерии данных.

Тема 9. Безопасность данных и управление доступом

Основы информационной безопасности. Шифрование. Архитектура управления доступом в облаке (YC IAM). Сервис-аккаунты, роли. Архитектура безопасного потока данных (пайплайна). Сети (VPC, Security Groups), управление секретами (Yandex Lockbox), безопасное хранение ключей.

Тема 10. Облачная инженерия данных

Сравнительная архитектура российских облаков: Yandex Cloud, SberCloud, VK Cloud Services. Ключевые сервисы для данных. Архитектура "как код" (IaC). Введение в Terraform для провижининга всей инфраструктуры пайплайна в YC.

Контейнеризация пайплайнов (Docker). Интеграция с Yandex Container Registry.
Современные тренды и итоговая архитектура: Data Mesh на российском стеке (обзорно).
Сборка полной картины из изученных компонентов.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Введение в инженерию данных	10	6	2	4	4
2	Работа с данными на языке Python	28	12	4	8	16
3	Базы данных и SQL для инженерии данных	26	12	4	8	14
4	Основы распределенных систем и введение в Big Data	28	12	4	8	16
5	Комплексные потоки данных и введение в оркестрацию	16	8	2	6	8
6	Масштабирование потоков данных. Сложные вопросы оркестрации	24	12	4	8	12
7	Потоковая обработка данных (Streaming)	26	12	4	8	14
8	Качество данных, тестирование и метрики	18	8	2	6	10
9	Безопасность данных и управление доступом	12	6	2	4	6
10	Облачная инженерия данных	28	12	4	8	16
	Итого	216	100	32	68	116

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Введение в инженерии данных	Чем ежедневные задачи инженера данных фундаментально отличаются от задач data-сайентиста? Как бы вы объяснили разницу между ETL и ELT бизнес-аналитику, который не является техническим специалистом? Кейс: У стартапа есть накопленные данные в CSV-файлах и потребность в первых дашбордах. С чего должен начать инженер данных и какую простейшую архитектуру предложить? Какие риски возникают при отсутствии инженера данных в команде, где аналитики работают с данными напрямую?	Семинар
Работа с данными на языке Python	В каких сценариях Pandas является идеальным инструментом, а когда его использование становится проблемой? Вы получаете из API JSON с вложенной структурой. Какие стратегии его "распаковки" и нормализации в "плоскую" таблицу вы знаете? Какие у них плюсы и минусы? При парсинге веб-сайта вы столкнулись с блокировками по IP. Какие стратегии обхода этой проблемы можно реализовать на этапе разработки пайплайна? Как вы будете обрабатывать пропущенные значения в колонке, которая критична для последующего анализа? Ваши действия будут одинаковы для числовых и категориальных данных?	Семинар, выполнение практических заданий
Базы данных и SQL для инженерии данных	Когда вы предпочтете использовать оконные функции вместо группировок и вложенных запросов? Приведите примеры. Кейс: Пайплайн, который раньше выполнялся 10 минут, теперь работает 2 часа. С чего вы начнете расследование и какие инструменты SQL используете для диагностики? Объясните на примере, в чем разница между полной (Full) и инкрементальной (Incremental) загрузкой данных. Какие бизнес-сценарии оправдывают сложность реализации инкрементальной загрузки? Когда вы выбрали бы документную NoSQL Базу данных (например, MongoDB) вместо реляционной PostgreSQL для хранения данных пайплайна?	Семинар, выполнения практических заданий
Основы распределенных систем и введение в	Объясните, почему CAP-теорема важна при выборе базы данных для распределенной системы. Приведите примеры CP- и AP-систем. Кейс: Вам	Семинар

Big Data	нужно обработать 1 ТБ логов. Объясните, почему и когда вы выбрали бы PySpark вместо Pandas. Что, кроме объема, повлияет на ваше решение? В чем философская разница между Data Lake и Data Warehouse? Почему современная архитектура часто включает оба этих слоя? Какие преимущества дает хранение данных в колоночных форматах (Parquet/ORC) именно для задач аналитики? Есть ли у них недостатки?	
Комплексные потоки данных и введение в оркестрацию	Какие критерии вы заложите в техзадание для своего пайплайна, чтобы он считался надежным и готовым к продакшену? (Подумайте об ошибках, мониторинге, данных). В чем принципиальная разница между запуском скрипта по расписанию в Cron и использованием оркестратора типа Apache Airflow? Как бы вы спроектировали пайплайн, чтобы он мог корректно перезапуститься с места сбоя, а не с начала?	Семинар
Масштабирование потоков данных. Сложные вопросы оркестрации	В каких ситуациях задача в Airflow должна быть разбита на несколько более мелких задач, а когда лучше сделать одну большую? Как вы организуете передачу небольшого объема данных (например, даты обработки) между задачами в DAG? Какие способы вы знаете и какой самый надежный? Кейс: Ваш DAG начал падать из-за временной недоступности API. Какие механизмы Airflow вы используете, чтобы сделать пайплайн устойчивым к таким сбоям?	Семинар
Потоковая обработка данных (Streaming)	Опишите бизнес-кейс, где потоковая обработка данных критически важна и принесет реальную бизнес-ценность. А где можно обойтись пакетной обработкой? В чем ключевое различие между архитектурными паттернами Lambda и Карра? В каком сценарии Карра будет явным победителем? Какие проблемы искажения данных (data skew) могут возникнуть в потоковой обработке и как их можно избежать или маскировать?	Семинар
Качество данных, тестирование и метрики	Какие 3 метрики качества данных вы считаете самыми важными для мониторинга в первую очередь? Обоснуйте свой выбор. Где в пайплайне вы будете размещать проверки качества данных: сразу после извлечения, после трансформации или перед загрузкой? Почему? Кейс: Вы обнаружили, что 5% строк в ключевой таблице внезапно содержат NULL в важной колонке. Каковы ваши немедленные действия и долгосрочные решения?	Семинар
Безопасность данных и управление	Как принцип наименьших привилегий (Principle of Least Privilege) должен применяться к сервис-	Семинар

доступом	аккаунту в Yandex Cloud, который используется для запуска пайплайна? Какие данные в вашем пайплайне стоило бы зашифровать не только "в покое" (at-rest), но и "в движении" (in-transit)? Все ли? Кто и по какому принципу должен иметь доступ к сырым данным в Data Lake, а кто — к агрегированным витринам в DWH?	
Облачная инженерия данных	Инфраструктура как код (IaC) на Terraform — это обязательный навык для инженера данных или прерогатива DevOps? Какие экономические и технические факторы будут влиять на ваш выбор между управляемым сервисом (например, YC Managed Service for Kafka) и развертыванием кластера самостоятельно на виртуальных машинах? Как выбор архитектуры повлияет на стоимость поддержки и масштабирования системы через 2 года?	Практическое задание

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Введение в инженериию данных	Адаптация стандартов технического проектирования (ГОСТ 34-й серии) для проектов в области больших данных. Базовые сервисы российских облачных платформ (Yandex Cloud, SberCloud) для инженерии данных: сравнительный обзор.	Работа с литературой, подготовка докладов
Работа с данными на языке Python	Методы работы со сложными вложенными структурами JSON (библиотека <code>pd.json_normalize</code>). Сравнительный обзор библиотек и фреймворков для веб-скрапинга (BeautifulSoup, Scrapy, Selenium).	Работа с документацией, написание небольших программ
Базы данных и SQL для инженерии данных	Индексы на выражениях (function-based indexes) в PostgreSQL: принцип работы и сценарии применения. Паттерн Slowly Changing Dimensions (SCD): концепция и реализация. Транзакции в PostgreSQL: обеспечение свойств ACID.	Работа с документацией и литературой, подготовка докладов.
Основы распределенных систем и введение в Big Data	Архитектура распределенной файловой системы Apache HDFS. Принципы партиционирования данных в Apache Spark и формате Parquet для оптимизации производительности. Сравнение управляемых колоночных хранилищ данных (ClickHouse, Google BigQuery): архитектура и модель использования.	Работа с документацией и литературой
Комплексные потоки данных и введение в оркестрацию	Базовые концепции Apache Airflow: DAG, Operator, Task, Task Instance. Механизмы расписания (scheduling) в оркестраторах: cron expression vs timedelta.	Работа с литературой, документацией
Масштабирование потоков данных. Сложные вопросы оркестрации	Назначение и сценарии использования основных типов Operators в Apache Airflow (BashOperator, PythonOperator, Sensor). Механизм XCom для передачи данных между задачами в Airflow: возможности и ограничения. Управление конфигурацией и секретами в Airflow: Connections и Variables.	Работа с документацией, подготовка докладов
Потоковая обработка данных (Streaming)	Архитектура Apache Kafka: роли Producer, Consumer, Broker, понятия Topic, Partition и Offset.	Работа с документацией

	Семантики доставки сообщений в распределенных системах (at-most-once, at-least-once, exactly-once). Концепция водяных знаков (Watermarks) в потоковой обработке для решения проблемы «опоздавших данных».	
Качество данных, тестирование и метрики	Библиотека pytest для модульного тестирования функций трансформации данных. Валидация данных на основе схемы (Schema Validation) в контексте инженерии данных. Инструмент Great Expectations для декларативной проверки качества данных.	Работа с документацией, написание небольших программ
Безопасность данных и управление доступом	Модель управления доступом (IAM) в Yandex Cloud: иерархия ресурсов и ролей. Методы шифрования данных: server-side vs client-side encryption, сценарии применения. Принципы сетевой безопасности для data-платформ: изоляция компонентов, использование частных подсетей и бастий-хостов.	Работа с документацией
Облачная инженерия данных	Основы Terraform для управления облачной инфраструктурой (на примере провайдера Yandex Cloud). Сравнение managed-сервисов для инженерии данных на российских облачных платформах. Концепция Data Mesh: основные принципы и их влияние на роль инженера данных.	Работа с документацией

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерная тематика проектной работы

1. Проект ETL: разработка процесса извлечения, преобразования и загрузки данных из разнородных источников в единую базу данных для дальнейшего анализа.
2. Разработка Data Warehouse: проектирование хранилища данных для крупного предприятия с учётом особенностей предметной области и бизнес-задач организации.
3. Создание OLAP-куба: построение куба данных для оперативного анализа продаж продукции компании, включающего срезы по регионам, категориям товаров и временным периодам.
4. Работа с большими данными: обработка большого объёма структурированных и полуструктурированных данных с использованием технологий Hadoop или Spark.
5. Автоматизация мониторинга качества данных: создание системы мониторинга целостности и согласованности данных, интегрированной с процессами обработки данных.
6. Data Lake vs Data Warehouse: сравнение подходов к хранению и обработке данных, демонстрация преимуществ каждого подхода применительно к конкретной задаче бизнеса.
7. Анализ производительности запросов: оптимизация SQL-запросов и индексирование таблиц для повышения эффективности работы приложений с базой данных.
8. Система управления метаданными: разработка инструмента для сбора, хранения и визуализации метаданных, используемых в процессах аналитики и принятия решений.
9. Интеграция BI-платформы: интеграция корпоративной системы бизнес-аналитики с источниками данных, настройка регулярных отчётов и панелей инструментов для руководства компании.

10. Обеспечение безопасности данных: реализация мер защиты конфиденциальных данных при передаче и хранении, включая шифрование, контроль доступа и аудит операций с информацией.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ПКН-3 Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней

1) Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: модели и форматы данных: реляционная модель, документная модель, колоночные форматы (Parquet, ORC), форматы для потоковой передачи (Avro). Принципы хранения данных: Data Lake (сырые/очищенные/витринные слои), Data Warehouse (схемы «звезда», «снежинка»); критерии выбора СУБД и систем хранения: OLTP vs OLAP, особенности row-oriented vs column-oriented хранилищ; методы оценки сложности и производительности операций в распределенных системах (шардирование, партиционирование, индексный поиск);

Уметь: анализировать шаблоны доступа к данным (частота чтения/записи, типы запросов) и на их основе выбирать подходящие системы хранения; проектировать схему хранения (партиционирование, кластеризация) для оптимизации выполнения типовых запросов; выбирать форматы сериализации и хранения данных (Parquet, Avro) исходя из требований к скорости чтения/записи и сжатию; оценивать производительность запросов и операций ETL, выявляя «узкие места» на уровне данных;

Типовые контрольные задания

Задание 1: Для паттерна запросов «агрегации по дате и категории» предложите схему партиционирования и ключи сортировки в колоночном хранилище для их ускорения.

Задание 2: Обоснуйте выбор между форматами Parquet и Avro для долгосрочного хранения сырых данных и для передачи сообщений в Kafka соответственно.

Задание 3: Дана потоковая система с высокими требованиями к записи. Сравните подходы к хранению данных (хранить сырые события, агрегированные окна) с точки зрения сложности и ресурсов.

Задание 4: Спроектируйте схему Data Warehouse (витрину) для анализа продаж, выбрав между схемой «звезда» и «снежинка» и обосновав выбор.

Задание 5: Проанализируйте план выполнения (EXPLAIN) SQL-запроса. Определите потенциальные проблемы (full scan, сложные джойны) и предложите оптимизации на уровне схемы данных или индексов.

2) Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: специфические требования к системам данных: объем, скорость поступления (velocity), разнообразие (variety), задержка обработки (latency), согласованность (consistency), свежесть данных (freshness); архитектурные паттерны для данных: Lambda, Kappa, Data Lakehouse, Medallion Architecture; критерии выбора компонентов технологического стека: пакетная (Spark) vs потоковая (Flink) обработка, выбор оркестратора, системы мониторинга; принципы документирования архитектуры данных (Data Flow Diagrams, диаграммы пайплайнов);

Уметь: выявлять и формулировать требования к данным: источники, частота обновления, целевые потребители (аналитики, ML-модели), необходимые атрибуты качества; переводить бизнес-требования в технические спецификации для пайплайнов (SLA по времени выполнения, допустимый процент потерь); выбирать и обосновывать архитектурный паттерн (например, Карра вместо Lambda) на основе требований к консистентности, простоте поддержки и задержке; проектировать высокоуровневую схему пайплайнов данных, включая компоненты для приема, обработки, хранения и сервировки данных;

Типовые контрольные задания

Задание 1: Сформулируйте нефункциональные требования (SLA) к пайплайну обновления дашборда в реальном времени: допустимая задержка, полнота данных, время на восстановление после сбоя.

Задание 2: На основе требований «обработка 1 ТБ логов в день с задержкой не более 2 часов» выберите и обоснуйте технологию обработки (например, Spark) и стратегию оркестрации.

Задание 3: Сравните архитектуры Lambda и Карра для задачи детекции аномалий в финансовых транзакциях. Какой паттерн предпочтительнее и почему?

Задание 4: Создайте диаграмму потоков данных (Data Flow Diagram) для системы рекомендаций, отобразив этапы от сбора сырых кликов до формирования предсказаний.

Задание 5: Проанализируйте новый бизнес-требование (например, добавить источник данных из соцсетей). Оцените его влияние на существующую архитектуру пайплайна.

3) Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы проектирования потоков данных: идемпотентность, повторная запускаяемость (replayability), обработка ошибок, стратегии повторных попыток (retry); практики обеспечения надежности: мониторинг, логирование, алертинг, версионирование данных и кода; принципы «Инфраструктуры как кода» (IaC) и контейнеризации применительно к задачам инженерии данных; методы декомпозиции сложного пайплайна на независимые, переиспользуемые этапы (tasks, jobs);

Уметь: проектировать DAG (Directed Acyclic Graph) в оркестраторе (например, Airflow), декомпозируя ETL-процесс на логические, перезапускаемые задачи; инкапсулировать логику трансформации данных в переиспользуемые модули (Python-пакеты, Docker-образы, dbt-модели); реализовывать механизмы обработки сбоев, проверки качества данных и уведомлений в пайплайне; конфигурировать пайплайны с помощью параметров и переменных окружения для обеспечения гибкости и переносимости между средами (dev/staging/prod);

Типовые контрольные задания

Задание 1: Спроектируйте DAG в Apache Airflow для ежедневного ETL-процесса. Выделите задачи, определите зависимости между ними и точки для возможных повторов (retry).

Задание 2: Создайте параметризованный dbt-модель для расчета агрегатов, которая может переиспользоваться для разных бизнес-юнитов или дат.

Задание 3: Реализуйте в пайплайне механизм проверки качества данных (например, проверку на наличие дубликатов или выход за границы), инкапсулировав логику в отдельный оператор/функцию.

Задание 4: Предложите дизайн для перезапуска (replay) пайплайна за произвольный исторический период. Как обеспечить идемпотентность при таком перезапуске?

Задание 5: Разработайте конфигурацию (YAML/JSON) для ETL-задачи, позволяющую гибко задавать источник, набор трансформаций и целевое хранилище без изменения кода.

**ПКП-2 Способность разрабатывать, согласовывать и управлять исполнением
технического задания и технического проекта с использованием технологий больших
данных**

1) Работает со стандартами, в том числе адаптирует стандарты для специфических требований больших данных

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: ключевые российские и международные стандарты и нормативные документы в области разработки программного обеспечения и информационных систем (например, ГОСТ 34-й серии, ГОСТ Р ИСО/МЭК 12207, IEEE 830); специфику проектов больших данных, отличающую их от классических ИТ-проектов (масштабируемость, разнообразие данных, требования к производительности и времени обработки); принципы адаптации общих стандартов под особенности технологий больших данных (например, специфика требований к надежности и безопасности для распределенных систем типа Apache Spark или Kafka);

Уметь: анализировать требования проекта больших данных и определять применимые разделы стандартов; адаптировать шаблоны и структуру стандартной технической документации (ТЗ, ТП) под специфику задач big data (например, включение разделов по архитектуре данных, стратегиям масштабирования, выбору форматов хранения); обосновывать выбор и адаптацию стандартов при согласовании документации с заказчиком или смежными подразделениями;

Типовые контрольные задания

Задание 1: Проведите сравнительный анализ требований ГОСТ 34.602 и специфики проекта по построению потокового пайплайна. Выделите разделы стандарта, требующие наибольшей адаптации.

Задание 2: На основе шаблона технического задания (ТЗ) по ГОСТу добавьте разделы, специфичные для проекта Data Lake (например, требования к зонам хранения, метаданным, форматам файлов).

Задание 3: Разработайте фрагмент технического проекта (ТП) с обоснованием выбора технологии (например, Apache Spark вместо однопоточной обработки), ссылаясь на нефункциональные требования к производительности и масштабируемости.

Задание 4: Проанализируйте кейс с проблемой в пайплайне данных (например, потерю данных). Предложите, какие положения стандартов по управлению качеством и надежностью были нарушены и как их адаптировать для big data.

Задание 5: Подготовьте краткий регламент (инструкцию) по оформлению технической документации для проектов больших данных в гибкой (Agile) среде разработки.

2) Разрабатывает технические задания и технические проекты для технологий больших данных

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: структуру, содержание и логику построения документов «Техническое задание (ТЗ)» и «Технический проект (ТП)» в соответствии с установленными в организации/отрасли шаблонами; методы и техники выявления, анализа и формализации требований (функциональных, нефункциональных, бизнес-требований) к системам больших данных; принципы архитектурного проектирования систем big data, включая выбор компонентов (хранилища, процессоры, оркестраторы), проектирование потоков данных и схем взаимодействия; критерии приемки (Acceptance Criteria) и методы планирования испытаний для технологий больших данных;

Уметь: проводить интервью и workshops с заказчиками и стейкхолдерами для выявления и структурирования требований; разрабатывать ТЗ, включающее четкие цели, границы проекта, функциональные требования (источники данных, процессы трансформации, выходные витрины) и нефункциональные требования (производительность, доступность, безопасность); разрабатывать ТП, включающий обоснование выбора технологического стека, архитектурные схемы (логическую и физическую), план миграции данных и план тестирования; формулировать конкретные, измеримые и достижимые критерии успеха проекта и приемки результатов.

Типовые контрольные задания

Задание 1: На основе описания бизнес-задачи (например, «Необходима система прогнозирования оттока клиентов») разработайте раздел «Цели и задачи» и перечень функциональных требований для ТЗ.

Задание 2: Для заданного нефункционального требования (например, «Обработка 1 ТБ данных за 1 час») подберите и обоснуйте в ТП технологию обработки (например, Apache Spark) и конфигурацию инфраструктуры.

Задание 3: Разработайте архитектурную блок-схему (диаграмму) системы в рамках ТП, отображающую потоки данных от источников через ETL/ELT до витрин и BI-инструментов.

Задание 4: Составьте план поэтапного внедрения (миграции) системы из ТП, указав последовательность работ, ресурсы и точки контроля.

Задание 5: Сформулируйте критерии приемки (Acceptance Criteria) для пайплайна данных, включая метрики качества данных (свежесть, полнота) и корректности обработки.

3) Реализует управление рабочими проектами технологической инфраструктуры больших данных

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основы управления проектами (Project Management): процессы инициации, планирования, исполнения, контроля и завершения в контексте ИТ-проектов; основные методологии управления (Waterfall, Agile/Scrum, Kanban) и их применимость к проектам в области данных; методы оценки трудозатрат, сроков и рисков (Risk Management) при реализации проектов технологической инфраструктуры; принципы управления изменениями (Change Management) и конфигурациями в ходе выполнения проекта;

Уметь: разрабатывать и поддерживать план проекта (Project Plan), включающий календарный график работ, распределение ресурсов и бюджет; выявлять, анализировать и планировать реагирование на риски проекта (создавать Risk Register); организовывать и контролировать работу проектной команды, проводить регулярные статус-встречи (stand-ups), отслеживать прогресс по метрикам (KPI); управлять изменениями в проекте: обрабатывать запросы на изменение (Change Request), оценивать их влияние на сроки, бюджет и содержание проекта; формировать отчетность о ходе проекта для заказчика и руководства;

Типовые контрольные задания

Задание 1: На основе утвержденного ТЗ составьте план-график (Roadmap) проекта, используя метод декомпозиции работ (Work Breakdown Structure - WBS).

Задание 2: Выявите и оцените (по вероятности и влиянию) 3-5 ключевых рисков для проекта развертывания кластера обработки данных. Заполните матрицу рисков и предложите меры по их mitigation.

Задание 3: Смоделируйте ситуацию «запрос на изменение» от заказчика (например, добавление нового источника данных). Проанализируйте его влияние на сроки и бюджет, подготовьте решение.

Задание 4: На основе предоставленных данных о статусе задач (отставание, перерасход) подготовьте краткий аналитический отчет для руководства с предложениями по корректирующим действиям.

Задание 5: Разработайте чек-лист (план) мероприятий по завершению проекта (Project Closure), включающий действия по документированию, передаче в эксплуатацию и подведению итогов.

Примеры практико-ориентированных заданий

1. Разработка базовый уровень:

Напишите SQL-запрос для расчета скользящего среднего выручки по месяцам.

Реализуйте ETL-процесс для переноса данных из CSV-файла в базу данных с базовой валидацией.

Создайте простой DAG в Airflow с двумя последовательными задачами.

2. Разработка продвинутого уровня:

Реализуйте инкрементальную загрузку данных с обработкой изменений (CDC).

Настройте мониторинг качества данных в работающем пайплайне.

Реализуйте обработку потоковых данных с использованием Spark Streaming.

3. Технические кейсы:

Проанализируйте архитектуру пайплайна и предложите оптимизации для улучшения производительности.

Разработайте стратегию обеспечения качества данных в распределенной системе.

Предложите решение для миграции устаревшего ETL-процесса на современную облачную платформу.

4. Управленческие кейсы:

Разработайте план перехода с пакетной на потоковую обработку данных.

Рассчитайте стоимость владения data-платформой в облаке за год.

Составьте техническое задание на разработку системы мониторинга данных.

Примерные вопросы для подготовки к зачету, экзамену

Примерные вопросы для подготовки к зачету

1. Архитектура и роли: Объясните разницу между ETL и ELT. На примере конкретного бизнес-кейса (например, построение дашборда для отдела маркетинга) обоснуйте, какой подход вы бы выбрали и почему.

2. Python для ETL: Опишите поэтапный процесс написания скрипта на Python (с использованием Pandas) для очистки сырых данных из CSV-файла. Какие типичные проблемы данных вы ожидаете и как будете их решать?

3. SQL для анализа: Вам дана схема базы данных интернет-магазина (таблицы orders, users, products). Напишите SQL-запрос, который покажет ежемесячную выручку и количество уникальных покупателей за последний год.
4. Проектирование БД: Что такое нормализация базы данных? Опишите процесс приведения таблицы к Третьей нормальной форме (3NF) на практическом примере.
5. Оптимизация: Ваш SQL-запрос начал выполняться слишком медленно. Опишите пошаговый план ваших действий для диагностики и устранения проблемы.
6. Распределенные системы: В чем заключается основная идея парадигмы MapReduce? Объясните на понятном примере, как с ее помощью можно посчитать частоту встречаемости слов в большом наборе текстов.
7. Data Lake vs Data Warehouse: Сравните концепции Data Lake и Data Warehouse по ключевым параметрам: данные, схема, цель использования. Почему современная архитектура часто включает оба этих слоя?
8. Введение в Spark: Каковы основные компоненты архитектуры Apache Spark (Driver, Executor)? В чем ключевое преимущество Spark перед однопоточной обработкой в Pandas для больших объемов данных?
9. Основы оркестрации: Зачем нужны оркестраторы пайплайнов, такие как Apache Airflow, если можно использовать cron? Перечислите как минимум три преимущества Airflow.
10. Промежуточный проект: Опишите шаги, которые вы предпримете для построения простого, но надежного ETL-пайплайна, который ежедневно забирает данные из API, преобразует их и загружает в базу данных. Какие компоненты будут ключевыми?

Примерные вопросы для подготовки к экзамену

1. Оркестрация: Опишите жизненный цикл DAG в Apache Airflow. Что такое Operator и Sensor? Приведите примеры их использования в реальном пайплайне.
2. Поточковая обработка: Опишите архитектурный паттерн Карра. В чем его основное отличие от паттерна Lambda и в каких сценариях его использование предпочтительнее?
3. Качество данных: Что такое Data Quality? Перечислите и охарактеризуйте три ключевые метрики качества данных. На каком этапе пайплайна вы бы встроили их проверку и почему?

4. Безопасность в облаке: Опишите принцип наименьших привилегий (Principle of Least Privilege). Как бы вы применили его при настройке сервисного аккаунта для кластера Yandex Data Proc, который читает данные из Object Storage и пишет в ClickHouse?
5. Инфраструктура как код (IaC): В чем заключаются основные преимущества использования Terraform для управления облачной инфраструктурой данных по сравнению с ручным созданием ресурсов через веб-консоль?
6. Поточковые технологии: Опишите роль Apache Kafka в data-пайплайне. Что такое Topic, Partition и Consumer Group и как они обеспечивают масштабируемость и отказоустойчивость?
7. Техническое проектирование: Каковы основные разделы технического проекта (ТП) на разработку системы обработки данных? Чем его содержание принципиально отличается от технического задания (ТЗ)?
8. Управление проектами: Какие ключевые риски (технические и организационные) могут возникнуть при миграции устаревшего ETL-процесса на новую облачную платформу? Предложите методы mitigation для двух из них.
9. Современные тренды: Опишите принципы архитектуры Data Mesh. Как, по вашему мнению, эта парадигма меняет роль и ответственность инженера данных в большой организации?
10. Интеграционный вопрос: Вам поручили спроектировать систему для обнаружения аномалий в финансовых транзакциях в режиме, близком к реальному времени. Опишите высокоуровневую архитектуру такого решения, указав, какие технологии из изученных вы бы использовали и почему.

Пример экзаменационного билета

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №

1. В чем заключаются основные преимущества использования Terraform для управления облачной инфраструктурой данных по сравнению с ручным созданием ресурсов через веб-консоль? **(20 баллов)**
2. Сравните концепции Data Lake и Data Warehouse по ключевым параметрам: данные, схема, цель использования. Почему современная архитектура часто включает оба этих слоя? **(20 баллов)**
3. Разработайте стратегию обеспечения качества данных в распределенной системе **(20 баллов)**

Примерные темы курсовой работы /курсового проекта

1. Построение полноценного ETL-пайплайна с использованием стека Yandex Cloud.
2. Разработка системы потоковой обработки данных интернет-магазина.
3. Создание инфраструктуры данных для сервиса аналитики в реальном времени.
4. Создание отказоустойчивой платформы сбора и обработки телеметрии с IoT-устройств.
5. Разработка системы мониторинга качества данных (Data Quality) для кредитного конвейера банка.
6. Проектирование системы кэширования и предварительных агрегатов для ускорения OLAP-запросов.
7. Проектирование и внедрение платформы Feature Store для ML-моделей в продуктовой IT-компании
8. Автоматизация сбора, очистки и анализа данных из социальных сетей для исследования рынка
9. Создание и оптимизация Data Lake для хранения и обработки логов веб-приложения на базе Yandex Object Storage и Apache Spark
10. Создание аналитической витрины данных интернет-магазина с использованием dbt и облачного DWH (Yandex ClickHouse)
11. Разработка и автоматизация ETL-пайплайна для загрузки данных из публичных API в облачное хранилище (Yandex Cloud)
12. Автоматизация сбора, очистки и анализа данных из социальных сетей для исследования рынка.
13. Проектирование и реализация системы проверки качества данных (Data Quality) в ETL-процессе с использованием Great Expectations

14. Проектирование системы кэширования и предварительных агрегатов для ускорения OLAP-запросов

15. Построение системы алертинга аномалий в финансовых транзакциях на основе потоковой аналитики

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Нормативные правовые акты:

РЕКОМЕНДАЦИИ УЧАСТНИКАМ ФИНАНСОВОГО РЫНКА ПО ПОСТРОЕНИЮ ЭФФЕКТИВНОЙ СИСТЕМЫ УПРАВЛЕНИЯ ДАННЫМИ https://cbr.ru/content/document/file/170697/recommendations_27122024.pdf

Основная литература:

1. Сидоров, И. Г. Методы и технологии обработки больших данных [Электронный ресурс] : учебно-методическое пособие для студентов направлений подготовки 09.03.03 «прикладная информатика» и 01.03.02 «прикладная математика и информатика» / Сидоров И. Г. Москва : Московский Политех, 2024 282 с. Книга из коллекции Московский Политех - Информатика <https://e.lanbook.com/book/482792> ISBN 978-5-2760-2843-9. [БИК ID: RU-LAN-BOOK-482792]
2. Алетдинова, А. А. Интеллектуальный анализ больших данных [Электронный ресурс] : учебное пособие / Алетдинова А. А., Муртазина М. Ш. Новосибирск : НГТУ, 2023 66 с. Утверждено Редакционно-издательским советом университета в качестве учебного пособия Книга из коллекции НГТУ - Информатика <https://e.lanbook.com/book/404567> ISBN 978-5-7782-4899-1. [БИК ID: RU-LAN-BOOK-404567]
3. Пальмов, С. В. Основы сбора и обработки больших данных [Электронный ресурс] : учебное пособие / Пальмов С. В. Самара : ПГУТИ, 2023 285 с. Книга из коллекции ПГУТИ - Информатика <https://e.lanbook.com/book/411830>. [БИК ID: RU-LAN-BOOK-411830]
4. Горожанина, Е. И. Высокопроизводительные вычисления и анализ больших данных [Электронный ресурс] : учебное пособие / Горожанина Е. И. Самара : ПГУТИ, 2022 132 с. Книга из коллекции ПГУТИ - Информатика <https://e.lanbook.com/book/411386>. [БИК ID: RU-LAN-BOOK-411386]

Дополнительная литература:

1. Анализ данных / Новикова О. А., Андрианова Е. Г. Ч. 1 : Анализ данных. Часть 1 : Учебное пособие . Ч. 1 / Новикова О. А., Андрианова Е. Г. Москва : РТУ МИРЭА, 2020 162 с. Книга из коллекции РТУ МИРЭА - Экономика и менеджмент <https://e.lanbook.com/book/167597>. [БИК ID: RU-LAN-BOOK-167597]

2. Осин, А. В. Технологии обеспечения информационной безопасности больших данных в компьютерных сетях. Информационная безопасность в системах обработки данных на примере Hadoop и Spark [Электронный ресурс] : учебно-методическое пособие по выполнению лабораторных работ для магистров, направление подготовки 10.04.01 «информационная безопасность», профиль «интеллектуальные технологии безопасности компьютерных систем» / Осин А. В., Хализев К. А. Москва : МТУСИ, 2025 18 с. Книга из коллекции МТУСИ - Информатика <https://e.lanbook.com/book/501212>. [БИК ID: RU-LAN-BOOK-501212]
3. Сыромятников, В. П. Структуры и алгоритмы обработки данных: Практикум [Электронный ресурс] / Сыромятников В. П. Москва : РТУ МИРЭА, 2020 244 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/163915>. [БИК ID: RU-LAN-BOOK-163915]

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ФСТЭК России <https://fstec.ru/>
2. Специализированный сайт по тематике информационной безопасности <http://all-ib.ru>
3. Five Big Data Technologies <https://www.jigsawacademy.com/big-data-5-new-technologies-emerge-2017/>
4. Свод знаний по управлению проектами PMBoK(англ. ProjectManagementBodyofKnowledge) – 6-е издание, сентябрь 2017г.
5. <http://pandas.pydata.org/>
6. Дистрибутив anaconda: <http://anaconda.org/>
7. Python Documentation <http://python.org/doc/>

10. Методические указания для обучающихся по освоению дисциплины

10.1. Организация учебного процесса

Дисциплина изучается в течение двух семестров и состоит из аудиторных занятий (лекций и семинаров) и самостоятельной работы студентов. Эффективное усвоение материала предполагает регулярную подготовку к каждому занятию. Студентам рекомендуется еженедельно уделять не менее 2 часов внеаудиторному изучению дисциплины дополнительно к очным занятиям.

Студент должен изучить лекции заранее перед семинарскими занятиями, чтобы иметь представление о теоретическом материале и активно включаться в обсуждение на семинаре.

Участвовать в дискуссиях и выполнять поставленные преподавателем задания.

10.2. Изучение теоретического материала

Во время лекционных занятий студент должен записать не только точные определения, но и важные идеи, различия подходов и технологии, а также иллюстрировать понимание материала схемами и диаграммами. Это способствует лучшему усвоению материала. Студент обязан самостоятельно проработать темы, предусмотренные программой, используя рекомендуемые учебные пособия, стандарты и официальные справочные руководства (документацию Apache Spark, Airflow, Kafka и др.). Важно не только знакомиться с материалами, но и анализировать применение теории на практике. Дополнительно рекомендуются специализированные публикации и блоги (Habr, Medium), а также веб-ресурсы компаний-поставщиков сервисов обработки больших данных (например, Yandex Cloud).

10.3. Практические занятия

Практическая работа составляет основу освоения дисциплины. Каждый студент обязан подготовить рабочую среду (локально установить Docker, виртуальное окружение Python и прочие инструменты), согласно инструкции, приведённой в начале курса. Рекомендуемая последовательность действий при выполнении практической части:

- Внимательное прочтение технического задания.
- Декомпозиция задачи на отдельные шаги (получение данных, обработка, преобразование, хранение и верификация).

- Выбор подходящего инструмента для каждого этапа на основании ранее изученного инструментария.
- Поэтапная реализация решения с проверкой работоспособности промежуточных этапов.
- Документирование всех выполненных шагов, включая обнаруженные трудности и пути их преодоления.
- Итоговая сдача должна включать сам рабочий код, пояснительный отчет с описанием выбранной архитектуры и указанием технических характеристик реализации (скорость обработки, масштабируемость и т.п.).

10.4. Курсовой проект

Курсовой проект представляет собой финальное задание, целью которого является демонстрация комплексного понимания предмета и применение приобретённых знаний и навыков.

Этапы курсового проекта:

- Выбор темы: Темы выбирают из предлагаемого списка и согласовывают с преподавателем. Целесообразно выбирать тему, соответствующую личным интересам и текущему уровню компетенций студента.
- Планирование: Необходимо составить подробный график выполнения проекта и разбить весь процесс на последовательные этапы (анализ требований, проектирование, разработку, тестирование, подготовку отчётности).
- Выполнение: Результатом должно стать создание функционирующей системы или готового прототипа, размещённого в облачном или локальном окружении.
- Защита: Студент должен представить проект на публичной защите, продемонстрировав его работоспособность, обосновывая принятые архитектурные решения и готовностью отвечать на вопросы комиссии.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Python 3.8
2. Windows
3. Docker
4. Kubernetes
5. Реляционная СУБД
6. Kaspersky

Современные профессиональные базы данных и информационные справочные системы:

1. Информационно-правовая система «Гарант»
2. Информационно-правовая система «Консультант Плюс»
3. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
4. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. **Учебная аудитория** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)
2. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)
3. Библиотека, читальный зал