

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Ректор Белорусского
государственного университета

А.Д.Король

26 мая 2025 г.

Регистрационный № 3191/б.



СЕРВИС-ОРИЕНТИРОВАННЫЕ ПРОГРАММНЫЕ СИСТЕМЫ

Учебная программа учреждения образования по учебной дисциплине для
специальности:

6-05-0533-11 Прикладная информатика

Профилизация: Комплексное проектирование и управление жизненным циклом
сложных информационных систем

2025 г.

Учебная программа составлена на основе ОСВО 6-05-0533-11-2023, учебного плана БГУ № 6-5.3-59/05 от 15.05.2023.

СОСТАВИТЕЛЬ:

И.С.Войтешенко, доцент кафедры технологий программирования факультета прикладной математики и информатики Белорусского государственного университета, кандидат технических наук, доцент

РЕЦЕНЗЕНТ:

Д.В.Баровик, заместитель директора ОАО «Центр банковских технологий», кандидат физико-математических наук, доцент

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой технологий программирования БГУ
(протокол № 17 от 15.05.2025)

Научно-методическим советом БГУ
(протокол № 10 от 22.05.2025)

Заведующий кафедрой



А.Н.Курбацкий

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель учебной дисциплины – получение студентами базовых знаний в построении систем сервис-ориентированной архитектуры (СОА), в том числе приложений микросервисной архитектуры; освоение методов и компьютерных технологий, применяемых для разработки и обеспечения безопасности систем и приложений указанных классов.

Задачи учебной дисциплины:

- лекции должны знакомить студентов с парадигмой сервис-ориентированной архитектуры; принципами, подходами и технологиями разработки сервисов и систем СОА; микросервисной архитектурой и контейнеризацией; со способами и технологиями управления микрослужбами и многоконтейнерными приложениями; с проблемами и решениями обеспечения безопасности микрослужб и сервис-ориентированных систем;
- лабораторные работы должны быть связаны с освоением способов проектирования и разработки систем СОА; с овладением технологиями разработки веб-сервисов и контейнеризации монолитных приложений;
- самостоятельная работа должна развивать навыки самостоятельного освоения методов выбора и применения способов и технологий проектирования и разработки сервисов, и систем СОА, навыки коллективной разработки, развертывания и тестирования систем СОА.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к дисциплинам профилизации компонента учреждения образования.

Программа составлена с учетом **межпредметных связей** с учебными дисциплинами. Основой для изучения учебной дисциплины являются дисциплины модуля «Программирование», а также дисциплина «Операционные системы». Дисциплина «Сервис-ориентированные программные системы» способствует успешному прохождению практики и написанию дипломных работ.

Требования к компетенциям

Освоение учебной дисциплины «Сервис-ориентированные программные системы» должно обеспечить формирование следующих компетенций:

универсальные компетенции:

Владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации;

Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий;

Работать в команде, толерантно воспринимать социальные, этнические, конфессиональные, культурные и иные различия;

Быть способным к саморазвитию и совершенствованию в профессиональной деятельности;

Проявлять инициативу и адаптироваться к изменениям в профессиональной деятельности.

базовые профессиональные компетенции:

Строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного и иных парадигм программирования.

Применять знания в области принципов функционирования, архитектур и программных реализаций операционных систем, структурной организации компьютеров и компьютерных систем, методах обработки данных для выбора вычислительных средств решения практических задач;

специализированные компетенции:

Использовать программные средства и технологии для создания прикладного программного обеспечения;

Проектировать и тестировать программное обеспечение различных видов. Разрабатывать насыщенные Интернет приложения. Разрабатывать техническую документацию на программное обеспечение.

В результате изучения дисциплины студент должен:

знать:

- базовые теоретические понятия СОА, принципы и подходы к проектированию программных систем СОА;
- понятие микросервиса и его отличительные признаки, технологии и инструменты, связанные с применением микросервисов;
- основы контейнеризации микросервисных приложений, способы управления микрослужбами и многоконтейнерными приложениями,
- способы и технологии обеспечения безопасности в системах СОА и приложениях, основанных на микросервисной архитектуре;

уметь:

- ставить и решать задачи, связанные с проектированием, разработкой, защитой и развертыванием систем СОА, оценивать целесообразность применения парадигмы СОА и микросервисной архитектуры при проектировании приложений и распределенных систем;

иметь навык:

- применения технологий разработки приложений сервис-ориентированной архитектуры на основе возможностей программной платформы;
- разработки приложений Docker на основе контейнера;
- проверки подлинности с помощью возможностей программной платформы и внешних провайдеров аутентификации для обеспечения безопасности приложений СОА и приложений микросервисной архитектуры.

Структура учебной дисциплины

Дисциплина изучается в 5 семестре. В соответствии с учебным планом всего на изучение учебной дисциплины отведено для очной формы получения

высшего образования – 108 часов, в том числе 68 аудиторных часа, лекции - 34 часа, лабораторные занятия – 34 часа. Из них:

Лекции - 34 часа, лабораторные занятия – 30 часов, управляемая самостоятельная работа – 4 часа.

Трудоемкость учебной дисциплины составляет 3 зачетных единицы.

Форма промежуточной аттестации – экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Введение в сервис-ориентированную архитектуру. Веб-сервисы первого и второго поколений.

Тема 1.1. Принципы проектирования и построения сервис-ориентированных систем.

История теории и практики развития распределенных систем. Концепция сервис-ориентированной архитектуры (архитектурного стиля).

Преобразование бизнес-логики в логику приложений в соответствии с парадигмой СОА. Принципы и подходы к проектированию программных систем СОА. Интероперабельность. Слабое связывание. Веб-сервисы. Формальный контракт использования, состояние и ресурсы сервиса.

Тема 1.2. Сервисы SOAP.

Протокол SOAP. Структура SOAP-сообщения. Заголовок SOAP и его стандартные атрибуты.

Язык описания веб-сервисов WSDL. Основные элементы WSDL 2.0 файла.

Разработка SOAP веб-сервисов. Прокси-классы. Тестирование и применение SOAP веб-сервисов.

Тема 1.3. Стандарты второго поколения веб-сервисов.

Стандарты WS-*, их назначение. стек протоколов безопасности сервисов. Стандарт WS.Security. Использование заголовков SOAP для аутентификации пользователей. Стандарт WS-Addressing. Расширенная адресация конечных точек. Асинхронные вызовы. Концепция Web Service Resource Framework (WSRF). Состояние веб-сервиса и веб-ресурсы.

Тема 1.4. Веб-службы Windows Communication Foundation (WCF).

Протокол (язык) Java Script Object Notation (JSON).

Адрес (оконечная точка), привязка и контракт. Контракт о службе. Контракт о данных. Контракт о сообщениях. Стандартные привязки.

Определение контракта. Определение оконечной точки. Размещение службы в процессе с возможностью прослушивания входящих запросов.

Переход с WCF на CoreWCF, Web API, gRPC.

Раздел 2. Сервисы REST API и gRPC.

Тема 2.1. Сервисы REST API.

Ограничения архитектурного стиля REST. REST сервисы.

ASP.NET Web API как платформа для сборки REST-приложений на базе .NET (.NET Core, .NET Framework). Разработка ASP.NET Core Web API служб. Контроллеры и обработка запросов. Тестирование контроллеров. Валидация в Web API. Использование служб Web API.

Протокол OData и его использование.

Тема 2.2. Сервисы gRPC.

gRPC как вариант технологии RPC разработки распределенных систем.

Структура проекта gRPC в Visual Studio. Синтаксис proto (proto3).
Определение сервиса. Реализация сервиса. Клиент для gRPC-сервиса.

Сравнение сфер применения, преимуществ и слабых сторон gRPC и Web API.

Тема 2.3. Вопросы безопасности при использовании REST API и gRPC.

Опасные уязвимости REST API (на основании списков OWASP Top 10).
Стандарты аутентификации. JWT-токены. Спецификация OAuth 2.0.

Использование защиты данных ALTS в gRPC. Поддержка шифрования SSL/TLS в gRPC. Аутентификация на основе токенов при использовании gRPC.

Раздел 3. Сервисы в программных платформах и приложениях.

Тема 3.1. Сервисы и внедрение зависимостей.

Внедрение зависимостей (Dependency Injection) в языках программирования.

Внедрение зависимостей и сервисы в ASP.NET Core. Коллекция встроенных сервисов IServiceCollection, методы ее расширения. Информация о сервисах коллекции. Создание и использование собственных сервисов. Способы передачи зависимостей, их жизненный цикл.

Тема 3.2. Сервисы в приложениях интернета вещей.

Набор облачных сервисов Yandex Cloud. Сервис Yandex IoT Core. «Бессерверные» (serverless) сервисы. Yandex Cloud Functions. MQTT-протокол. MQTT брокеры. Инструмент визуального потокового программирования Node-RED, Интеграция Node-RED и Yandex IoT Core.

Тема 3.3. Сервисы в программных платформах моделирования мобильных роботов.

Мобильные роботы и окружающая их среда. Программные платформы моделирования мобильных роботов. Виртуальные среды. Понятия и использование сервисов в различных платформах моделирования мобильных роботов.

Раздел 4. Микросервисная архитектура.

Тема 4.1. Введение в микросервисы.

Понятие микросервиса и его основные отличительные признаки. Технологии и инструменты, связанные с применением микросервисов: контейнеры, Docker, Kubernetes. Шлюзы API. Обмен сообщениями. Потoki событий.

Тема 4.2. Микросервисы и данные.

Особенности моделирования предметных областей микрослужб. Предметно-ориентированное проектирование. Ограниченный контекст. Определение границ модели предметной области для каждой микрослужбы.

Проблемно-ориентированное проектирование (DDD), принцип Command-query separation (COS), шаблон архитектуры CQRS и их использование в микросервисах.

Сравнение владения данными: единая база данных и микрослужбы. Запросы для извлечения данных из нескольких микрослужб. Шаблон материализованного представления. Шлюз API. "Холодные данные" в центральных базах данных.

Распределенные транзакции: модульный монолит, двухфазная фиксация, согласованность в конечном счете (Saga pattern). Хореография и оркестровка.

Тема 4.3. Взаимодействие микросервисов.

Синхронное и асинхронное взаимодействие. Команды, запросы, события. Подписка на события. Асинхронное взаимодействие как способ повышения автономности микрослужб. Частые и длинные опросы.

Тема 4.4. Контейнеризация Docker.

Понятие о проекте с открытым исходным кодом Docker. Сравнение контейнеров Docker с виртуальными машинами. Терминология Docker. Контейнеры, образы и реестры Docker.

Тема 4.5. Процесс разработки приложений на основе Docker.

Выбор средств разработки. Рабочий процесс разработки приложений Docker на основе контейнера. Dockerfile. Разработка приложений Docker на основе контейнера с использованием Visual Studio.

Тема 4.6. Управление микрослужбами и многоконтейнерными приложениями.

Проблема управления микрослужбами. Управление микрослужбами при создании корпоративных приложений. Узлы, кластеры, оркестраторы, планировщики.

Тема 4.7. Балансировка нагрузки в высоконагруженных распределенных системах.

Проблемы, решаемые балансировщиками нагрузки. Классификации уровней балансировки. Алгоритмы балансировки нагрузки. Сравнение существующих решений для балансировки нагрузки.

Тема 4.8. Service Mesh (сетка сервисов).

Функциональное назначение Service Mesh в микросервисной архитектуре. Структура Service Mesh, взаимодействие компонентов. Окружение для развертывания Service Mesh. Сравнительный анализ функциональных возможностей реализаций Service Mesh.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования с применением дистанционных образовательных технологий (ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Введение в сервис-ориентированную архитектуру. Веб-сервисы первого и второго поколений.	6			2			
1.1	Принципы проектирования и построения сервис-ориентированных систем	2						Устный опрос
1.2	Сервисы SOAP	2						Устный опрос
1.3	Стандарты второго поколения веб-сервисов	1						Устный опрос
1.4	Веб-службы Windows Communication Foundation (WCF).	1			2			Отчет по лабораторной работе с устной защитой
2	Сервисы REST API и gRPC.	6			8			
2.1	Сервисы REST API.	2			2			Контрольная работа. Отчет по лабораторной работе с устной защитой
2.2	Сервисы gRPC.	2			4			Отчет по лабораторной работе с устной защитой
2.3	Вопросы безопасности при использовании REST API и gRPC.	2			2			Отчет по лабораторной работе с устной защитой. Проверка программного кода приложения

3	Сервисы в программных платформах и приложениях.	6			4		4	
3.1	Сервисы и внедрение зависимостей.	2						Устный опрос
3.2	Сервисы в приложениях интернета вещей.	2			2		2	Отчет по лабораторной работе с устной защитой.
3.3	Сервисы в программных платформах моделирования мобильных роботов.	2			2		2	Контрольная работа. Отчет по лабораторной работе с устной защитой
4	Микросервисная архитектура	16			16			
4.1	Введение в микросервисы.	2						Устный опрос
4.2	Микросервисы и данные.	2			2			Устный опрос
4.3	Взаимодействие микросервисов.	2			2			Отчет по лабораторной работе с устной защитой
4.4	Контейнеризация Docker.	2			2			Контрольная работа. Отчет по лабораторной работе с устной защитой
4.5	Процесс разработки приложений на основе Docker.	2			4			Отчет по лабораторной работе с устной защитой
4.6	Управление микрослужбами и многоконтейнерными приложениями.	2			4			Отчет по лабораторной работе с устной защитой
4.7	Балансировка нагрузки в высоконагруженных распределенных системах	2			2			Контрольная работа
4.8	Service Mesh (сетка сервисов)	2						Устный опрос
	ИТОГО	34			30		4	

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Перальта, Х. А. Микросервисы и API / Хосе Аро Перальта ; пер. с англ. А. И. Ларина ; науч. ред. А. Петраки, А. Смирнов]. – Санкт-Петербург ; Москва ; Минск : Питер, 2024. – 463 с. – URL: <https://ibooks.ru/reading.php?short=1&productid=395858>.
2. Деннис, У. Kubernetes для разработчиков / У. Деннис ; [пер. с англ. В. Сауль ; науч. ред. А. Патрин]. – Санкт-Петербург ; Москва ; Минск : Питер, 2025. – 380 с.
3. Уолш, Д. Podman в действии. / Дэниэл Уолш. – Санкт-Петербург : Питер, 2024. – 352 с. – URL: <https://ibooks.ru/reading.php?short=1&productid=393766>.
4. Стренгхольт, П. Масштабируемые данные. Лучшие шаблоны высоконагруженных архитектур / П. Стренгхольт ; [пер. с англ. С. Черников]. – Санкт-Петербург ; Москва ; Минск : Питер, 2022. – 567 с.
5. М. Клеппман. Высоконагруженные приложения, масштабирование, поддержка. – Астана: Спринт Бук, 2024 – 637 с. – URL: <https://ibooks.ru/reading.php?short=1&productid=357916>.

Дополнительная литература

1. Микрослужбы .NET: Архитектура контейнерных приложений .NET. [Электрон. ресурс]. – <https://docs.microsoft.com/ru-ru/dotnet/architecture/microservices/>
2. Cesar de la Torre. Containerized Docker Application Lifecycle with Microsoft Platform and tools. [Электрон. ресурс]. – <https://aka.ms/dockerlifecleebook>
3. Ньюмен, С. Создание микросервисов / С. Ньюмен, – М.: Прогресс книга, 2023. – 624 с.
4. Ньюмен, С. От монолита к микросервисам / С. Ньюмен, – БНВ-СПб, 2021. – 272 с. – URL: <https://ibooks.ru/reading.php?short=1&productid=385768>.
5. Беллемар, Адам. Создание событийно-управляемых микросервисов / Адам Беллемар. – БХВ-Петербург, 2023. – 320 с.
6. Индрасири К. gRPC: запуск и эксплуатация облачных приложений. Go и Java для Docker и Kubernetes / К. Индрасири, Д. Курупу. – СПб.: Питер, 2021. – 221 с. – URL: <https://ibooks.ru/reading.php?short=1&productid=374465>.
7. Дергачев А.М., Кореньков Ю.Д., Логинов И.П., Сафронов А.Г., Технологии веб-сервисов– СПб: Университет ИТМО, 2021. – 100 с.
8. Rendle, Mark. gRPC for WCF Developers / Mark Rendle, Miranda Steiner. – Microsoft Corporation, 2023. – 104 с.
9. Обновление серверного проекта WCF для использования CoreWCF в .NET 6 [Электрон. ресурс]. – <https://learn.microsoft.com/ru-ru/dotnet/core/porting/upgrade-assistant-wcf>

10. What to do with WCF in .NET 8 and Beyond [Электрон. ресурс]. – <https://blog.inedo.com/dotnet/windows-comm-framework/>
11. Радченко, Г.И. Распределенные вычислительные системы / Г.И. Радченко. – Челябинск: Фотохудожник, 2012. – 184 с.
12. Лёве, Д. Создание служб WCF. / Д. Лёве. – СПб.: Питер, 2008. – 592 с.
13. WS-Security specification (OASIS) [Электрон. ресурс]. – <http://www.oasis-open.org/specs/index.php#wssv1.0>
14. REST: From Research to Practice /E. Wilde, C. Pautasso (ред.) // Springer Science+Business Media, LLC, 2011.
15. Allamaraju, Subbu. RESTful Web Services Cookbook / Subbu Allamaraju. – СПб.: O'Reilly, 2010. – 296 с.
16. Kocher, Parminder Singh. Microservices and Containers / Parminder Singh Kocher. – Addison-Wesley Professional, 2018. – 304с.
17. Введение в API [Электрон. ресурс]. – <https://systems.education/api-intro>
18. В. И. Дравица, А. Н. Курбацкий, А. В. Решетняк, А. В. Старцев, А. А. Андрушевич, И.С. Войтешенко. Основы технологий Интернета вещей: учебно-методическое пособие для студентов учреждений высшего образования, обучающихся по специальности "Прикладная информатика (по направлениям)", направлению специальности "Прикладная информатика (программное обеспечение компьютерных систем)" / Белорусский государственный университет, Факультет прикладной информатики, Кафедра технологий программирования. - Минск : Республиканский институт высшей школы, 2022. - 108 с.
19. Техническая документация Yandex Cloud [Электронный ресурс]. — Режим доступа: <https://cloud.yandex.ru/docs/> — Дата доступа: 07.04.2025.
20. Node-RED [Electronic resource] — Mode of access: <https://nodered.org/> — Date of access: 07.04.2025
21. Андрушевич А.А., Войтешенко И.С., Емельянова О.Ю. О среде имитационного моделирования компонентов и приложений в Интернете вещей на основе инструментов визуального потокового программирования и облачных сервисов. Информатика. 2022;19(2):100-116. <https://doi.org/10.37661/1816-0301-2022-19-2-100-116>
22. What is a service mesh? // Red Hat [Electronic resource]. – Mode of access: <https://www.redhat.com/en/topics/microservices/what-is-a-service-mesh>.
23. Istio [Electronic resource]. – Mode of access: <https://istio.io/latest/>.
24. Docker [Electronic resource]. – Mode of access: <https://www.docker.com/>.
Kubernetes: лучшие практики = Kubernetes Best Practices : Раскрой потенциал главного инструмента в отрасли / Брендан Бернс [и др. ; пер. с англ. А. Павлов]. – Санкт-Петербург [и др.] : Питер, 2021. – 286 с. – URL: <https://ibooks.ru/reading.php?short=1&productid=377023>.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Объектом диагностики компетенций студентов являются знания, умения, практический опыт, полученные ими в результате изучения учебной дисциплины. Выявление учебных достижений студентов осуществляется с помощью мероприятий текущего контроля и промежуточной аттестации.

Текущий контроль работы студента проходит в следующих формах:

- технические: лабораторные работы, выполняемые на компьютере. Они оцениваются исходя из читаемости и оптимизированности программного кода, а также путём проверки программного кода на тестовых примерах;

- устно-письменные: устная и/или письменная (в виде отчёта) защита лабораторных работ, оцениваемая на основе полноты и последовательности ответа (отчёта), полноты раскрытия содержания выполненного задания, понимания работы алгоритмов и методов, использованных при выполнении задания;

- устные: устные опросы, проводимые в целях первичного мониторинга усвоения материала студентами и оцениваемые исходя из полноты и последовательности ответа, понимания основных понятий, методов и алгоритмов, изложенных на лекционных или лабораторных занятиях.

Формой промежуточной аттестации по дисциплине «Сервис-ориентированные программные системы» предусмотрен экзамен.

Формирование итоговой отметки в ходе проведения контрольных мероприятий текущей аттестации (примерные весовые коэффициенты, определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации):

- устный опрос – 10%,
- проверка программного кода лабораторных работ – 30%,
- защита отчётов по лабораторным работам – 30%,
- контрольные работы – 30%.

В случае успешной защиты отчётов по всем лабораторным работам допускается определение результатов промежуточной аттестации по дисциплине без проведения дополнительного опроса на экзамене. При этом явка обучающегося на экзамен является обязательной.

Итоговая отметка по дисциплине рассчитывается на основе итоговой отметки текущей аттестации (модульно-рейтинговой системы оценки знаний) 40 % и экзаменационной отметки 60 %.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 3.2. Сервисы в приложениях интернета вещей. (2 ч)

Задание. Ознакомиться с возможностями, предоставляемыми сервисом Yandex IoT Core.

Форма контроля – отчет по лабораторной работе с устной защитой.

Тема 3.3. Сервисы в программных платформах моделирования мобильных роботов. (2 ч)

Изучение и анализ демонстрационных (учебных) примеров, предоставляемых некоторыми средами моделирования и визуализации мобильных роботов.

Форма контроля – отчет по лабораторной работе с устной защитой.

Примерная тематика лабораторных занятий

Лабораторная работа № 1. Преобразование бизнес-логики в логику приложений в соответствии с парадигмой СОА.

Лабораторная работа № 2. Веб-службы Web API.

Лабораторная работа № 3. Веб-службы gRPC, разработка и использование.

Лабораторная работа № 4. Обеспечение безопасности приложений при использовании веб-служб Web API и gRPC.

Лабораторная работа № 5. Групповой проект: разработка распределенной системы с использованием веб-служб Web API и gRPC.

Лабораторная работа № 6. Изучить, проанализировать, запустить на выполнение и продемонстрировать работу интернет-магазина в виде контейнеризованного приложения на основе существующего проекта eShopOnContainers.

Лабораторная работа № 7. Проектирование и реализация приложения с микросервисной архитектурой.

Лабораторная работа № 8. Управление микрослужбами и многоконтейнерными приложениями.

Лабораторная работа № 9. Реализация алгоритмов балансировки нагрузки.

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используются следующие инновационные подходы:

практико-ориентированный подход, который предполагает:

- освоение содержания образования через решения практических задач;
- приобретение навыков эффективного выполнения разных видов профессиональной деятельности;
- ориентацию на генерирование идей, реализацию групповых студенческих проектов, развитие предпринимательской культуры;
- использованию процедур, способов оценивания, фиксирующих сформированность профессиональных компетенций.

метод проектного обучения, который предполагает:

- способ организации учебной деятельности студентов, развивающий актуальные для учебной и профессиональной деятельности навыки планирования, самоорганизации, сотрудничества и предполагающий создание собственного продукта;

- приобретение навыков для решения исследовательских, творческих, социальных, предпринимательских и коммуникационных задач.

Методические рекомендации по организации самостоятельной работы обучающихся

Самостоятельная работа с целью изучения материала учебной дисциплины предполагает работу с рекомендованной учебной литературой и Интернет-ресурсами. Теоретические сведения закрепляются выполнением лабораторных заданий, при выполнении которых следует руководствоваться методическими разработками, размещенными в электронной библиотеке университета и на образовательном портале. Также могут быть предложены дополнительные задания (тесты, задания для самостоятельного выполнения) для самооценки и более глубокого усвоения полученного материала.

Примерный перечень вопросов к экзамену

1. Парадигма СОА. Концепция сервис-ориентированной архитектуры (архитектурного стиля). Факторы целесообразности и нецелесообразности использования сервис-ориентированной архитектуры

2. Принципы проектирования и построения сервис-ориентированных систем. Интероперабельность. Слабое связывание. Сервисы, веб-сервисы. Автономность сервисов. Повторное использование. Формальный контракт использования сервисов. Состояние и ресурсы сервиса.

3. Протоколы распределенного взаимодействия при построении СОА-приложений. Протокол SOAP.

4. Протоколы распределенного взаимодействия при построении СОА-приложений. Протокол (язык) JavaScript Object Notation (JSON).

5. Описание и распознавание веб-сервисов. Язык описания веб-сервисов WSDL.

6. Разработка .asmx веб-сервисов с использованием Visual Studio. Тестирование .asmx сервисов. Прокси-классы. Применение веб-сервисов.

7. Безопасность .asmx сервисов. Стандарт WS.Security. Использование заголовков SOAP.

8. Основы технологии WCF. Адрес (оконечная точка), привязка и контракт. Контракт о службе. Контракт о данных. Контракт о сообщениях.

9. Стандартные привязки в WCF.

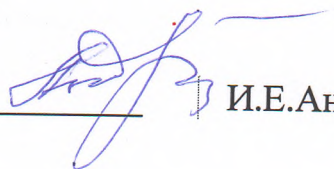
10. Безопасность WCF-служб.

11. REST сервисы. Ограничения архитектурного стиля REST. Безопасность REST сервисов. Создание REST сервисов в WCF.
12. ASP.NET Web API как платформа для сборки REST-приложений на базе .NET.
13. Разработка ASP.NET Web API служб. Использование служб Web API.
14. Авторизация с помощью JWT в Web Api
15. RPC. gRPC сервисы.
16. gRPC для WCF-разработчиков.
17. Понятие микросервиса и его основные отличительные признаки.
18. Технологии и инструменты, связанные с применением микросервисов. Контейнеры, Docker и Kubernetes.
19. Докеры и виртуальные машины.
20. Взаимодействие микросервисов.
21. Микросервисы и данные.
22. Микросервисы и распределенные транзакции. Двухфазная фиксация (two-phase commit protocol, 2PC). Согласованность в конечном счете (Saga pattern).
23. Управление микрослужбами и многоконтейнерными приложениями.
24. Шаблоны (принципы, парадигмы) (CQS) и CQRS.
25. Алгоритмы балансировки нагрузки в высоконагруженных распределенных системах.
26. Сетка сервисов (Service Mesh).

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УО

Название учебной дисциплины, с которой требуется согласование	Название Кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Сетевые приложения в среде Linux	Кафедра многопроцессорных систем и сетей	Предложения отсутствуют	Рекомендовать к утверждению учебную программу (протокол № 17 от 15.05.2025)

Заведующий кафедрой многопроцессорных систем и сетей
кандидат математических наук, доцент


 И.Е.Андрушкевич

15.05.2025

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УО
на ____/____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры
_____ (протокол № ____ от _____ 202__ г.)

Заведующий кафедрой

УТВЕРЖДАЮ

Декан факультета
