

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

О.Г. Прохоренко

«05» июля 2023 г.

Регистрационный № УД – 12893/уч.



РАЗРАБОТКА И ЗАЩИТА СЕРВИС-ОРИЕНТИРОВАННЫХ СИСТЕМ

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:**

1-98 01 01 Компьютерная безопасность (по направлениям)

направления специальности

**1-98 01 01– 01 Компьютерная безопасность
(математические методы и программные системы)**

2023 г.

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-98 01 01-2021, типового учебного плана №Р 98-1-003/пр-тип. от 02.07.2021г., учебных планов БГУ: № Р 98-1-005/уч. от 23.07.2021г., №Р98-1-024/уч. ин. от 09.08.2021 г.

СОСТАВИТЕЛЬ:

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

РЕЦЕНЗЕНТ

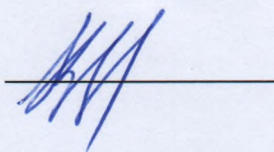
Д. В. Шункевич, заведующий кафедрой интеллектуальных информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой технологий программирования факультета прикладной математики и информатики БГУ
(протокол № 16 от 18 мая 2023 г.)

Научно-методическим советом БГУ
(протокол № 9 от 29.06.2023)

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 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; контейнеризацией монолитных .NET 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 специализации** компонента учреждения высшего 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 «Разработка и защита сервис-ориентированных систем» способствует успешному прохождению практики и написанию дипломных работ.

Требования к компетенциям

Освоение учебной дисциплины «Разработка и защита сервис-ориентированных систем» должно обеспечить формирование следующих компетенций:

универсальные компетенции:

- УК-1. Владеть основами исследовательской деятельности, осуществлять поиск, анализ и синтез информации;
- УК-2. Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий;
- УК-4. Работать в команде, толерантно воспринимать социальные, этнические, конфессиональные, культурные и иные различия;
- УК-5. Быть способным к саморазвитию и совершенствованию в профессиональной деятельности;

базовые профессиональные компетенции:

БПК-2. Строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного и иных парадигм программирования.

специализированные компетенции:

СК-7. Понимать принципы построения компьютерных систем и сетей, понимать и применять алгоритмы работы протоколов маршрутизации в IP-сетях, создавать сетевые приложения;

СК-11. Применять навыки проектирования и реализации систем безопасности, осуществлять выбор подходящего криптографического метода защиты типа данных и его реализации

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;

владеть:

- технологией разработки приложений сервис-ориентированной архитектуры с использованием возможностей, предоставляемых программной платформой.
- процессом разработки приложений Docker на основе контейнера;
- способами проверки подлинности с помощью ASP.NET Core Identity и внешних провайдеров аутентификации.

Структура учебной дисциплины.

Дисциплина изучается в шестом семестре. Всего на изучение учебной дисциплины отведено:

– для очной формы получения высшего образования – 108 часов, в том числе 64 аудиторных часа, из них: лекции - 32 часа, лабораторные занятия – 30 часов, аудиторная управляемая самостоятельная работа – 2 часа.

Трудоемкость учебной дисциплины составляет 3 зачетных единицы.

Форма промежуточной аттестации – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. 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. Сервисы, веб-сервисы. Формальный контракт использования сервисов. Состояние и ресурсы сервиса.

Тема 1.2. Протоколы распределенного взаимодействия при построении СОА-приложений.

Протокол SOAP. Структура SOAP-сообщения. Заголовок SOAP и его стандартные атрибуты. Обработка ошибок в SOAP-сообщениях. Использование в различных платформах (ASP.NET, Java).

Протокол (язык) Java Script Object Notation (JSON). Общие сведения, предназначение. Универсальные структуры данных языков программирования (коллекция пар ключ/значение, массивы и др.) и их реализация в JSON. Строки, числа в JSON. Сравнение протоколов SOAP и JSON.

Тема 1.3. .asmx веб-сервисы.

Язык описания веб-сервисов WSDL. Основные элементы WSDL 2.0 файла. Разработка .asmx сервисов. Тестирование .asmx сервисов. Прокси-классы. Применение .asmx веб-сервисов.

Тема 1.4. 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 1.5. Веб-службы Windows Communication Foundation (WCF).

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. REST API и gRPC.

Тема 2.1. Веб-службы Web 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.

Тема 2.2. 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. Микросервисная архитектура.

Тема 3.1. Введение в микросервисы.

Понятие об облачных приложениях. DevOps.

Технологии и инструменты, связанные с применением микросервисов: контейнеры, Docker, Kubernetes, шлюзы API, обмен сообщениями и потоки событий, бессерверные архитектуры.

Тема 3.2. 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 3.3. Взаимодействие микросервисов.

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

Тема 3.4. Контейнеризация Docker.

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

Тема 3.5. 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 3.6. Управление микрослужбами и многоконтейнерными приложениями.

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

Тема 3.7. Обеспечение безопасности микрослужб и веб-приложений .NET.

Проверка подлинности с помощью ASP.NET Core Identity. Использование внешних провайдеров аутентификации. Проверка подлинности с помощью поставщика удостоверений OpenID Connect или OAuth 2.0. Выпуск токенов безопасности службой ASP.NET Core. Использование токенов JWT.

Реализация авторизации на основе ролей. Реализация авторизации на основе политик.

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.	Веб-сервисы первого и второго поколений	10			2			
1.1.	Принципы проектирования и построения сервис-ориентированных систем	2						Устный опрос
1.2.	Протоколы распределенного взаимодействия при построении СОА-приложений	2						Устный опрос
1.3.	.asmx веб-сервисы.	2						Устный опрос
1.4.	Стандарты второго поколения веб-сервисов	2						Устный опрос
1.5.	Веб-службы Windows Communication Foundation (WCF).	2			2			Отчет по лабораторной работе с устной защитой
2.	REST API и gRPC	6			10		2	
2.1.	Веб-службы Web API.	2			4			Отчет по лабораторной работе с устной защитой
2.2.	gRPC.	2			4			Отчет по лабораторной работе с устной защитой
2.3.	Вопросы безопасности при использовании REST API и gRPC.	2			2		2	Отчет по лабораторной работе с устной защитой. Проверка программного кода приложения
3.	Микросервисная архитектура	16			18			
3.1	Введение в микросервисы.	2			2			Отчет по

								лабораторной работе с устной защитой
3.2	Микросервисы и данные.	2			2			Отчет по лабораторной работе с устной защитой
3.3	Взаимодействие микросервисов.	2						Устный опрос
3.4	Контейнеризация Docker.	2			2			Отчет по лабораторной работе с устной защитой
3.5	Процесс разработки приложений на основе Docker.	2			4			Отчет по лабораторной работе с устной защитой
3.6	Управление микрослужбами и многоконтейнерными приложениями.	2			4			Отчет по лабораторной работе с устной защитой
3.7	Обеспечение безопасности микрослужб и веб- приложений .NET.	4			4			Отчет по лабораторной работе с устной защитой
	ИТОГО	32			30		2	

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

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/dockerlifecyleebook>
3. Ньюмен, С. Создание микросервисов / С. Ньюмен, – М.: Прогресс книга, 2023. – 624 с.
4. Ньюмен, С. От монолита к микросервисам / С. Ньюмен, – BHV-СПб, 2021. – 272 с.
5. Беллемар, Адам. Создание событийно-управляемых микросервисов / Адам Беллемар. – БХВ-Петербург, 2023. – 320 с.
6. Индрасири К., Курупу Д. gRPC: запуск и эксплуатация облачных приложений. Go и Java для Docker и Kubernetes / К. Индрасири, Д. Курупу. – СПб.: Питер, 2022. – 224 с.
7. Дергачев А.М., Кореньков Ю.Д., Логинов И.П., Сафронов А.Г., Технологии веб-сервисов– СПб: Университет ИТМО, 2021. – 100 с.
8. Руководство по ASP.NET 8 Core [Электрон. ресурс]. – <https://metanit.com/sharp/aspnet6/>
9. Rendle, Mark. gRPC for WCF Developers / Mark Rendle, Miranda Steiner. – Microsoft Corporation, 2023. – 104 с.

Перечень дополнительной литературы

1. Обновление серверного проекта WCF для использования CoreWCF в .NET 6 [Электрон. ресурс]. – <https://learn.microsoft.com/ru-ru/dotnet/core/porting/upgrade-assistant-wcf>
2. What to do with WCF in .NET 8 and Beyond [Электрон. ресурс]. – <https://blog.inedo.com/dotnet/windows-comm-framework/>
3. Радченко, Г.И. Распределенные вычислительные системы / Г.И. Радченко. – Челябинск: Фотохудожник, 2012. – 184 с.
4. Лёве, Д. Создание служб WCF. / Д. Лёве.– СПб.: Питер, 2008. - 592 с.
5. SOAP 1.2specification (W3C) [Электрон. ресурс].- <http://www.w3.org/TR/soap12-part1/>
6. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language [Электрон. ресурс]. – www.w3.org/TR/wsdl
7. Web Services Description Language (WSDL) Version 2.0 Part 0: Primer [Электрон. ресурс]. – <http://www.w3.org/TR/wsdl20-primer/>
8. WS-Security specification (OASIS) [Электрон. ресурс]. – <http://www.oasis-open.org/specs/index.php#wssv1.0>

9. Bertino, Elisa. Security for Web Services and Service-Oriented Architectures. / Elisa Bertino, Lorenzo D. Martino, Federica Paci, Anna C. Squicciarini. – Berlin, Heidelberg: Springer-Verlag, 2010. – 222 с.
10. REST: From Research to Practice /E. Wilde, C. Pautasso (ред.) // Springer Science+Business Media, LLC, 2011.
11. Allamaraju, Subbu. RESTful Web Services Cookbook / Subbu Allamaraju. – СПб.: O'Reilly, 2010. – 296 с.
12. Kocher, Parminder Singh. Microservices and Containers / Parminder Singh Kocher. – Addison-Wesley Professional, 2018. – 304с.
13. Введение в API [Электрон. ресурс]. – <https://systems.education/api-intro>

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 «Разработка и защита сервис-ориентированных систем» учебным планом предусмотрен **зачет**.

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

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 2.3. Вопросы безопасности при использовании REST API и gRPC. (2 ч)

Задание. Обеспечить защиту разработанного ASP.NET Core приложения. Форма контроля – проверка программного кода приложения.

Примерная тематика лабораторных занятий

Лабораторная работа № 1. Веб-службы Web API.

Лабораторная работа № 2. Веб-службы gRPC, разработка и использование.

Лабораторная работа № 3. Обеспечение безопасности приложений при использовании веб-служб Web API и gRPC.

Лабораторная работа № 4. Групповой проект: разработка распределенной системы с использованием веб-служб Web API и gRPC.

Лабораторная работа № 5. Изучить, проанализировать, запустить на выполнение и продемонстрировать работу интернет-магазина в виде контейнеризованного приложения на основе существующего проекта eShopOnContainers.

Лабораторная работа № 6. Проектирование и реализация приложения с микросервисной архитектурой.

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 зачету

1. Принципы проектирования и построения сервис-ориентированных систем.
2. Протокол SOAP. Структура SOAP-сообщения.
3. Язык описания веб-сервисов WSDL.
4. .asmx веб-сервисы.
5. Стандарты второго поколения веб-сервисов. WS.Security.
6. Стандарты второго поколения веб-сервисов. Стандарт WS-Addressing.
7. Веб-службы Windows Communication Foundation (WCF). Переход с WCF на CoreWCF, Web API, gRPC.
8. Ограничения архитектурного стиля REST. REST сервисы.
9. Веб-службы Web API.
10. Технология RPC разработки распределенных систем
10. gRPC сервисы.
11. Сравнение сфер применения, преимуществ и слабых сторон gRPC и Web API.
12. Опасные уязвимости REST API (на основании списков OWASP Top 10).
13. JWT-токены и их использование при аутентификации.
14. Использование защиты данных ALTS в gRPC.
15. Аутентификация на основе токенов при использовании gRPC.
16. Понятие об облачных приложениях.
17. Технологии и инструменты, связанные с применением микросервисов.
18. Микросервисы и данные.
19. Взаимодействие микросервисов.
20. Контейнеризация Docker.
21. Процесс разработки приложений на основе Docker.
22. Управление микрослужбами и многоконтейнерными приложениями.
23. Обеспечение безопасности микрослужб.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

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)
Технология ASP.NET разработки и защиты серверных Web-приложений	Технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения, (протокол № 16 от 18 мая 2023 г.)

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО
ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

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

Заведующий кафедрой
технологий программирования _____

УТВЕРЖДАЮ
Декан ФПМИ
