

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Ректор Белорусского
государственного университета

А.Д.Король

15 июля 2024 г.

Регистрационный №УД- 13287/уч.



ВЕБ-СЕРВИСЫ

Учебная программа учреждения образования по учебной дисциплине для
специальности:

**1-31 03 08 Математика и информационные технологии
(по направлениям)**

Направления специальности:

1-31 03 08-01 Веб-программирование и интернет-технологии

2024 г.

Учебная программа составлена на основе ОСВО 1-31 03 08-2021 и учебных планов №G31-1-011/уч. от 25.05.2021, №G31-1-003/уч.ин. от 31.05.2021, №G31-1-004/уч.з от 31.05.2021, №G31-1-220/уч. от 22.03.2022, №G31-1-225 уч.ин. от 27.05.2022, №G31-1-218/уч.з. от 27.05.2022.

СОСТАВИТЕЛЬ:

Денис Николаевич Зинченко, старший преподаватель кафедры веб-технологий и компьютерного моделирования механико-математического факультета Белорусского государственного университета.

РЕЦЕНЗЕНТЫ:

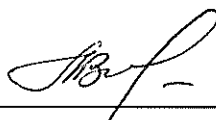
П.В. Гляков, профессор кафедры информационных технологий в культуре Учреждения образования «Белорусский государственный университет культуры и искусств», кандидат физико-математических наук.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой веб-технологий и компьютерного моделирования БГУ
(протокол № 19 от 05.06.2024);

Научно-методическим советом БГУ
(протокол № 9 от 28.06.2024)

Заведующий кафедрой



М.В.Игнатенко

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа по учебной дисциплине «Веб-сервисы» относится к дисциплинам специализации и разработана для студентов 4 курса очной формы получения высшего образования, для студентов 4 курса заочной формы обучения получения высшего образования обучающихся по специальности **1-31 03 08 Математика и информационные технологии (по направлениям)** направления специальности 1-31 03 08-01 Математика и информационные технологии (веб-программирование и интернет-технологии).

Цели и задачи учебной дисциплины

Цель учебной дисциплины – освоение учащимися современных технологий и методологий разработки сервисного ПО, основанных на использовании серверных языков программирования, облачных технологий и прикладных протоколов и развития ими навыков проектирования и разработки веб-сервисов.

Задачи учебной дисциплины:

1. Получение знаний об основных принципах и концепциях, связанных с веб-сервисами.
2. Углубление знаний о концепциях, положенных в основу проектирования и разработки веб-сервисов.
3. Формирование знаний, необходимых для изучения методов защиты веб-сервисов и передаваемых данных.
4. Получение знаний о способах интеграции веб-сервисов с различными системами и приложениями.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к дисциплинам специализации компонента учреждения высшего образования.

Связи с другими учебными дисциплинами, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и др.

Учебная дисциплина «Веб-сервисы» опирается на учебные дисциплины «Технологии программирования», «Веб-программирование», «Разработка веб-приложений».

Изучение дисциплины «Веб-сервисы» и выполнение соответствующих задач должны способствовать формированию у студентов базовых навыков анализа и применения современных методов для решения научных и прикладных задач.

Требования к компетенциям

Освоение учебной дисциплины «Веб-сервисы» должно обеспечить формирование следующих универсальных, базовых профессиональных и специализированных компетенций:

универсальные компетенции:

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

базовые профессиональные компетенции:

БПК. Применить современные технологии и базовые конструкции языков программирования для реализации алгоритмических прикладных задач и разработки веб-проектов.

специализированные компетенции:

СК. Осуществлять анализ контекста и поставленной проблемы, аргументированно выбирать оптимальный способ ее решения, согласовывать частичные проекты решения в общую согласованную архитектуру, выполнять реализацию проекта с учетом оценки накопленных и поступающих данных.

СК. Разрабатывать информационные решения для поиска и извлечения данных, ставить и тестировать прикладные гипотезы на основе данных, разрабатывать специализированные модели машинного обучения и алгоритмы анализа данных, развертывать модели в том числе в облачной среде, осуществлять визуализацию и бизнес-анализ полученных решений.

В результате освоения учебной дисциплины студент должен:

знать:

- основные протоколы использующиеся в веб-разработке;
- иметь глубокие знания протоколов HTTP, RPC, SOAP;
- паттерны проектирования веб-сервисов и веб-приложений;
- основы работы с облачными технологиями: Yandex Cloud, Amazon Web Services, Microsoft Azure;
- основные методики разработки микросервисной архитектуры;
- способы построения бэкенд-архитектур;
- основные сквозные аспекты приложений;

уметь:

- реализовывать монолитные и микросервисные системы;
- выбрать адекватный подход для реализации конкретной задачи данной предметной области;
- построить и программно реализовать функционал приложения;
- реализовать серверную часть компоненты веб-сервиса, с помощью самостоятельно выбранного языка программирования и архитектурного подхода;
- протестировать и отладить веб-сервисы для обеспечения их корректной работы;
- анализировать производительность реализуемой системы результатов и сравнивать эффективность различных методик, а также автоматизировать процесс тестирования и развертывания компонент веб-сервиса;

владеть:

- навыками работы с современными серверными языками объектно-ориентированного и функционального программирования;
- навыками построения распределенных систем;
- основными приемами реализации сквозных аспектов приложений: журналирование ошибок, обеспечение безопасности доступа к функциональности включая аутентификацию и авторизацию;

— навыками работы с инструментами и платформами для разработки веб-сервисов.

Структура учебной дисциплины

Дисциплина изучается в 7 для очной и в 8 семестре для заочной. Всего на изучение учебной дисциплины «Веб-сервисы» отведено:

— для очной формы получения высшего образования – 90 часов, в том числе 52 аудиторных часов, из них: лекции – 36 часов, лабораторные занятия – 16 часов, управляемая самостоятельная работа (УСР) – 2 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – экзамен.

— для заочной формы получения высшего образования – 42 часа, в том числе – 12 аудиторных часов, из них: лекции – 8 часов, лабораторные занятия – 4 часа.

Форма промежуточной аттестации – экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1 Введение в веб-сервисы.

Обзор современных технологий и методов построения веб-сервисов. Основные паттерны проектирования веб-ориентированного ПО. Обзор архитектурных стилей, протоколов и стандартов веб-сервисов. Обзор монолитного и микросервисного подходов.

Тема 2 Рассмотрение SOAP протокола при построении веб-сервисной архитектуры.

Протоколы SOAP, RPC, TCP.

Адрес, связывание контракт. Типы контрактов, способы связывания адресов и контрактов. Контракты данных, контракты сервисов.

Язык XML и структура SOAP-сообщений, обзор языка описания веб-сервисов (WSDL), сериализация и десериализация объектов, протоколы транспортного и прикладного уровней и стандарты, коды состояний.

Описание интерфейсов и сервисов, атрибуты Data Contract, Service Contract.

Производительность. Журналирование ошибок, обработка исключений, сессия, куки.

Тема 3 REST архитектура при построении веб-сервисов.

Statefull, stateless подходы. Основные принципы и паттерны при реализации RESTful.

Правила построения URL, именование адресов и методов. Виды Token, аутентификация, авторизация, регистрация. Авторизация формами, авторизация токенами, OAuth 2.0, Single sign-on, инвалидация кэша и токенов.

Форматы данных: JSON, XML.

Тема 4 Протоколы прикладного уровня. HTTP 1.1, HTTP/2, HTTP/3.

HTTP/3 протокол на базе QUIC, использование UDP.

Мультиплексирование. Приоритеты. Сжатие HTTP-заголовков. Базовая настройка HTTP/2 в Nginx и Apache. Оптимизация сайтов. Встраивание изображений с помощью DataURI. Конкатенация JS и CSS. Доменное шардирование.

Коды состояний. Обработка ошибок. Заголовки запроса, тело запроса. Заголовки безопасности. Шифрование трафика. Типы передаваемых данных.

Cross Origin Site Scripting. Идемпотентные и неидемпотентные запросы, кроссдоменные запросы.

Тема 5 Способы построения бэкенд архитектуры в разработке веб-сервисов. Монолитная архитектура.

Use-Case Oriented Design, многослойная архитектура.

Внедрение зависимостей, инверсия управления, инверсия зависимостей, контейнер управления зависимостями. Время жизни объектов, стратегии создания объектов. Реализация поведения и моделей.

Уровни абстракций приложения, делегирование обязанностей.

Тема 6 Микросервисы. Сервис-ориентированная архитектура.

Свойства архитектуры. Разбиение через сервисы. Smart endpoints, dump pipes. Децентрализованное управление. API-ресурсы. Сервис-ориентированная архитектура.

Автоматизация инфраструктуры, Docker, Kubernetes. Проектирование под отказ. Синхронная и асинхронная коммуникация вместе с gRPC, RabbitMQ, Apache Kafka.

Bounded Context. Независимая разработка. Масштабируемость. Повторное использование.

Развертывание. Continuous Integration, Continuous Delivery.

Тема 7 Event Sourcing.

Событийно-ориентированный подход. Проекции. UI ориентированный на процесс. Event Store. Взаимно компенсирующие события. CRUD операции. Clients. Command Handlers.

Использование снапшотов состояния для повышения производительности. Transaction log tailing pattern. Брокер сообщений Apache Kafka. Журналирование, модули.

Тема 8 Обзор WebSocket. GraphQL.

Коммуникационный протокол. Клиент-серверная модель. Полнодуплексная связь.

Обработка сообщений и событий. Обработка ошибок. Обновления в реальном времени. Оптимизация и тестирование производительности приложений с использованием WebSocket.

GraphQL. Apollo Client. Кэширование. Ленивые запросы. Оптимизация запросов.

Тема 9 Облачные технологии построения веб-сервисов.

SAAS, IAAS, PAAS. Servless архитектура, Yandex Cloud, Azure Functions, Amazon Lambda. Гибридные облачные приложения.

Azure Pipelines. App Service. Container Service. Контейнеризация, Kubernetes Service Fabric. Load Balancer. Log Analytics. Виртуальные сети. Оркестрация контейнеризации.

Yandex Cloud Functions, Network Load Balancer, Virtual Private Cloud, Servless Integrations.

Azure vs. Yandex Cloud vs. AWS. Сервера. Хранилища данных. API Gateway. Децентрализация данных. Стратегия кэширования Redis.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

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

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Введение в веб-сервисы.	2						Опрос
2	Рассмотрение SOAP протокола при построении веб-сервисной архитектуры.	4			2			Защита лабораторной работы, опрос
3	REST архитектура при построении веб-сервисов.	6			2		2	Защита лабораторной работы, опрос. Опрос по результатам выполнения заданий УСР.
4	Протоколы прикладного уровня. HTTP 1.1, HTTP/2, HTTP/3.	4			2			Защита лабораторной работы, опрос,
5	Способы построения бэкенд архитектуры в разработке веб-сервисов. Монолитная архитектура.	4			2			Защита лабораторной работы, опрос

6	Микросервисы. Сервис-ориентированная архитектура.	6				2		Защита лабораторной работы, опрос
7	Event Sourcing.	2				2		Защита лабораторной работы, опрос
8	Обзор WebSocket. GraphQL.	4				2		Защита лабораторной работы, опрос
9	Облачные технологии построения веб-сервисов.	4				2		Защита лабораторной работы, опрос
	ВСЕГО ЧАСОВ	36				16	2	

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Заочная форма получения высшего образования

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Форма контроля
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное	
1	2	3	4	5	6	7	8
1	Введение в веб-сервисы.	1					Опрос
2	Рассмотрение SOAP протокола при построении веб-сервисной архитектуры.	1					Опрос
3	REST архитектура при построении веб-сервисов.	1			1		Защита лабораторной работы, опрос
4	Протоколы прикладного уровня. HTTP 1.1, HTTP/2, HTTP/3.	1					Опрос
5	Способы построения бэкенд архитектуры в разработке веб-сервисов. Монолитная архитектура.	1					Опрос
6	Микросервисы. Сервис-ориентированная архитектура.	1			2		Защита лабораторной работы, опрос
7	Event Sourcing.	1					Опрос
8	Обзор WebSocket. GraphQL.	1					Опрос
9	Облачные технологии построения веб-сервисов.				1		Опрос
	ВСЕГО ЧАСОВ	8			4		

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Ньюмен, С. Создание микросервисов. 2-е издание. / Сэм Ньюмен — СПб.: «Питер». 2024.
2. Дергачев, А. Технологии веб-сервисов. / А. Дергачев, Ю. Кореньков, И. Логинов, А. Сафронов — СПб.: Университет ИТМО. 2021.
3. *Newman, S. Building Microservices Designing Fine-Grained Systems.* / Sam Newman — O'Reilly Media, 2015.
4. Ньюкомер, Э. Веб-сервисы. XML, WSDL, SOAP и UDDI. Для профессионалов. / Эрик Ньюкомер — СПб.: «Питер». 2003.
5. *Gourley, G. HTTP: The Definitive Guide* / David Gourley, Brian Totty, Marjorie Sayer, Anshu Aggarwal, Sailu Reddy — O'Reilly Media, 2002.
6. *Richardson, L. RESTful Web APIs* / Leonard Richardson, Mike Amudsen, Sam Ruby — O'Reilly Media, 2013.

Дополнительная литература

1. Хорсдал, К. Микросервисы на платформе .NET / К. Хорсдал. — СПб.: «Питер». 2018.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Диагностика результатов учебной деятельности по учебной дисциплине проводится преподавателем, как правило, во время аудиторных занятий. Для диагностики используются:

- устный опрос;
- защита лабораторных работ;
- опрос по результатам выполнения заданий УСР.

Формой промежуточной аттестации по дисциплине «Веб-сервисы» учебным планом предусмотрен экзамен.

Для формирования итоговой отметки по учебной дисциплине используется модульно-рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущей и промежуточной аттестации студентов по учебной дисциплине.

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

- защита лабораторных работ – 75%;

- управляемая самостоятельная работа (УСР) – 25%;

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

Примерный перечень заданий для управляемой самостоятельной работы

Тема 3. REST архитектура при построении веб-сервисов. (2ч.)

Работа с данными в формате JSON, обработка ошибок, реализация пагинации, поиск сущностей, поддержка CORS.

(Форма контроля — опрос по результатам выполнения заданий УСР).

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используются:

— *методы и приемы развития критического мышления*, которые представляют собой систему, формирующую навыки работы с информацией в процессе чтения и письма; понимании информации как отправного, а не конечного пункта критического мышления;

— *практико-ориентированный подход*, который предполагает освоение содержание образования через решения практических задач.

Методические рекомендации по организации самостоятельной работы

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

— изучение основной и дополнительной литературы по учебной дисциплине;

— изучение материалов электронных источников по вопросам учебной дисциплины, в том числе размещенных в электронной библиотеке БГУ elib.bsu.by;

— подготовка к лабораторным занятиям;

— выполнение домашних заданий и лабораторных работ.

Для организации самостоятельной работы студентов по учебной дисциплине рекомендуется использовать возможности образовательного портала edummf.bsu.by и системы управления обучения Moodle для размещения текущей организационной информации и учебно-программных материалов, в том числе вопросов к экзаменам/зачетам, а также контроля выполнения заданий.

Основной формой УСР с учетом специфики дисциплины является самостоятельное под методическим управлением преподавателя выполнение студентами учебных (исследовательских) заданий. Количество часов на УСР указано в учебно-методической карте. Конкретная форма проведения и

контроля УСР выбирается преподавателем. В качестве основного вида контроля выполнения УСР предлагается проверка выполнения учебных заданий. Кроме того, знание материала, рассматриваемого студентами в рамках УСР, проверяется в числе прочих на текущей аттестации (экзамен или зачет).

Примерный перечень вопросов к экзамену

1. Протоколы SOAP, RPC, TCP.
2. Адрес, связывание, контракт.
3. Типы контрактов, способы связывания адресов и контрактов.
4. Контракты данных, контракты сервисов.
5. Язык XML, язык описания веб-сервисов WSDL.
6. Протоколы транспортного и прикладного уровней.
7. Описание интерфейсов и сервисов, атрибуты Data Contract, Service Contract.
8. Stateless, statefull подходы.
9. Правила построения URL, именованье адресов и методов в REST.
10. Виды Token в REST.
11. Регистрация, авторизация, аутентификация.
12. Авторизация формами, авторизация токенами.
13. OAuth и OAuth2.
14. Single sign-on.
15. Инвалидация кэша и токенов.
16. Форматы JSON, XML в REST.
17. Различия между REST и RESTful.
18. Коды состояний HTTP.
19. Обработка ошибок HTTP.
20. Заголовки запроса, тело запроса. Заголовки безопасности.
21. Cross Origin Site Scripting.
22. Идемпотентные и неидемпотентные запросы.
23. Кроссдоменные запросы.
24. Use-Case Oriented Design, многослойная архитектура.
25. Уровни абстракций приложения, делегирование обязанностей.
26. Микросервисы. Свойства архитектуры. Разбиение через сервисы.
27. Smart endpoints and dumb pipes в микросервисной архитектуре.
28. Децентрализация управления в микросервисной архитектуре.
29. Service Oriented Architecture.
30. Bounded Context. Масштабируемость. Повторное использование.
31. Развертывание. Continuous Integration, Delivery Integration.
32. Event Sourcing. Событийно-ориентированный подход.
33. Event Store. Использование снапшотов состояния для повышения производительности.
34. Облачные технологии. Актуальные сервисы в Azure, Amazon, Yandex Cloud.
35. Облачные технологии. Контейнеризация, Kubernetes Service Fabric.

36. Облачные технологии. Хранилища данных. API Gateway. Децентрализация данных. Стратегии кэширования Redis.

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)
Учебная дисциплина не требует согласования			

Заведующий кафедрой
к. физ.-мат. н., доцент



М.В. Игнатенко

05. июня 2024г.

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
 _____ (протокол № ____ от _____ 202_ г.)
 (название кафедры)

Заведующий кафедрой

 (ученая степень, ученое звание)

 (И.О.Фамилия)

УТВЕРЖДАЮ

Декан факультета

 (ученая степень, ученое звание)

 (И.О.Фамилия)