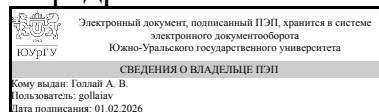


ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:
Заведующий выпускающей
кафедрой



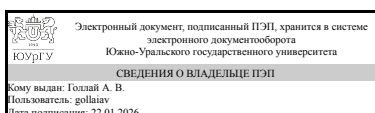
А. В. Голлай

РАБОЧАЯ ПРОГРАММА

**дисциплины 1.Ф.П0.07 Микросервисная архитектура и основы DevOps
для направления 09.03.02 Информационные системы и технологии
уровень Бакалавриат
профиль подготовки ИТ-инженерия
форма обучения очная
кафедра-разработчик Центр подготовки топ-специалистов в сфере ИТ "Цифровой
Урал"**

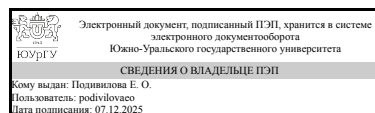
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, утверждённым приказом Минобрнауки от 19.09.2017 № 926

Зав.кафедрой разработчика,
д.техн.н., доц.



А. В. Голлай

Разработчик программы,
к.техн.н., доцент



Е. О. Подивилова

1. Цели и задачи дисциплины

Целью дисциплины является сформировать у студентов целостное представление о принципах, паттернах и инструментах разработки и эксплуатации распределенных приложений на основе микросервисной архитектуры в рамках DevOps-культуры. Задачи дисциплины: изучить эволюцию архитектур от монолита к микросервисам, их преимущества и компромиссы; освоить ключевые паттерны проектирования, взаимодействия и обеспечения устойчивости сервисов; познакомиться с практиками и инструментами DevOps для автоматизации жизненного цикла ПО (CI/CD, контейнеризация, оркестрация, мониторинг); выработать практические навыки проектирования, развертывания, наблюдения и поддержки микросервисной системы с использованием современного технологического стека (Docker, Kubernetes, системы CI/CD и др.).

Краткое содержание дисциплины

Дисциплина охватывает полный жизненный цикл создания и эксплуатации распределенных систем, начиная с философии DevOps, сравнительного анализа монолитной и микросервисной архитектур и принципов их проектирования на основе Domain-Driven Design (DDD). Студенты осваивают ключевые технологии контейнеризации (Docker) и оркестрации (Kubernetes), паттерны межсервисного взаимодействия (REST, gRPC, брокеры сообщений) и обеспечения устойчивости. Практическая часть курса фокусируется на автоматизации процессов сборки, тестирования и развертывания через настройку CI/CD-пайплайнов, а также на инструментах наблюдаемости (мониторинг, логирование, трейсинг) для построения надежных, масштабируемых и легко обслуживаемых приложений.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
ПК-1 Способен разрабатывать и отлаживать программный код	Знает: возможности используемой системы управления версиями и вспомогательных инструментальных программных средств; установленный регламент использования системы управления версиями. Умеет: регистрировать изменения исходного текста программного кода в системе управления версиями, сохранять изменения программного кода в соответствии с регламентом управления версиями, выполнять слияние, разделение и сравнение исходных текстов программного кода. Имеет практический опыт: использования систем управления версиями и вспомогательных инструментальных программных средств при разработке и отладке программного кода
ПК-3 Способен выполнять интеграцию программных модулей и компонентов и проверять работоспособность выпусков программного продукта	Знает: методы и средства сборки и интеграции программных модулей и компонентов, проверки работоспособности выпусков программных продуктов Умеет: выполнять процедуры сборки

	программных модулей и компонентов в программный продукт; производить настройки параметров программного продукта и осуществлять запуск процедур сборки; проводить проверку работоспособности программного продукта Имеет практический опыт: выполнения интеграции программных модулей и компонентов и проверки работоспособности выпусков программного продукта
ПК-12 Способен оптимизировать производительность программного обеспечения	Знает: методы и средства мониторинга производительности компьютерного программного обеспечения; метрики производительности программного обеспечения; современные инструменты мониторинга производительности программного обеспечения Умеет: применять методы и средства мониторинга производительности компьютерного программного обеспечения; интерпретировать диагностические данные мониторинга производительности компьютерного программного обеспечения; определять "узкие места" программного обеспечения Имеет практический опыт: мониторинга производительности программного обеспечения

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
Математическая логика и теория алгоритмов, Основы программирования на Python, Вычислительные методы в ИТ, Объектно-ориентированное программирование, Программирование на языках высокого уровня, Разработка мобильных приложений, Основы теории булевых функций, Алгоритмы и структуры данных, Информатика, Языки программирования низкого уровня, Основы операционных систем и администрирование Linux, Анализ требований и проектирование ПО	Интеграция и API

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Дисциплина	Требования
Информатика	Знает: методы и приемы алгоритмизации поставленных задач; нотации и программное обеспечение для графического отображения алгоритмов, принципы работы современных

	информационных технологий; принципы работы программных средств, базовые понятия информатики и информационных технологий, основные форматы представления информации для автоматизированной обработки; основные принципы работы вычислительных систем и их компонентов; ОПК-2.1. 3-3. Знает основные принципы решения задач с помощью компьютера, понятие алгоритма, основные алгоритмические структуры; состав, назначение функциональных компонентов и программного обеспечения персонального компьютера; Умеет: использовать методы и приемы формализации и алгоритмизации поставленных задач, использовать современные информационные технологий при решении задач профессиональной деятельности; использовать программные средства при решении задач профессиональной деятельности, работать с информацией, представленной в различных формах; разрабатывать алгоритмы для решения типовых задач; работать с персональным компьютером, применять современное ПО, в т.ч., отечественного производства, для решения практических задач Имеет практический опыт: навыками использования современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности, разработки алгоритмов для решения типовых задач
Вычислительные методы в ИТ	Знает: основные принципы построения и исследования численных методов и технологий, принципы организации вычислительных процессов и структуры данных, области применения вычислительных методов и реализующих их алгоритмов, знать содержательную сторону возникающих практических задач в области системного анализа и анализа данных , алгоритмы решения типичных задач, области и способы их применения; методы и приемы алгоритмизации поставленных задач Умеет: разрабатывать вычислительные схемы и оценивать точность полученных результатов и их влияние на проектируемые ИТ-решения, использовать аппарат численных методов для решения аналитических и исследовательских задач, интерпретировать полученные результаты , применять алгоритмы решения типовых задач в соответствующих областях Имеет практический опыт: применения известных методик для построения вычислительных алгоритмов, реализации вычислительных схем на современных языках программирования,

	реализации вычислительных схем на современных языках программирования
Основы программирования на Python	Знает: алгоритмы решения типичных задач, области и способы их применения; синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования, особенности синтаксиса языка Python и базовых библиотек для работы с информацией (NumPy, Pandas, Matplotlib) Умеет: применять выбранные языки программирования для написания программного кода, создавать эффективные программы на Python для решения простых и сложных задач обработки данных Имеет практический опыт: применения технологий написания читаемого и поддерживаемого кода на Python
Языки программирования низкого уровня	Знает: методы и средства оптимизации производительности компьютерного программного обеспечения; современные инструменты оптимизации производительности программного обеспечения, архитектуру современных процессоров, особенности работы с оборудованием процессора; принципы взаимодействия ПО с аппаратурой Умеет: оптимизировать программный код с использованием специализированных программных средств; вырабатывать варианты оптимизации производительности компьютерного программного обеспечения; проводить оценку и обоснование рекомендуемых решений по оптимизации производительности, разрабатывать низкоуровневый код для встроенного программного обеспечения и драйверов Имеет практический опыт:
Анализ требований и проектирование ПО	Знает: методы выявления, анализа и разработки требований при проектировании сложных программных систем; нотации моделирования бизнес-процессов в нотации BPMN для формирования требований к ПО; классификацию и структуру проектной документации на программные системы; стандарты и нотации моделирования (UML, BPMN IDEF0) для описания архитектуры и процессов; основы архитектуры программных систем и влияние архитектурных решений на качество ПО; правила оформления и ведения спецификаций, технических заданий, основные стандарты оформления технической документации на компьютерное программное обеспечение, возможности типовой ИС, предметную область автоматизации, инструменты и методы моделирования бизнес-процессов, методы и средства проектирования компьютерного программного обеспечения, методы и средства проектирования программных интерфейсов

	Умеет: создавать и описывать объектно-ориентированные модели предметной области; разрабатывать техническую и проектную документацию на основе требований и стандартов; применять унифицированные языки моделирования (UML, IDEF0) для описания архитектуры и компонентов системы; использовать шаблоны проектирования при разработке архитектуры и спецификаций модулей; моделировать бизнес-процессы в нотации BPMN и анализировать требования, выявлять противоречия и формировать корректные спецификации; оформлять документы в соответствии с установленными нормами и шаблонами организации, применять заданные стандарты и шаблоны для составления и оформления технической документации, анализировать исходную документацию в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС, разрабатывать документы в рамках выполнения работ и управления работами по созданию (модификации) и сопровождению ИС, выбирать средства реализации требований к компьютерному программному обеспечению, вырабатывать варианты реализации компьютерного программного обеспечения, проводить оценку и обоснование рекомендуемых решений Имеет практический опыт: навыками подготовки и оформления документации (техническое задание, спецификация, концепция проекта, архитектурное описание) в соответствии со стандартами; инструментами и средствами CASE-технологий для создания диаграмм и спецификаций; навыками работы с инструментальными средствами моделирования бизнес-процессов и архитектуры ПО
Математическая логика и теория алгоритмов	Знает: основные понятия и законы математической логики и теории алгоритмов, необходимые для построения моделей профессиональных задач, алгоритмы решения типичных задач, области и способы их применения; методы и приемы алгоритмизации поставленных задач Умеет: формализовать профессиональные задачи средствами математической логики и строить модели процессов с использованием методов теории алгоритмов, использовать методы и приемы формализации и алгоритмизации поставленных задач Имеет практический опыт: разработки и проверки правильности логико-математических конструкций и алгоритмов, используемых в расчетах и проектировании
Программирование на языках высокого уровня	Знает: синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные

	библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования; возможности используемой системы управления версиями и вспомогательных инструментальных программных средств; установленный регламент использования системы управления версиями, основные возможности современных интегрированных сред разработки программного обеспечения на языках высокого уровня, возможности компиляторов и компоновщиков под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программ Умеет: применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода; регистрировать изменения исходного текста программного кода в системе управления версиями; сохранять изменения программного кода в соответствии с регламентом управления версиями; выполнять слияние, разделение и сравнение исходных текстов программного кода, использовать функциональные возможности современных интегрированных сред разработки программного обеспечения на языках высокого уровня для разработки прикладных программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах. Имеет практический опыт: работы с основными современными интегрированными средами разработки программного обеспечения на языках высокого уровня, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Windows и Linux
Основы операционных систем и администрирование Linux	Знает: основные концепции и принципы построения современных операционных систем; архитектуру, функции и назначение подсистем ОС (управление процессами, памятью, вводом-выводом, файловыми системами); особенности организации многопользовательских и многозадачных систем, основные концепции современных операционных систем; особенности архитектуры и организации ОС семейства Linux, интерфейсы взаимодействия с внешней средой, архитектуру ядра Linux; принципы работы встраиваемых операционных систем, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий , основные широко распространенные

	операционные системы, принципы их работы Умеет: применять стандартные инструменты современных операционных систем (в том числе Linux и Windows) при решении задач профессиональной деятельности; использовать интерфейсы командной строки и API для взаимодействия с ОС, использовать стандартные инструменты современных ОС при решении задач профессиональной деятельности; устанавливать, настраивать и администрировать операционные системы на базе Linux, портировать код между различными вариантами загрузчиков и ОС; выполнять разработку на стыке программного обеспечения и оборудования, применять программные среды разработки информационных систем и технологий для решения прикладных задач различных классов, устанавливать и настраивать операционную систему, создавать прикладные программы в терминах API ОС Имеет практический опыт: работы с основными интерфейсами ОС (командный интерфейс и API); средствами диагностики и анализа состояния системы; приёмами управления вычислительными ресурсами ОС, навыками практической работы в среде Linux, владения технологиями, использования основных видов интерфейсов операционной системы Windows
Алгоритмы и структуры данных	Знает: синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования, фундаментальных свойств алгоритмов и структур данных; методы решения алгоритмических задачи с применением структур данных в соответствии с особенностями предметной области Умеет: применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; возможности имеющейся технической и/или программной архитектуры для написания программного кода, проектировать алгоритмическое решение на основе выбранной структуры данных; использовать средства разработки и отладки современной интегрированной среды программирования Имеет практический опыт: разработки программного кода с использованием языков программирования, определения и манипулирования данными, применяя выбранную систему контроля версий и инструментальные программные средства, применения широкого набора приемов, методов

	и технологий программирования различных задач
Разработка мобильных приложений	Знает: синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования. Умеет: применять выбранные языки программирования для написания программного кода мобильных приложений, использовать выбранную среду программирования, возможности имеющейся технической и/или программной архитектуры для написания программного кода мобильных приложений. Имеет практический опыт: разработки программного кода модулей мобильных приложений с использованием языков программирования
Основы теории булевых функций	Знает: теоретические основы и понятийный аппарат теории булевых функций; возможности и область применения аппарата теории булевых функций для понимания принципов работы современных информационных технологий и программных средств; алгоритмы решения типичных задач, области и способы их применения; методы и приемы алгоритмизации поставленных задач. Умеет: применять аппарат теории булевых функций при решении задач профессиональной деятельности; использовать методы и приемы формализации и алгоритмизации поставленных задач. Имеет практический опыт: применения аппарата теории булевых функций при решении задач профессиональной деятельности
Объектно-ориентированное программирование	Знает: методы и средства проверки работоспособности компьютерного программного обеспечения, синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; методологии разработки компьютерного программного обеспечения; технологии программирования; возможности используемой системы управления версиями и вспомогательных инструментальных программных средств; установленный регламент использования системы управления версиями, основные возможности современных интегрированных сред разработки программного обеспечения на объектно-ориентированных языках программирования, возможности компиляторов программных проектов под различные операционные системы, наборы инструкций для системных утилит автоматической сборки программного

	обеспечения и установки программных пакетов объектно-ориентированных библиотек и фреймворков, принципы чистого кода, SOLID, DRY, KISS и др., принципы предметно-ориентированного проектирования (ПОП) программного обеспечения, паттерны проектирования и антипаттерны. Умеет: применять методы и средства проверки работоспособности компьютерного программного обеспечения, применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода; регистрировать изменения исходного текста программного кода в системе управления версиями; сохранять изменения программного кода в соответствии с регламентом управления версиями; выполнять слияние, разделение и сравнение исходных текстов программного кода, использовать функциональные возможности современных интегрированных сред разработки программного обеспечения на объектно-ориентированных языках программирования для разработки прикладных программ, использовать утилиты автоматической сборки и развертывания программ в операционных системах, разрабатывать модульный и тестируемый программный код, выполнять модульное, интеграционное и нагрузочное тестирование, проводить рефакторинг для повышения качества кода, применять принципы ПОП при разработке программного обеспечения на языках программирования высокого уровня абстракций и в LowCode и NoCode системах. Имеет практический опыт: работы с основными современными интегрированными средами разработки программного обеспечения на объектно-ориентированных языках, разработки, отладки и развёртывания программного обеспечения в операционных системах семейства Windows и Linux
--	---

4. Объём и виды учебной работы

Общая трудоемкость дисциплины составляет 3 з.е., 108 ч., 72,5 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам
		в часах
		Номер семестра
		6

Общая трудоёмкость дисциплины	108	108
<i>Аудиторные занятия:</i>	64	64
Лекции (Л)	32	32
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	32	32
Лабораторные работы (ЛР)	0	0
<i>Самостоятельная работа (СРС)</i>	35,5	35,5
Выполнение домашних заданий	30	30
Подготовка к экзамену	5,5	5,5
Консультации и промежуточная аттестация	8,5	8,5
Вид контроля (зачет, диф.зачет, экзамен)	-	экзамен

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Введение в DevOps и эволюция архитектур	8	4	4	0
2	Основы микросервисной архитектуры и проектирование	12	6	6	0
3	Контейнеризация и оркестрация. Инфраструктура как код	18	8	10	0
4	Непрерывная интеграция и доставка (CI/CD). Автоматизация	16	8	8	0
5	Наблюдаемость, мониторинг, безопасность и эксплуатация микросервисов	10	6	4	0

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Эволюция архитектур: монолит, модульный монолит, SOA, микросервисы. Сравнение, преимущества и недостатки. Примеры реальных кейсов.	2
2	1	Основные типы архитектур ПО (монолитная и микросервисная). Архитектурные паттерны и шаблоны проектирования. Преимущества и недостатки микросервисной архитектуры	2
3	2	Проектирование микросервисов: Domain-Driven Design (DDD) и декомпозиция.	2
4	2	Коммуникация в распределенных системах: синхронные и асинхронные паттерны.	4
5	3	Контейнеризация. Docker как стандарт индустрии.	2
6	3	Оркестрация контейнеров. Введение в Kubernetes (K8s).	2
7	3	Kubernetes: управление конфигурацией, хранением и безопасностью.	2
8	3	Инфраструктура как код (IaC). Принципы и инструменты.	2
9	4	Непрерывная интеграция (CI). Принципы и инструменты автоматизации сборки и тестирования.	4
10	4	Непрерывная доставка и развертывание (CD). Стратегии релизов.	4
11	5	Наблюдаемость (Observability). Логирование в распределенных системах	2
12	5	Мониторинг и метрики. От сбора данных до алертинга.	2
13	5	Безопасность в жизненном цикле DevOps (DevSecOps).	2

5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов
1	1	Введение в инструменты. Работа в команде с Git.	4
2	2	Проектирование и создание первого микросервиса	2
3	2	Контейнеризация сервиса. Dockerfile и Docker Compose.	4
4	3	Межсервисное взаимодействие. Создание второго микросервиса и коммуникация.	4
5	3	Первое развертывание в Kubernetes.	2
6	3	Конфигурация, секреты и управление состоянием в K8s.	4
7	4	Основы CI. Настройка пайплайна для сборки и тестирования.	4
8	4	Непрерывная доставка (CD). Автоматический деплой в Dev-окружение.	2
9	4	Расширяемость микросервисов. Поддержка горизонтального масштабирования путём добавления новых инстансов микросервисов.	2
10	5	Система мониторинга и журналирования	4

5.3. Лабораторные работы

Не предусмотрены

5.4. Самостоятельная работа студента

Выполнение СРС			
Подвид СРС	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов
Выполнение домашних заданий	Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система.	6	30
Подготовка к экзамену	Кочер, П. С. Микросервисы и контейнеры Docker : руководство / П. С. Кочер ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2019. — 240 с. — ISBN 978-5-97060-739-8. — Текст : электронный // Лань : электронно-библиотечная система.	6	5,5

6. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

6.1. Контрольные мероприятия (КМ)

№	Се-	Вид	Название	Вес	Макс.	Порядок начисления баллов	Учи-
---	-----	-----	----------	-----	-------	---------------------------	------

КМ	местр	контроля	контрольного мероприятия		балл		тыва- ется в ПА
1	6	Текущий контроль	Создание и тестирование микросервиса	1	5	1 балл - задание выполнено не полностью, не продемонстрированы результаты выполнения практической работы 2 балла - задание выполнено не полностью, не даны ответы на теоретические вопросы 3 балла - продемонстрированы результаты выполнения практической работы, есть существенные ошибки в решении, даны не четкие ответы на теоретические вопросы 4 балла - продемонстрированы результаты выполнения практической работы, есть незначительные ошибки в решении или даны не четкие ответы на теоретические вопросы 5 баллов - продемонстрированы результаты выполнения практической работы, нет ошибок в результатах задания, даны ответы на теоретические вопросы по материалам данной работы	экзамен
2	6	Текущий контроль	Контейнеризация микросервиса	1	5	1 балл - задание выполнено не полностью, не продемонстрированы результаты выполнения практической работы 2 балла - задание выполнено не полностью, не даны ответы на теоретические вопросы 3 балла - продемонстрированы результаты выполнения практической работы, есть существенные ошибки в решении, даны не четкие ответы на теоретические вопросы 4 балла - продемонстрированы результаты выполнения практической работы, есть незначительные ошибки в решении или даны не четкие ответы на теоретические вопросы 5 баллов - продемонстрированы результаты выполнения практической работы, нет ошибок в результатах задания, даны ответы на теоретические вопросы по материалам данной работы	экзамен
3	6	Текущий контроль	Автоматизация	1	5	1 балл - задание выполнено не полностью, не продемонстрированы результаты выполнения практической работы 2 балла - задание выполнено не полностью, не даны ответы на теоретические вопросы 3 балла - продемонстрированы результаты выполнения практической	экзамен

						работы, есть существенные ошибки в решении, даны не четкие ответы на теоретические вопросы 4 балла - продемонстрированы результаты выполнения практической работы, есть незначительные ошибки в решении или даны не четкие ответы на теоретические вопросы 5 баллов - продемонстрированы результаты выполнения практической работы, нет ошибок в результатах задания, даны ответы на теоретические вопросы по материалам данной работы	
4	6	Текущий контроль	Мониторинг	1	5	1 балл - задание выполнено не полностью, не продемонстрированы результаты выполнения практической работы 2 балла - задание выполнено не полностью, не даны ответы на теоретические вопросы 3 балла - продемонстрированы результаты выполнения практической работы, есть существенные ошибки в решении, даны не четкие ответы на теоретические вопросы 4 балла - продемонстрированы результаты выполнения практической работы, есть незначительные ошибки в решении или даны не четкие ответы на теоретические вопросы 5 баллов - продемонстрированы результаты выполнения практической работы, нет ошибок в результатах задания, даны ответы на теоретические вопросы по материалам данной работы	экзамен
5	6	Промежуточная аттестация	Экзамен	-	5	5 баллов - дан четкий ответ на билет	экзамен

6.2. Процедура проведения, критерии оценивания

Не предусмотрены

6.3. Паспорт фонда оценочных средств

Компетенции	Результаты обучения	№ KM				
		1	2	3	4	5
ПК-1	Знает: возможности используемой системы управления версиями и вспомогательных инструментальных программных средств; установленный регламент использования системы управления версиями.	+				+
ПК-1	Умеет: регистрировать изменения исходного текста программного кода в системе управления версиями, сохранять изменения программного кода в соответствии с регламентом управления версиями, выполнять слияние, разделение и сравнение исходных текстов программного кода.	+				

ПК-1	Имеет практический опыт: использования систем управления версиями и вспомогательных инструментальных программных средств при разработке и отладке программного кода	+			
ПК-3	Знает: методы и средства сборки и интеграции программных модулей и компонентов, проверки работоспособности выпусков программных продуктов		++		+
ПК-3	Умеет: выполнять процедуры сборки программных модулей и компонентов в программный продукт; производить настройки параметров программного продукта и осуществлять запуск процедур сборки; проводить проверку работоспособности программного продукта		++		
ПК-3	Имеет практический опыт: выполнения интеграции программных модулей и компонентов и проверки работоспособности выпусков программного продукта		++		
ПК-12	Знает: методы и средства мониторинга производительности компьютерного программного обеспечения; метрики производительности программного обеспечения; современные инструменты мониторинга производительности программного обеспечения				++
ПК-12	Умеет: применять методы и средства мониторинга производительности компьютерного программного обеспечения; интерпретировать диагностические данные мониторинга производительности компьютерного программного обеспечения; определять "узкие места" программного обеспечения				+
ПК-12	Имеет практический опыт: мониторинга производительности программного обеспечения				+

Типовые контрольные задания по каждому мероприятию находятся в приложениях.

7. Учебно-методическое и информационное обеспечение дисциплины

Печатная учебно-методическая документация

а) основная литература:

Не предусмотрена

б) дополнительная литература:

Не предусмотрена

в) отечественные и зарубежные журналы по дисциплине, имеющиеся в библиотеке:

Не предусмотрены

г) методические указания для студентов по освоению дисциплины:

1. Сборка Docker

из них: учебно-методическое обеспечение самостоятельной работы студента:

1. Сборка Docker

Электронная учебно-методическая документация

№	Вид литературы	Наименование ресурса в электронной форме	Библиографическое описание
1	Основная	ЭБС издательства	Кочер, П. С. Микросервисы и контейнеры Docker : руководство /

	литература	Лань	П. С. Кочер ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2019. — 240 с. — ISBN 978-5-97060-739-8. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/123710 (дата обращения: 01.12.2025)
2	Основная литература	ЭБС издательства Лань	Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. — ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/131719 (дата обращения: 01.12.2025)
3	Основная литература	ЭБС издательства Лань	Карнелл, Д. Микросервисы Spring / Д. Карнелл, И. У. Санчес ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2022. — 490 с. — ISBN 978-5-97060-971-2. — Текст : электронный // Лань : электронно-библиотечная система. https://e.lanbook.com/book/241172 (дата обращения: 01.12.2025)

Перечень используемого программного обеспечения:

1. -Oracle VirtualBox(бессрочно)
2. Docker-Docker(бессрочно)

Перечень используемых профессиональных баз данных и информационных справочных систем:

Нет

8. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Практические занятия и семинары		Компьютерный класс
Лекции		Мультимедийная аудитория с проектором