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
Утвержено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: JMyeE9AyttdXHNHBLPjFDr+uOi/mlYos
Дата: 08.10.2025 г.

Н.В. Корнеев

Разработка и интеграция

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:
01.04.02 - Прикладная математика и информатика,
Образовательная программа «Инженер машинного обучения»
Направленность программы:
«Инженер машинного обучения»

Рекомендовано
Факультет информационных технологий и анализа больших данных
(протокол № 2 от 18.11.2025 г.)
Одобрено
Кафедра информационных технологий
(протокол № 6 от 22.10.2025 г.)

© Москва 2025

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	4
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	4
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	5
5.1.	Содержание дисциплины	5
5.2.	Учебно-тематический план	7
5.3.	Содержание семинаров	8
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	10
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	10
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	12
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	14
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	21
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	21
10.	Методические указания для обучающихся по освоению дисциплины	24
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	24
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	26

1. Наименование дисциплины

«Разработка и интеграция».

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
ПК-6	Способность настраивать и управлять инфраструктурой развертывания прикладных программных решений	Демонстрировать способность настраивать и поддерживать серверную и сетевую инфраструктуру для развертывания интеллектуальных систем	**знать:** действующие и перспективные технологии и инструменты настройки и поддержки серверной и сетевой инфраструктуры **уметь:** настраивать и поддерживать серверную и сетевую инфраструктуру для развертывания интеллектуальных систем с использованием современных инструментов
		Использовать в работе основные принципы администрирования сложных систем, инструментальные средства управления операциями разработки ПО	**знать:** действующие и перспективные технологии и инструментальные средства управления операциями разработки ПО **уметь:** администрировать сложные системы с использованием инструментальных средств управления операциями разработки ПО

3. Место дисциплины в структуре образовательной программы

Дисциплина «Разработка и интеграция» относится к «Модулю направленности программы магистратуры».

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)**	**Модуль 5 (в часах)**	**Модуль 6 (в часах)**
Общая трудоёмкость дисциплины	5/180	72	108
Контактная работа-Аудиторные занятия	54	24	30
Лекции	18	8	10
Семинары, практические занятия	36	16	20
Самостоятельная работа	126	48	78
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. Этапы разработки ПО

Прототипирование. Классическая водопадная модель. Спиральная модель проектирования. Инкрементная модель. Жизненный цикл разработки ПО. Идеальная разработка ПО. Быстрая разработка приложений. Гибкие методологии(Agile). Экстремальное программирование. Scrum методология. Инженерия требований. Управление рисками, задачами и дефектами. Внедрение DevOps в существующий проект.

Тема 2. Непрерывная интеграция, поставка (CI/CD)

Непрерывная интеграция. Непрерывная поставка. Непрерывное развертывание. Непрерывная поставка: этапы, коммерческая ценность. GitLab , GitGub и непрерывная поставка на примере фреймворков Django / Flask . Переход к непрерывной интеграции. Инструменты непрерывной интеграции. Магистральная разработка. Тестирование ПО. Тестирование ПО при непрерывной поставке.

Тема 3. Архитектурные паттерны

Паттерны проектирования, как подход к повторному использованию удачных решений. Принципы применения паттернов. Классификация паттернов. Каталог паттернов. Разработка веб-сервисов на примере фреймворков Django / Flask .

Тема 4. Микросервисная и монолитная архитектура

Архитектура FTGO . Преимущество монолитной архитектуры. Микросервисная архитектура и ее преимущества и недостатки. Стратегии декомпозиции. Межпроцессное взаимодействие в микросервисной архитектуре. Бизнес-архитектура и проектирование бизнес-логики в микросервисной архитектуре. Тестирование микросервисов. Развертывание микросервисов. Разработка веб-сервисов с применением Deep learning & Machine learning на примере фреймворков Django / Flask .

Тема 5. Системы мониторинга

Мониторинг, как средство обеспечения качества функционирующей системы. Типы мониторинга, основные подходы и инструменты реализации. Zabbix .

Тема 6. Обзор систем управления конфигурациями

Системы управления конфигурациями, основные определение и обзор существующих. Ansible основы. Playbook . Сетевые модули привязанные к конкретной ОС. Модули ресурсов. Получение структурированного вывода.

Тема 7. Оркестрация

Системы оркестрации контейнеров и зачем они нужны. Преимущества и недостатки. Задачи оркестраторов контейнеров. Kubernetes . Архитектура Kubernetes . Создание кластеров. Мониторинг и журналирование. Безопасность в Kubernetes . Работа с хранилищем данных.

Тема 8. Концепция инфраструктура как код

Описание инфраструктуры в виде кода. Основы Terraform . Начало работы с Terraform . Базовый синтаксис.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоятельная работа
			Общая, в т.ч.:	Лекции	Семинары, практические занятия	
1	Этапы разработки ПО	16	6	2	4	10
2	Непрерывная интеграция, поставка (CI/CD)	36	16	4	12	20
3	Архитектурные паттерны	14	4	2	2	10
4	Микросервисная и монолитная архитектура	14	4	2	2	10
5	Системы мониторинга	28	8	2	6	20
6	Обзор систем управления конфигурациями	28	8	2	6	20
7	Оркестрация	24	4	2	2	20
8	Концепция инфраструктура как код	20	4	2	2	16
	Итого	180	54	18	36	126

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	**Перечень вопросов для обсуждения на семинарах, практических занятиях**	**Формы проведения занятий**
Этапы разработки ПО	Прототипирование. Классическая водопадная модель. Спиральная модель проектирования. Инкрементная модель. Жизненный цикл разработки ПО. Идеальная разработка ПО. Быстрая разработка приложений. Гибкие методологии(Agile). Экстремальное программирование. Scrum методология. Инженерия требований.	Интерактивная форма, работа на компьютере.
Непрерывная интеграция, поставка (CI/CD)	Непрерывная интеграция. Непрерывная поставка. Непрерывное развертывание. Непрерывная поставка: этапы, коммерческая ценность. GitLab и непрерывная поставка. Переход к непрерывной интеграции. Инструменты непрерывной интеграции. Магистральная разработка. Django.	Интерактивная форма, работа на компьютере.
Архитектурные паттерны	Паттерны проектирования, как подход к повторному использованию удачных решений. Принципы применения паттернов. Классификация паттернов. Каталог паттернов. Django и Flask.	Интерактивная форма, работа на компьютере.
Микросервисная и монолитная архитектура	Микросерсвисная архитектура и ее преимущества и недостатки. Стратегии декомпозиции. Межпроцессное взаимодействие в микросервисной архитектуре. Бизнес-архитектура и проектирование бизнес-логики а микросервисной архитектуре. Тестирование микросервисов. Развертывание микросервисов. Deep learning & Machine learning. Django/Flask.	Интерактивная форма, работа на компьютере.
Системы мониторинга	Мониторинг, как средство обеспечения качества функционирующей системы. Типы мониторинга, основные подходы и инструменты реализации. Zabbix.	Интерактивная форма, работа на компьютере.
Обзор систем управления конфигурациями	Системы управления конфигурациями, основные определение и обзор существующих. Ansible основы. Playbook. Сетевые модули, привязанные к конкретной ОС.	Интерактивная форма, работа на компьютере.
Оркестрация	Системы оркестрации контейнеров и зачем они нужны. Преимущества и недостатки. Задачи оркестраторов контейнеров. Kubernetes. Архитектура Kubernetes. Создание кластеров. Мониторинг и журналирование. Безопасность в Kubernetes.	Интерактивная форма, работа на компьютере.
Концепция	Написание инфраструктуры в виде кода. Основы	Интерактивная форма,

инфраструктура как код	Terraform. Начало работы с Terraform. Базовый синтаксис.	работа на компьютере.

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**
Этапы разработки ПО	Экстремальное программирование. Scrum методология. Инженерия требований. Управление рисками, задачами и дефектами. Внедрение DevOps в существующий проект.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия.
Непрерывная интеграция, поставка (CI/CD)	Тестирование ПО. Тестирование ПО при непрерывной поставке.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия.
Архитектурные паттерны	Классификация паттернов. Каталог паттернов.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия.
Микросервисная и монолитная архитектура	Архитектура FTGO. Преимущество монолитной архитектуры.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия. Прохождение бесплатного онлайн-курса по тематике дисциплины (р.9.11).
Системы мониторинга	Zabbix. Основы работы с Zabbix.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия.
Обзор систем управления конфигурациями	Playbook. Сетевые модули привязанные к конкретной ОС. Модули ресурсов. Получение структурированного вывода.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия.
Оркестрация	Безопасность в Kubernetes. Работа с хранилищем данных.	Работа с учебной литературой. Решение типовых задач. Разбор

		вопросов по теме занятия.
Концепция инфраструктура как код	Основы Terraform. Начало работы с Terraform. Базовый синтаксис.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия. Прохождение бесплатного онлайн-курса по тематике дисциплины (р.9.27).

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

Описание проектной работы

В рамках проектной работы студенту необходимо разработать паттерн на базе микросервисной архитектуры по индивидуальному заданию. Этапы проектной работы включают в себя: анализ предметной области; разработку технического задания; проектирование архитектуры паттерна с микросервисами (не более 5 микросервисов) с реализацией в контейнерах и выбор инструментов разработки и развертывания; разработку кодов микросервисов, в IDE или облачной платформе типа SaaS , например Replit ; описание этапов контейнеризации и оркестрации контейнеров, а также мониторинг и тестирование микросервисов на кластере, на базе Kubernetes с помощью minikube и Docker или на платформе Yandex Cloud, с подключением например Prometheus, Grafana, cAdvisor и node-exporter; подготовку отчета по проектной работе в текстовом редакторе включающий все указанные выше этапы; подготовку презентации по результатам проектной работы в объёме не менее 15 слайдов. Индивидуальное задание для проектной работы выбирается студентом из предложенного списка, допускается инициатива студента по авторскому выбору задания для проектной работы, окончательная задача утверждается преподавателем.

Примерная тематика проектной работы

1. Разработка паттерна для обеспечения безопасности приложения при угрозе изменения компонентов информационной системы.

2. Разработка паттерна с ML-сервисом на основе Docker-образа в DataSphere с реализацией объектного распознавания на GPU.

3. Разработка паттерна для обеспечения безопасности приложения при угрозе несанкционированного доступа к активному виртуальному сетевому оборудованию из физической сети.

4. Разработка паттерна для развертывания RAG с временными фильтрами и хранилищем векторных изображений с искусственным интеллектом

5. Разработка паттерна для обеспечения безопасности приложения при угрозе несанкционированного доступа к защищаемым виртуальным устройствам из физической сети.

6. Разработка паттерна для модели архитектуры глубокого обучения на тензорном вычислительном модуле (TCU).

7. Разработка паттерна для обеспечения безопасности приложения при угрозе несанкционированного доступа к защищаемым виртуальным машинам из виртуальной сети.

8. Разработка паттерна для обеспечения безопасности приложения при угрозе подмены модели машинного обучения.

9. Разработка паттерна для обеспечения безопасности приложения при угрозе модификации модели машинного обучения.

10. Разработка паттерна для обеспечения безопасности приложения при угрозе нарушения функционирования технологий искусственного интеллекта.

11. Разработка паттерна для обеспечения безопасности приложения при угрозе хищения обучающих данных.

12. Разработка паттерна для обеспечения безопасности приложения при угрозе использования скомпрометированного доверенного источника обновлений.

13. Разработка паттерна для обеспечения безопасности системы при угрозе несанкционированного доступа при помощи сторонних сервисов.

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

ПК-6 Способность настраивать и управлять инфраструктурой развертывания прикладных программных решений

1) Демонстрировать способность настраивать и поддерживать серверную и сетевую инфраструктуру для развертывания интеллектуальных систем

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

Знать: действующие и перспективные технологии и инструменты настройки и поддержки серверной и сетевой инфраструктуры

Уметь: настраивать и поддерживать серверную и сетевую инфраструктуру для развертывания интеллектуальных систем с использованием современных инструментов

Типовые контрольные задания

1. Опишите, как вы поняли - в чем основное отличие паравиртуализации и виртуализации на основе ОС.

2. Выберите один тип из вариантов использования организации физических серверов, в зависимости от условий использования.

3. Возможно ли совмещать несколько типов виртуализации на одном сервере? Приведите пример такого совмещения.

4. Написать bash скрипт, который будет создавать PostgreSQL бэкап. Для автоматического запуска скрипта создаём запись в Cron.

5. Написать скрипт, который будет выполнять проверку состояния диска и, если места меньше чем 85%, то высылать алерт на почту. Для отправки писем напрямую из консоли можно использовать ssmtp клиент.

6. Разработать и настроить инфраструктуру в облаке с помощью Terraform для приложения, которое будет тестироваться на устойчивость и надежность.

7. Описать конфигурацию мониторинг сервера в Ansible. Конфигурация должна включать настройку Prometheus, Grafana, Alertmanager на сервере мониторинга и конфигурацию с открытием портов в firewall правилах на серверах, которые мы хотим мониторить. Для мониторинга инстансов, ставим node - exporter , который выводит метрики на порт 9100 (его мы и открываем на отслеживаемых серверах). В целях безопасности порт 9100 открываем только для мониторинг инстанса.

2) Использовать в работе основные принципы администрирования сложных систем, инструментальные средства управления операциями разработки ПО

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

Знать: действующие и перспективные технологии и инструментальные средства управления операциями разработки ПО

Уметь: администрировать сложные системы с использованием инструментальных средств управления операциями разработки ПО

Типовые контрольные задания

1. Проведите анализ инструментов и средств разработки паттерна для модификации модели машинного обучения.

2. Разработайте техническое задание на паттерн для модификации модели машинного обучения.

3. Спроектируете архитектуру паттерна для модификации модели машинного обучения с микросервисами и реализацией в контейнерах.

4. Осуществите выбор инструментов разработки и развертывания паттерна для модификации модели машинного обучения.

5. Опишите этапы контейнеризации и оркестрации паттерна для модификации модели машинного обучения.

6. Опишите этапы мониторинга и тестирование микросервисов на кластере Kubernetes с подключением например Prometheus , Grafana , cAdvisor и node - exporter.

7. Опишите этапы мониторинга и тестирования микросервисов в Yandex Cloud.

Примеры практико-ориентированных заданий

1. Выберите тип один из вариантов использования организации физических серверов, в зависимости от условий использования.

Организация серверов:

• физические сервера

• паравиртуализация

• виртуализация уровня ОС

Условия использования:

• Высоконагруженная база данных, чувствительная к отказу

• Различные Java-приложения

• Windows системы для использования Бухгалтерским отделом

• Системы, выполняющие высокопроизводительные расчеты на GPU

Опишите, почему вы выбрали к каждому целевому использованию такую организацию.

2. Составьте таблицу возможностей различных брокеров сообщений. На основе таблицы сделайте обоснованный выбор решения.

Решение должно соответствовать следующим требованиям:

- Поддержка кластеризации для обеспечения надежности

- Хранение сообщений на диске в процессе доставки

- Высокая скорость работы

- Поддержка различных форматов сообщений

- Разделение прав доступа к различным потокам сообщений

- Простота эксплуатации

Обоснуйте свой выбор.

3. Для микросервиса осуществляющего модификацию модели машинного обучения разработать приложение и разместить его на GitHub.

4. Импортировать проект приложения из GitHub в Replit. Настроить файлы конфигурации и развернуть приложение на Replit. Протестировать работу приложения на синтетических данных.

5. Создать скрипт для развертывания шаблона приложения на Replit. Поделиться ссылкой на скрипт с преподавателем по корпоративной электронной почте.

Примерные вопросы для подготовки к зачету, экзамену

Примерные вопросы для подготовки к зачету

1. Прототипирование.

2. Классическая водопадная модель.

3. Спиральная модель проектирования.

4. Инкрементная модель.

5. Жизненный цикл разработки ПО.

6. Идеальная разработка ПО.

7. Быстрая разработка приложений.

8. Гибкие методологии(Agile).

9. Экстремальное программирование.

10. Scrum методология.

11. Инженерия требований.

12. Управление рисками, задачами и дефектами.

13. Внедрение DevOps в существующий проект

14. Чем отличается GitLab от Git?

15. Что делают операции add, commit, push в Git?

16. Для чего нужен GitLab Runner?

17. Как называется файл для CI/CD в GitLab?

18. Что такое “ветка” в Git и почему переименовали master в main?

19. Зачем нужен Docker?

20. Чем Docker отличается от LXC?

21. Из каких частей состоит имя образа?

22. Чем Docker контейнер отличается от Docker образа?

23. Как ПО в Docker контейнере может взаимодействовать с системой и другими контейнерами?

24. Зачем нужен Dockerfile?

25. Зачем нужен docker-compose?

26. Непрерывная интеграция.

27. Непрерывная поставка.

28. Непрерывное развертывание.

29. Непрерывная поставка: этапы, коммерческая ценность.

30. GitLab и непрерывная поставка.

31. Переход к непрерывной интеграции.

32. Инструменты непрерывной интеграции.

33. Магистральная разработка.

34. Тестирование ПО.

35. Тестирование ПО при непрерывной поставке.

Примерные вопросы для подготовки к экзамену

1. Паттерны проектирования, как подход к повторному использованию удачных решений.

2. Принципы применения паттернов.

3. Классификация паттернов.

4. Каталог паттернов.

5. Архитектура FTGO.

6. Преимущество монолитной архитектуры.

7. Микросервисная архитектура и ее преимущества и недостатки.

8. Стратегии декомпозиции.

9. Межпроцессное взаимодействие в микросервисной архитектуре.

10. Бизнес-архитектура и проектирование бизнес-логики и микросервисной архитектуре.

11. Тестирование микросервисов.

12. Развертывание микросервисов.

13. Мониторинг, как средство обеспечения качества функционирующей системы.

14. Типы мониторинга, основные подходы и инструменты реализации.

15. Системы управления конфигурациями, основные определение и обзор существующих.

16. Ansible основы.

17. Playbook.

18. Сетевые модули, привязанные к конкретной ОС.

19. Модули ресурсов. Получение структурированного вывода.

20. Системы оркестрации контейнеров и зачем они нужны. Преимущества и недостатки.

21. Задачи оркестраторов контейнеров.

22. Kubernetes. Архитектура Kubernetes.

23. Мониторинг и журналирование.

24. Безопасность в Kubernetes.

25. Что общего и в чем отличие между Ansible, Vagrant и Terraform?

26. Что означает термин Provision в Vagrant?

27. Что такое module, playbook, role в Ansible?

28. Назовите каталоги в роли Ansible

29. Что должно быть установлено на целевой системе для управления через Ansible?

30. Каким образом Vagrant управляет VM?

31. Как реализован Ansible provision в Vagrant?

32. Почему Kubernetes часто обозначают K8s?

33. Как связаны понятия Deployment, Replica Sets, Pods?

34. Для чего используются Namespace?

35. Что общего и в чем различие между Service и Ingress?

36. Что используется для подключения Pod к общим файлам?

37. Для чего нужен Helm и что используется для его описания?

Пример экзаменационного билета

Вопрос 1. Тестирование микросервисов **(15 баллов)**

Вопрос 2. Как связаны понятия Deployment , Replica Sets , Pods ? **(15 баллов)**

Вопрос 3. Создать веб-приложение на Spring Boot, которое будет обращаться к базе данных (выбрать предметную область). Приложение должно иметь слои: model, dao, service, controller. Написать Dockerfile **(30 баллов)**

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

Основная литература:

1. Баланов, А. Н. DevOps: интеграция и автоматизация [Электронный ресурс] : учебное пособие для вузов / Баланов А. Н. 2-е изд., стер. Санкт-Петербург : Лань, 2025 240 с. Книга из коллекции Лань - Информатика https://e.lanbook.com/book/440162 ISBN 978-5-507-50491-6. [БИК ID: RU-LAN-BOOK-440162]

2. Васильева, М. А. Система контроля версий. Основы командной разработки [Электронный ресурс] : учебное пособие для вузов / Васильева М. А.,Филипченко К. М. ; Филипченко К. М. Санкт-Петербург : Лань, 2022 144 с. Книга из коллекции Лань - Информатика https://e.lanbook.com/book/261089 ISBN 978-5-507-44630-8. [БИК ID: RU-LAN-BOOK-261089]

Дополнительная литература:

1. Горбенко, А.О. Программная инженерия : Учебник / А.О. Горбенко, А.А. Попов Электрон. дан. Москва : КноРус, 2025 307 с. Режим доступа: book.ru Internet access https://book.ru/book/958136 ISBN 978-5-406-14751-1. [БИК ID: RU\bookru\bibl\ 958136]

2. Жматов, Д. В. Технологии разработки программных приложений, основанные на использовании Git, Docker, Gradle: Практикум [Электронный ресурс] / Жматов Д. В.,Петрова А. А. Москва : РТУ МИРЭА, 2025 70 с. Книга из коллекции РТУ МИРЭА - Информатика https://e.lanbook.com/book/504851 ISBN 978-5-7339-2592-9. [БИК ID: RU-LAN-BOOK-504851]

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

1. Информационно-образовательный портал Финуниверситета: https://org.fa.ru

2. Электронная библиотека Финансового университета (ЭБ) http://elib.fa.ru/ (http://library.fa.ru/files/elibfa.pdf)

3. Научная электронная библиотека eLibrary.ru http://elibrary.ru

4. Электронно-библиотечная система BOOK.RU http://www.book.ru

5. Электронно-библиотечная система "Университетская библиотека ОНЛАЙН" http://biblioclub.ru/

6. Электронно-библиотечная система издательства "Лань" https://e.lanbook.com/

7. Образовательная платформа "ЮРАЙТ" https://urait.ru/

8. "Деловая онлайн библиотека" издательства "Альпина Паблишер" http://lib.alpinadigital.ru/en/library

9. •Электронно-библиотечная система Znanium http://www.znanium.ru/

10. Математические журналы: полнотекстовая коллекция Математического института им. В.А. Стеклова РАН https://www.mathnet.ru/

11. •Национальная электронная библиотека http://нэб.рф/

12. Ресурсы информационно-аналитического агентства по финансовым рынкам Cbonds.ru https://cbonds.ru/

13. СПАРК https://spark-interfax.ru/

14. •CNKI. Academic Reference https://ar.oversea.cnki.net/

15. •Электронные продукты издательства Elsevier http://www.sciencedirect.com

16. Emerald: Management eJournal Portfolio https://www.emerald.com/insight/

17. Реферативная база данных по математике MathSciNET https://mathscinet.ams.org/mathscinet/

18. •Коллекция научных журналов Oxford University Press https://academic.oup.com/journals/

19. •Электронные коллекции книг и журналов издательства Springer: http://link.springer.com/

20. Платформа STATISTA https://www.statista.com/

21. •База данных научных журналов издательства Wiley https://onlinelibrary.wiley.com/

22. Основы Terraform [Электронный ресурс] URL: https://habr.com/ru/company/otus/blog/696694/

23. Для чего вам нужен Terraform? [Электронный ресурс] URL: https://habr.com/ru/company/nixys/blog/692138/

24. Что такое модули Terraform и как они работают? [Электронный ресурс] URL: https://habr.com/ru/post/553576/

25. Terraform official documentation [Электронный ресурс] URL: https://www.terraform.io/

26. Бесплатный онлайн курс «Инженер облачных сервисов» https://practicum.yandex.ru/ycloud/

27. Principles of Machine Learning / Microsoft. – https://www.edx.org/course/principles-machine-learning-microsoft-dat203-2x-3

28. Applied Machine Learning / Microsoft. – https://www.edx.org/course/applied-machine-learning-microsoft-dat203-3x-1

29. Сервис визуализации и анализа данных от Yandex Cloud. URL: https://cloud.yandex.ru/docs/datalens/

10. Методические указания для обучающихся по освоению дисциплины

Организация самостоятельной работы основана на учебно-тематическом плане изучения дисциплины, где указана тематика лекций, практических занятий, вопросы и задания для самостоятельного изучения. При подготовке к практическому занятию необходимо повторить или, если это требуется, изучить соответствующий теоретический материал. Практические занятия проходят, как правило, в интерактивной форме и преподаватель учитывает активность студентов в процессе решения предложенных задач и поиска ответов на вопросы. Домашние задания следует выполнять регулярно при подготовке к практическим занятиям. В большинстве своем задания являются типовыми, и образцы их решения содержатся в рекомендованных пособиях, в материале лекций и практических занятий. Контроль выполнения домашних заданий осуществляется в ходе практических занятий в процессе выборочного собеседования.

Проектная работа является одной из основных форм текущего контроля самостоятельной работы студентов по дисциплине Каждый вариант проектной работы содержит комплексное исследование, выполняя которое студент демонстрирует умение реализовывать изученные методы и инструменты на компьютере. Оценка за проектную работу выставляется по итогам проверки отчета и презентации работы. Эта оценка является существенной компонентой оценки самостоятельной работы студента в течение семестра.

Дисциплина адаптирована для студентов с ограниченными возможностями здоровья.

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. Figma
2. Kaspersky
3. Microsoft Office (Windows)
4. Операционная система
5. Текстовый редактор

6. Docker

7. Kubernets

8. Anaconda

9. Виртуальные машины

10. Yandex Cloud

11. Web сервер node.js версии не ниже 18 LTS

12. Комплект свободно распространяемого программного обеспечения

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. Язык программирования Python 3. https://pythonworld.ru/

5. eLIBRARY.RU

6. Яндекс 360

7. Информационная система Томсон Рейтер (Refinitiv)

8. Информационная система СПАРК

9. Компьютерная техника

10. Информационная система Bloomberg

11. База научных статей Национального бюро экономических исследований (NBER).

12. Электронная библиотека диссертаций Российской государственной библиотеки https://dvs.rsl.ru/

13. СУБД MongoDB

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

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

1. **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)