Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Макаренко Елена Николаевна
Должность: Ректор
Дата подписания: 17.09.2026 18:25:25
Уникальный программный ключ:
c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Ростовский государственный экономический университет (РИНХ)»

УТВЕРЖДАЮ
Начальник
учебно-методического управления
Т.К. Платонова
«25» мая 2026 г.

Рабочая программа дисциплины
Проектирование микросервисных архитектур программных систем

Направление подготовки
09.03.03 Прикладная информатика

Направленность (профиль) программы бакалавриата
09.03.03.02 Разработка и управление программными проектами в цифровой экономике

Для набора 2026 года

Квалификация
Бакалавр

КАФЕДРА **Информационных систем и прикладной информатики**

Распределение часов дисциплины по семестрам / курсам

Курс	4		Итого	
Вид занятий	УП	РП		
Лекции	4	4	4	4
Лабораторные	4	4	4	4
Итого ауд.	8	8	8	8
Контактная работа	8	8	8	8
Сам. работа	96	96	96	96
Часы на контроль	4	4	4	4
Итого	108	108	108	108

ОСНОВАНИЕ

Учебный план утвержден учёным советом Университета (протокол № 9 от 03.03.2026 г.).

Программу составил(и): д.э.н., профессор, Долженко А.И.

Зав. кафедрой: д.э.н., профессор С.М. Щербаков

Методический совет: д.э.н., профессор Е.Н. Тищенко

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	формирование у обучающихся теоретических знаний и практических навыков по проектированию микросервисных архитектур, разработке программного обеспечения, решения прикладных задач с использованием современных информационно-коммуникационных технологий.

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

ПК-2. Способен развертывать, настраивать, эксплуатировать и сопровождать информационные системы и сервисы

ПК-7. Способен осуществлять контроль качества в программных проектах

В результате освоения дисциплины обучающийся должен:

Знать:

перспективные архитектурные решения для программных систем и информационные технологии, модели команд для гибкой технологии разработки программных систем и подходы к управлению жизненным циклом программных продуктов (соотнесено с индикатором ПК-2.1)

методы модульного, интеграционного, нагрузочного, функционального и приемочного тестирования микросервисных программных систем (соотнесено с индикатором ПК-7.1)

Уметь:

использовать математические методы обработки, анализа и синтеза результатов разработки программных систем, обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений, проводить исследования производительности программных систем, верификации и тестирования (соотнесено с индикатором ПК-2.2)

разрабатывать, собирать и запускать модульные, интеграционные, нагрузочные, функциональные и приемочные тесты микросервисных программных систем (соотнесено с индикатором ПК-7.2)

Владеть:

навыками использования инструментария профилирования, верификации и тестирования программных систем, проектирования современных архитектурных решений и информационные технологии, использования гибких методологий и инструментальных средств управления жизненным циклом программных приложений (соотнесено с индикатором ПК-2.3)

навыками работы с инструментарием создания и проведения модульных, интеграционных, нагрузочных, функциональных и приемочных тестов микросервисных программных систем (соотнесено с индикатором ПК-7.3)

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Раздел 1. Микросервисные архитектуры

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
1.1	Тема 1. Эволюция архитектур программных приложений. Приводится обзор N-уровневых архитектур, монолитных архитектур и микросервисных архитектур. Демонстрируются приемы использования Spring, Spring Boot и Spring Cloud для создания микросервисов, а также описываются модели облачных вычислений	Лекционные занятия	4	2	ПК-2 ПК-7
1.2	Тема 2. Технологии Spring Cloud для построения микросервисов. Делается обзор Spring Cloud, который представляет коллекцию инструментов, объединяющую разработки с открытым исходным кодом многих компаний, таких как VMware, HashiCorp и Netflix. Spring Cloud упрощает установку и настройку наших проектов и предоставляет реализации шаблонов, особенно часто встречающихся в приложениях на основе Spring.	Самостоятельная работа	4	2	ПК-2 ПК-7
1.3	Тема 3. Микросервисная архитектура. Рассматриваются общие сведения о микросервисной архитектуре. Проводится сравнение монолитной и микросервисных архитектур, показаны преимущества микросервисной архитектуры для современных сложных, распределенных бизнес-приложений.	Самостоятельная работа	4	2	ПК-2 ПК-7
1.4	Тема 4. Создание микросервисов с использованием Spring Boot. Рассматриваются общие сведения о технологии создания микросервисов. При этом рассматриваются точки зрения архитекторов, разработчиков и инженеров DEV OPS на процесс создания и развертывания микросервисов.	Самостоятельная работа	4	2	ПК-2 ПК-7
1.5	Тема 5. Контейнеры в микросервисной архитектуре. Рассматриваются общие сведения о Docker и Docker Compose. Анализируются процессы развертывания микросервисов в Docker: создание образов (image) для микросервисов с помощью файлов Dockerfile и построение контейнеров, а также запуск контейнеров и тестирование HTTP-запросов.	Самостоятельная работа	4	2	ПК-2 ПК-7

1.6	Тема 6. Управление конфигурацией приложения. Рассматриваются общие сведения об управлении конфигурацией приложений, включая архитектуру управления конфигурацией, настройку сервера Spring Cloud Configuration Server, интеграцию сервера конфигурации с клиентом Spring Boot, подключение к источнику данных с использованием сервера конфигурации, защиту конфигурационных настроек в конфигурации с помощью симметричного шифрования.	Самостоятельная работа	4	2	ПК-2 ПК-7
1.7	Тема 7. Обнаружение служб в облаке. Рассматриваются общие сведения о маршрутизации служб, включая архитектуру обнаружение служб, создание и регистрацию службы в Spring Eureka, использование механизма обнаружения служб.	Самостоятельная работа	4	2	ПК-2 ПК-7
1.8	Тема 8. Тестирование микросервисов. Разработка модульных тестов для доменных сущностей, объектов значений, повествований, доменных сервисов, контроллеров, обработчиков событий и сообщений	Самостоятельная работа	4	2	ПК-2 ПК-7
1.9	Лаб. 1. Разработка микросервисов. Получить практические навыки в проектировании и разработке микросервисов.	Лабораторные занятия	4	2	ПК-2 ПК-7
1.10	Лаб. 2. Упаковка микросервиса в контейнер. Получить практические навыки в создании образов приложений с помощью файла Dockerfile, развертывании и запуске образов в контейнере, а также разработке и применении Docker Compose для запуска образов приложений, опубликованных в Docker Hub	Самостоятельная работа	4	4	ПК-2 ПК-7
1.11	Лаб. 3. Управление конфигурацией микросервисов. Получить практические навыки в создании и настройке Spring Cloud Configuration Server, интеграции Spring Cloud Config с клиентом Spring Boot, подключении к источнику данных с использованием Spring Cloud Config Server и защите конфиденциальных настроек в конфигурации.	Самостоятельная работа	4	4	ПК-2 ПК-7
1.12	Лаб.4. Тестирование микросервисов. Получить практические навыки в разработке модульных тестов для доменных сущностей, объектов значений, повествований, доменных сервисов, контроллеров, обработчиков событий и сообщений	Самостоятельная работа	4	4	ПК-2 ПК-7
1.13	Миграция монолитов на микросервисы.	Самостоятельная работа	4	22	ПК-2 ПК-7

Раздел 2. Управление микросервисами

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
2.1	Тема 9. Маршрутизация служб. Обнаружение служб играет важную роль для облачных приложений. Рассматривается механизм обнаружения служб с и дего соотношение с балансировщиком нагрузки. Анализируется порядок настройки Spring Netflix Eureka Server, приемы регистрации микросервисов Spring Boot в Eureka, использование библиотеки Spring Cloud Load и Balancer для балансировки нагрузки на стороне клиента.	Лекционные занятия	4	2	ПК-2 ПК-7
2.2	Тема 10. Безопасность микросервисов. Защита микросервисов – сложная и трудоемкая задача, которая включает реализацию нескольких уровней защиты, в том числе: прикладного уровня, инфраструктурного уровня, сетевого уровня. Рассказываются вопросы аутентификации и авторизации пользователей на прикладном уровне. Для реализации авторизации и аутентификации используется модуль Spring Cloud Security, а для защиты служб на основе Spring – Keycloak, программный компонент с открытым исходным кодом, реализующий управление идентификацией и доступом для современных приложений и служб.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.3	Тема 11. Событийно-ориентированная архитектура. Рассматриваются особенности событийно-ориентированных архитектур, использование Spring Cloud Stream для упрощения обработки событий, описывается порядок настройки Spring Cloud Stream, приемы публикации сообщений с использованием Spring Cloud Stream и Kafka, приемы получения сообщений с использованием Spring Cloud Stream и Kafka.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.4	Тема 12. Распределенные транзакции в микросервисной архитектуре. В микросервисной архитектуре транзакции охватывают несколько сервисов, каждый из которых имеет свою БД. В таких условиях приложение должно использовать более продуманный механизм работы с транзакциями. Традиционный подход к распределенным транзакциям нежизнеспособен в современных приложениях. Вместо него системы на основе микросервисов должны применять повествования. Повествования — это механизм, обеспечивающий согласованность данных в микросервисной архитектуре без применения распределенных транзакций.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.5	Тема 13. Распределенная трассировка.	Самостоятельная	4	2	ПК-2

	Рассматривается Spring Cloud Sleuth для внедрения трассировочной информации в запросы к службам, приемы агрегирования для просмотра журналов распределенных транзакций, способы преобразования, поиска, анализа и визуализации данных из журналов в масштабе реального времени. Показано, что трассировка пользовательских транзакций, охватывает несколько служб.	работа			ПК-7
2.6	Тема 14. Интеграционное и функциональное тестирование микросервисов. Интеграционные тесты проверяют корректность взаимодействия сервиса с инфраструктурой, включая базы данных и другие сервисы приложения. Обсуждаются компонентные тесты, которые являются формой приемочного тестирования сервисов. Компонентный тест проверяет работу сервиса в изоляции, используя заглушки вместо его зависимостей. Анализируются сквозные тесты, которые тестируют группу сервисов или целое приложение.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.7	Тема 15. Развертывание микросервисов. Рассматривается развертывание микросервисов, которое представляет собой сочетание двух взаимосвязанных концепций — процесса и архитектуры. Процесс развертывания заключается в доставке кода в промышленную среду и состоит из этапов, которые должны выполнить люди — разработчики или системные администраторы. Архитектура развертывания определяет структуру среды, в который этот код будет выполняться.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.8	Тема 16. Примеры реализации микросервисных архитектур. Рассматриваются примеры эффективной реализации микросервисных архитектур, анализируются достоинства, недостатки и перспективы развития.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.9	Лаб.5. Маршрутизация служб на базе Spring Eureka. Получить практические навыки в использовании архитектуры обнаружение служб, создании и регистрации службы в Spring Eureka, в использование механизма обнаружения служб.	Лабораторные занятия	4	2	ПК-2 ПК-7
2.10	Лаб. 6. Безопасность микросервисов. Получить практические навыки обеспечения защита микросервисов на прикладном уровне, инфраструктурном уровне, сетевом уровне на базе модулей Spring Cloud Security и Spring – Keycloak.	Самостоятельная работа	4	4	ПК-2 ПК-7
2.11	Лаб. 7. Распределенные транзакции. Получить практические навыки в управлении распределенными транзакциями на основе повествования, которое обеспечивает согласованность данных в микросервисной архитектуре.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.12	Лаб. 8. Распределенная трассировка. Получить практические навыки внедрения трассировочной информации в запросы к службам, приемы агрегирования для просмотра журналов распределенных транзакций, способы преобразования, поиска, анализа и визуализации данных из журналов в масштабе реального времени.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.13	Лаб. 9. Интеграционное и функциональное тестирование микросервисов. Получить практические навыки интеграционного, компонентного и сквозного тестирования .	Самостоятельная работа	4	2	ПК-2 ПК-7
2.14	Лаб.10. Развертывание микросервисов. Получить практические навыки развертывания микросервисов для доставки кода в промышленную среду.	Самостоятельная работа	4	2	ПК-2 ПК-7
2.15	Разработка бизнес-логики микросервисов	Самостоятельная работа	4	22	ПК-2 ПК-7
2.16	Подготовка к промежуточной аттестации	Зачет	4	4	ПК-2 ПК-7

4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Структура и содержание фонда оценочных средств для проведения текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Учебные, научные и методические издания

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
1	Емельянов А. А.	Прикладная информатика: журнал	Москва: Синергия ПРЕСС, 2006	ЭБС «Университетская библиотека онлайн»

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
2	Назаров С. В., Белоусова С. Н., Бессонова И. А., Гиляревский Р. С., Гудыно Л. П.	Введение в программные системы и их разработку: курс лекций	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	ЭБС «Университетская библиотека онлайн»
3	Битюцкая, Н. И.	Разработка программных приложений: лабораторный практикум	Ставрополь: Северо-Кавказский федеральный университет, 2015	ЭБС «IPR SMART»
4	Лисяк, В. В.	Разработка информационных систем: учебное пособие	Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2019	ЭБС «IPR SMART»
5	Лисяк В. В.	Разработка информационных систем: учебное пособие	Ростов-на-Дону, Таганрог: Южный федеральный университет, 2019	ЭБС «Университетская библиотека онлайн»

5.2. Профессиональные базы данных и информационные справочные системы

ИСС «КонсультантПлюс»
ИСС «Гарант»
Национальная электронная библиотека (НЭБ), https://rusneb.ru/

5.3. Перечень программного обеспечения

Операционная система РЕД ОС
Axiom GDK
GigaCode
Amplicode
Spring Boot
PostgreSQL
LibreOffice

5.4. Учебно-методические материалы для обучающихся с ограниченными возможностями здоровья

При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения:
- столы, стулья;
- персональный компьютер / ноутбук (переносной);
- проектор;
- экран / интерактивная доска.

Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными и/или свободно распространяемыми программными средствами и выходом в Интернет, и/или в специализированных лабораториях, предусмотренных образовательной программой.
Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.

Приложение 1

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

1. Описание показателей и критериев оценивания компетенций на различныхэтапах их формирования, описание шкал оценивания

1.1. Показатели и критерии оценивания компетенций:

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ПК-2: Способен развертывать, настраивать, эксплуатировать и сопровождать информационные системы и сервисы			
Знать перспективные архитектурные решения для программных систем и информационные технологии, модели команд для гибкой технологии разработки программных систем и подходы к управлению жизненным циклом программных продуктов	Знает перспективные архитектурные решения для программных систем, этапы жизненного цикла приложений, модели команд	Полнота и содержательность ответа Умение приводить примеры	Вопросы к зачету (1-19), тест (1-10), лабораторные задания (1-2)
Уметь использовать математические методы обработки, анализа и синтеза результатов разработки программных систем, обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений, проводить исследования производительности программных систем, верификации и тестирования	Применяет методы и подходы разработки микросервисных архитектур, методы оптимизации для улучшениякачества работы приложений; исследует производительность микросервисных архитектур в производственных системах	Полнота и содержательность ответа Умение Самостоятельно находить решение поставленных задач	Вопросы к зачету (1-19), тест (1-10), лабораторные задания (1-2)
Владеть навыками использования инструментария профилирования, верификации и тестирования программных систем, проектирования современных архитектурных решений и информационные технологии, использования гибких методологий и инструментальных средств управления жизненным циклом программных приложений	Использует фреймворки Spring Boot, Spring Cloud для создания микросервисных архитектур различного назначения	Полнота и содержательность ответа Умение приводить примеры Умение самостоятельно находить решение поставленных задач	Вопросы к зачету (1-19), тест (1-10), лабораторные задания (1-2)
ПК-7: Способен осуществлять контроль качества в программных проектах			
Знать методы модульного, интеграционного, нагрузочного, функционального и приемочного тестирования микросервисных программных систем	Знает методология модульного, интеграционного, нагрузочного, функционального и приемочного тестирования микросервисных программных систем	Полнота и содержательность ответа Умение приводить примеры	Вопросы к зачету (1-19), тест (1-10), лабораторные задания (1-2)
Уметьразрабатывать, собирать и запускать модульные, интеграционные, нагрузочные, функциональные и приемочные тесты микросервисных программных систем	Использует технологии разработки и запуска модульных, интеграционных, нагрузочных, функциональных и приемочных тестов	Полнота и содержательность ответа, умение самостоятельно находить решение поставленных задач	Вопросы к зачету (1-19), тест (1-10), лабораторные задания (1-2)

	микросервисных программных систем.		
Владеть навыками работы с инструментарием создания и проведения модульных, интеграционных, нагрузочных, функциональных и приемочных тестов микросервисных программных систем	Показывает навыки создания и проведения модульных, интеграционных, нагрузочных, функциональных и приемочных тестов микросервисных программных систем.	Полнота и содержательность ответа, умение приводить примеры, умение самостоятельно находить решение поставленных задач	Вопросы к зачету (1-19), тест (1-10), лабораторные задания (1-2)

1.2 Шкалы оценивания:

Текущий контроль успеваемости и промежуточная аттестация осуществляется в рамках накопительной балльно-рейтинговой системы в 100-балльной шкале:

50-100 баллов (зачтено)

0-49 баллов (не зачтено)

2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Вопросы к зачету

1. Эволюция архитектуры микросервисов
2. N-уровневая архитектура приложений
3. Монолитная архитектура приложений
4. Микросервисная архитектура приложений
5. Целесообразности перехода на микросервисы
6. Микросервисы со Spring
7. Облачные приложения
8. Приложения на основе микросервисов
9. Создание микросервиса с помощью Spring Boot
10. Преимущества облачных вычислений и микросервисов
11. Базовый шаблон разработки микросервисов
12. Шаблоны маршрутизации микросервисных приложений
13. Устойчивость клиентов микросервисных приложений
14. Шаблоны безопасности микросервисных приложений
15. Шаблоны журналирования микросервисных приложений
16. Шаблоны трассировки микросервисных приложений
17. Шаблон сбора метрик микросервисного приложения
18. Шаблоны сборки микросервисов
19. Шаблоны развертывания микросервисов

Задание включает два вопроса – один теоретический вопроса и одно практико-ориентированное задание из числа приведенных ниже лабораторных заданий.

Критерии оценивания:

- 50-100 баллов (зачтено) – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение и знакомство с литературой; наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются отдельные логические

и стилистические погрешности, обучающийся усвоил литературу, рекомендованную в рабочей программе дисциплины; наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- 0-49 баллов (не зачтено) – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

Тест

1. Назовите преимущества корпоративных n-уровневые приложений
   позволяют четко разделить задачи и рассматривать пользовательский интерфейс, данные и бизнес-логику по отдельности
   команды разработчиков могут работать над различными компонентами независимо друг от друга
   относительно легко найти квалифицированных разработчиков для многоуровневых проектов
   упрощается рефакторинг n-уровневого приложения

2. Назовите недостатки корпоративных n-уровневые приложений
   после внесения изменений в код приходится повторно запускать всё приложение
   сообщения курсируют вверх и вниз через уровни, что может быть неэффективным
   рефакторинг n-уровневого приложения после развертывания является сложной задачей
   команды разработчиков могут самостоятельно работать над различными компонентами

3. Назовите компоненты монолитного приложения
   пользовательский интерфейс
   бизнес-логика
   логика доступа к базе данных
   система обработки сообщений

4. Укажите правильные утверждения в отношении монолитных приложений
   монолитные приложения проще создавать чем более сложные многоуровневые архитектуры
   монолитные приложения проще развертывать, чем микросервисные архитектуры
   в монолитные приложения достаточно просто вносить изменения
   монолитные приложения обладают большой гибкостью и удобством сопровождения

5. Укажите правильное определение микросервиса
   это небольшая, слабо связанная распределенная служба
   это многоуровневое приложение
   это n-уровневое приложение
   это небольшое монолитное приложение

6. Укажите правильные утверждения в отношении микросервисов
   ~%50% микросервис представляет собой компонент с четко определенными обязанностями
   микросервисы являются простыми в управлении компонентами
   микросервисы приложения достаточно сильно взаимосвязаны
   микросервисы приложения имеют общую базу кода

7. Укажите правильные утверждения в отношении микросервисной архитектуры приложения
   разработчики микросервиса имеют свою, ориентированную на микросервис, базу кода
   разработчики микросервиса имеют свою, ориентированную на микросервис, базу данных
   разработчики приложения имеют общую базу кода
   разработчики приложения имеют общую интегрированную базу данных

8. Укажите правильные утверждения в отношении микросервисной архитектуры приложения

разработчики микросервиса имеют свой, ориентированный на микросервис, конвейер непрерывной интеграции
разработчики микросервиса имеют свою, ориентированную на микросервис, базу данных
разработчики приложения имеют универсальный конвейер непрерывной интеграции приложения
разработчики приложения имеют общую библиотеку модульного тестирования

9. Укажите правильные утверждения в отношении процесса разработки микросервисной архитектуры приложения
команда разработчиков микросервиса самостоятельно управляет разработкой кода микросервиса
команда разработчиков микросервиса самостоятельно определяет инфраструктуру микросервиса
архитетура базы данных микросервисных приложений разрабатывается с учетом общих требований к приложению
~%-50% при разработке микросервисных приложений репозиторий системы контроля версий должен быть общим для приложения

10. Какими характеристиками обладает микросервисная архитектура
~%30% логика приложения разбита на мелкие компоненты с четко определенными согласованными границами ответственности
~%30% каждый компонент отвечает за узкий круг задач
~%40% каждый компонент развертывается независимо от других
~%-100% компоненты должны быть написаны на одном языке программирования

Критерии оценивания:
Из имеющегося банка тестов формируется тестовое задание, содержащее 10 тестовых вопросов для одного обучающегося. Каждый тестовый вопрос содержит 4 варианта ответов, один или несколько из которых – верные.

- 17-20 б. – тест пройден на 85-100 %;
- 7-16 б. – тест пройден на 35-84 %;
- 0-6 б. – тест пройден на менее, чем 35 %.

Максимальное количество баллов за тест – 20.

Лабораторные задания

Лабораторное задание №1
Разработка микросервисов.
Цель работы: Получить практические навыки в проектировании и разработке микросервисов.

Лабораторное задание №2
Упаковка микросервиса в контейнер
Цель работы: получить навыки подготовки образов для приложений и упаковка их в Docker/

Критерии оценивания:

- 30-40 б. – задание выполнено верно;
- 20-29 б. – при выполнении задания были допущены неточности, не влияющие нарезультат;
- 10-19 б. – при выполнении задания были допущены ошибки;
- 0-9 б. – при выполнении задания были допущены существенные ошибки.

Максимальное количество баллов за лабораторные задания – 80 (2 задания по 40баллов).

3. Методические материалы, определяющие процедуры оценивания знаний, умений,навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

Текущий контроль успеваемости проводится с использованием оценочных средств,

представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

Промежуточная аттестация проводится в форме зачета.

Зачет проводится по расписанию промежуточной аттестации в устном виде. Количество вопросов в задании – 2 (один теоретический вопрос и одно практико- ориентированное задание). Проверка ответов и объявление результатов производится в день зачета. Результаты аттестации заносятся в ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику, должны ликвидировать задолженность в установленном порядке.

Приложение 2

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные занятия.

В ходе лекционных занятий рассматриваются основные теоретические вопросы, даются рекомендации для самостоятельной работы и подготовке к лабораторным занятиям.

В ходе лабораторных занятий углубляются и закрепляются знания студентов по ряду рассмотренных на лекциях вопросов, развиваются навыки практической работы.

При подготовке к лабораторным занятиям каждый студент должен:

– изучить рекомендованную учебную литературу;
– изучить конспекты лекций;
– подготовить ответы на все вопросы по изучаемой теме.

В процессе подготовки к лабораторным занятиям студенты могут воспользоваться консультациями преподавателя.

Вопросы, не рассмотренные на лекциях и лабораторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий посредством тестирования и выполнения лабораторных заданий. В ходе самостоятельной работы каждый студент обязан прочитать дополнительную литературу по изучаемой теме, дополнить конспекты лекций недостающим материалом, выписками из рекомендованных первоисточников, выделить непонятные термины, найти их значение в энциклопедических словарях.

Студент должен готовиться к предстоящему лабораторному занятию по всем обозначенным в рабочей программе дисциплины вопросам.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации студенты могут воспользоваться электронно-библиотечными системами. Также обучающиеся могут взять на дом необходимую литературу на абонементе университетской библиотеки или воспользоваться читальными залами.