Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шутов Олег Леонтьевич
Должность: Ректор
Дата подписания: 16.11.2026 12:01:53
Уникальный программный ключ:
6892313c2153d214b87fca0fd68c13fa12d41989

Разработка программно-информационных систем
09.03.04 Программная инженерия
2024 год набора
Приложение В
к основной профессиональной образовательной программе по направлению подготовки 09.03.04 Программная инженерия, утвержденной приказом от 28.08.2024 г. № 60-О
(в редакции приказа от 02.11.2026 г. № 87-О)

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КУБАНСКИЙ ИНСТИТУТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ»
(АНОО ВО «КИПО»)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.18 Введение в разработку веб-сайтов

Направление подготовки

09.03.04 Программная инженерия

Направленность (профиль)

Разработка программно-информационных систем

Уровень высшего образования
Бакалавриат

Квалификация

Бакалавр

Форма обучения
заочная

Год набора
2024

Рабочая программа дисциплины (модуля) Б1.О.18 «Введение в разработку веб-сайтов» предназначена для реализации основной профессиональной образовательной программы высшего образования.

Составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.04 Программная инженерия (Приказ Минобрнауки России от 19.09.2017 г. № 920 зарегистрирован в Минюсте России от 16.10.2017 г. № 48546).

СОДЕРЖАНИЕ

1 Цели и задачи изучения дисциплины(модуля)
1.1 Цель освоения дисциплины (модуля)
1.2 Задачи дисциплины (модуля)
1.3 Место дисциплины (модуля) в структуре образовательной программы
1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы
2. Структура и содержание дисциплины (модуля)
2.1 Распределение трудоёмкости дисциплины (модуля) по видам работ
2.2 Содержание дисциплины (модуля)
2.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
3. Образовательные технологии, применяемые при освоении дисциплины (модуля)
4. Фонды оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)
4.1. Структура оценочных средств для текущего контроля и промежуточной аттестации по дисциплине (модулю)
4.2. Типовые задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации по дисциплине (модулю)
5. Методические указания для обучающихся по освоению дисциплины (модуля)
6. Материально-техническое и учебно-методическое обеспечение по дисциплине (модулю)

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Целью изучения дисциплины Б1.О.18 «Введение в разработку веб-сайтов» является овладеть основными принципами разработки проектов на основе веб-технологий на языке программирования Python.

1.2 Задачи дисциплины

-изучить архитектуру современных веб-приложений (модель MVC), принципы работы протокола HTTP, основные методы разработки клиентской и серверной частей; освоить современные фреймворки для веб-разработки на языке Python (Flask, Django), шаблонизаторы (Jinja2) и технологии объектно-реляционного отображения (SQLAlchemy); ознакомиться с содержанием Единого реестра российских программ для электронных вычислительных машин и баз данных в части, касающейся веб-технологий и инструментов разработки;

-научиться разрабатывать технические спецификации на веб-компоненты и их взаимодействие (включая REST API), проектировать структуру веб-приложения и базы данных, реализовывать серверную логику обработки HTTP-запросов, создавать динамические веб-страницы с использованием шаблонов, внедрять системы аутентификации и разграничения прав доступа, применять ORM для работы с базами данных, а также осуществлять отладку и тестирование разработанных веб-приложений;

-получить практический опыт разработки и публикации полноценного веб-приложения на языке Python с использованием веб-фреймворков (Flask/Django), включая проектирование архитектуры, интеграцию с базами данных, реализацию пользовательских интерфейсов на основе шаблонов, настройку маршрутизации, обеспечение безопасности (аутентификация, защита от типовых уязвимостей), а также опыт развертывания готового приложения на удаленной платформе (например, PythonAnywhere, Heroku) и документирования разработанных компонентов в соответствии с требованиями технической спецификации.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Б1.О.18 «Введение в разработку веб-сайтов» относится к части, формируемой участниками образовательных отношений Блока 1 «Дисциплины (модули)» учебного плана.

В соответствии с рабочим учебным планом дисциплина изучается на 4 курсе по заочной форме обучения.

Вид промежуточной аттестации: экзамен.

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенции	Код и наименование индикаторов достижения результатов обучения по дисциплине	Планируемые результаты обучения
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к	ОПК-6.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.	**Знать:** основные языки и нотации для описания функциональных спецификаций (UML, IDEF0); структуру и содержание технической спецификации на программный компонент (API, модуль, сервис); требования к

Код и наименование компетенции	Код и наименование индикаторов достижения результатов обучения по дисциплине	Планируемые результаты обучения
проектированию, конструированию и тестированию программных продуктов	ОПК-6.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ОПК-6.3. Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	оформлению документации на web-компоненты и их интерфейсы. **Уметь:** анализировать требования к web-приложению и формализовывать их в виде технических спецификаций; разрабатывать спецификации на программные интерфейсы (API) для взаимодействия компонентов web-приложения; выбирать технологии и инструментальные средства для реализации специфицированных компонентов. **Владеть:** методами разработки и документирования архитектуры web-приложений; навыками составления и согласования технических спецификаций на программные компоненты с архитектором и разработчиками; навыками формализации требований к взаимодействию клиентской и серверной частей web-приложения.

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 5 зачетных единиц (180 час.), их распределение по видам работ представлено в таблице

Виды работ		Всего часов		
		ОФО	ОЗФО	ЗФО
Контактная работа, в том числе:		**-**	**-**	**12**
Аудиторные занятия (всего):		**-**	**-**	**12**
занятия лекционного типа		-	-	6
практические занятия		-	-	6
Иная контактная работа:				
Контрольная работа		-	-	-
Курсовая работа		-	-	-
Самостоятельная работа, в том числе:		**-**	**-**	**159**
Самоподготовка по темам (разделам) дисциплины		-	-	109
Подготовка к текущему контролю		-	-	50
Контроль:		**-**	**-**	**9**
Промежуточная аттестация (экзамен)		-	-	9
Общая трудоемкость	**час.**	**-**	**-**	**180**
	в том числе контактная работа	**-**	**-**	**12**
	зач. ед	**-**	**-**	**5**

2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы (темы) дисциплины, изучаемые на 4 курсе (заочная форма обучения)

№	Наименование темы/раздела	Количество часов				
		Всего	В том числе в виде практической подготовки	Аудиторная работа		Внеаудиторная работа (СР)
				Л	ПЗ	
1.	**Основы программирования**. 1.1. Простые алгоритмы 1.2. Простые конструкции 1.3. Массивы 1.4. Функции 1.5. Двумерные массивы 1.6. Массивы и строки 1.7. Рекурсия 1.8. Классы 1.9. Паттерны программирования	11	-	1	-	10
2.	**Веб-разработка с применением концепции MVC.** 2.1 Методы Веб-разработки с применением концепции MVC	12	-	1	-	11
3.	**Основы использования шаблонов.** 3.1 Создание веб-сервера и обработчиков url (flask).	14	-	1	-	13
4.	**Генерация страниц (шаблонизатор jinja2).** 4.1 Bootstrap - стилевые компоненты, css.	15	-	-	-	15
5.	**Хранение и работа с данными.** 5.1 Основные принципы хранения и работы с данными	11	-	1	-	10
5.	**Отображение данных в БД на объекты приложения (ORM).** 5.1 Flask-sqlalchemy.	10	-	-	-	10
6.	**Использование форм.** 6.1 Авторизация. 6.2 Разграничение прав доступа пользователей (flask-login).	11	-	1	-	10
7.	**Редактирование шаблонов - добавление функционала для авторизованных пользователей.** 7.1 Методы редактирования шаблонов.	11	-	-	1	10
8.	**Размещение проекта в интернете (pythonanywhere или heroku).** 8.1 Принципы и критерии размещения проекта в интернете	11	-	1	-	10
9.	**Фреймворк Django.** 9.1 Работа с шаблонами и моделями.	11	-	-	1	10
10.	**Использование библиотек Django для создания блога**. 10.1 Сложные запросы к данным.	21	-	-	1	20
11.	**Расширенные возможности шаблонного механизма**. 11.1 Принципы и задачи расширенных возможностей шаблонного механизма.	11	-	-	1	10

12.	**Выдача данных в форматах, отличных от HTML.** 12.1 Средства отладки и тестирования	11	-	-	1	10
13.	**Развертывание веб-приложений.** 13.1 Методы развертывание веб-приложений.	11	-	-	1	10
	ИТОГО по разделам дисциплины	171	-	6	6	159
	Контрольная работа	-	-	-	-	-
	Курсовая работа	-	-	-	-	-
	Промежуточная аттестация (экзамен)	9	-	-	-	-
	Общая трудоемкость по дисциплине	180	-	6	6	159

Примечание: Л – лекции, ПЗ – практические занятия / семинары, СР – самостоятельная работа обучающегося

При изучении дисциплины могут применятся электронное обучение, дистанционные образовательные технологии в соответствии с ФГОС ВО.

2.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине)

Самостоятельная работа – это индивидуальная познавательная деятельность обучающегося как на аудиторных занятиях, так и во внеаудиторное время. Самостоятельная работа должна быть многогранной и иметь четко выраженную направленность на формирование конкретных компетенций.

Цель самостоятельной работы – овладение знаниями, профессиональными умениями и навыками, опытом исследовательской деятельности и обеспечение формирования профессиональных компетенций, воспитание потребности в самообразовании, ответственности и организованности, творческого подхода к решению проблем.

Самостоятельная работа обучающихся направлена на углубленное изучение разделов и тем рабочей программы. Самостоятельная работа предполагает изучение литературных источников, выполнение контрольных заданий и работ, проведение исследований разного характера. Работа основывается на анализе литературных источников и других материалов, а также реальных фактов, личных наблюдений и т.д.

Самостоятельная работа включает разнообразный комплекс видов и форм работы обучающихся:

- работа с лекционным материалом, предусматривающая проработку конспекта лекций и учебной литературы;
- поиск (подбор) и обзор литературы, электронных источников информации по заданной проблеме курса, написание реферата (доклада, эссе), исследовательской работы по заданной проблеме;
- выполнение задания по пропущенной или плохо усвоенной теме;
- изучение материала, вынесенного на самостоятельную проработку (отдельные темы, параграфы);
- подготовка к практическим занятиям;
- подготовка к промежуточной аттестации.

№ п/п	Вид учебно-методического обеспечения
1.	Методические рекомендации по самостоятельной работе обучающихся.
2.	Методические рекомендации по изучению дисциплины.
3.	Вопросы для письменного/устного собеседования, реферат, сообщение, доклад, эссе,

	практико-ориентированные задания, мини-кейсы, задания в виде расчетных задач, ситуационные задачи.

Задания для самостоятельной работы обучающихся по дисциплине Б1.О.18 «Введение в разработку веб-сайтов» представлены в учебно-методическом отделе.

Контроль результатов самостоятельной работы обучающихся может осуществляться в пределах времени, отведенного на обязательные учебные занятия и внеаудиторную самостоятельную работу обучающихся по дисциплине, может проходить в письменной, устной или смешанной форме.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) при изучении данной дисциплины предоставлена возможность выбора технологий обучения в зависимости от степени заболевания и осознания своей деятельности. При этом содержание программы дисциплины не изменяется, изменяются, как правило, форма обучения и образовательные технологии. Также обучающимся, имеющим инвалидность, и лицам с ограниченными возможностями здоровья созданы условия комфортного психологического климата в процессе обучения и возможности оказания помощи в установлении полноценных межличностных отношений с другими обучающимися.

3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

В ходе изучения дисциплины предусмотрено использование следующих образовательных технологий: лекции, практические занятия, проблемное обучение, модульная технология, подготовка письменных аналитических работ, самостоятельная работа обучающихся.

Компетентностный подход в рамках преподавания дисциплины реализуется в использовании интерактивных технологий и активных методов (проектных методик, мозгового штурма, разбора конкретных ситуаций, анализа педагогических задач, педагогического эксперимента, иных форм) в сочетании с внеаудиторной работой.

Информационные технологии, применяемые при изучении дисциплины: использование информационных ресурсов, доступных в информационно-телекоммуникационной сети Интернет.

Адаптивные образовательные технологии, применяемые при изучении дисциплины – для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

4. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины Б1.О.18 «Введение в разработку веб-сайтов». Материалы для проведения текущего контроля успеваемости и промежуточной аттестации размещены в фонде оценочных средств по дисциплине Б1.О.18 «Введение в разработку веб-сайтов».

4.1. Структура оценочных средств для текущего контроля и промежуточной аттестации по дисциплине (модулю)

№ п/п	Код и наименование индикаторов	Результаты обучения	Наименование оценочного средства	
			Текущий контроль	Промежуточная

	достижения результатов обучения по дисциплине			аттестация
1	ОПК-6.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.	**Знать:** основные языки и нотации для описания функциональных спецификаций (UML, IDEF0); структуру и содержание технической спецификации на программный компонент (API, модуль, сервис); требования к оформлению документации на web-компоненты и их интерфейсы. **Уметь:** анализировать требования к web-приложению и формализовывать их в виде технических спецификаций; разрабатывать спецификации на программные интерфейсы (API) для взаимодействия компонентов web-приложения; выбирать технологии и инструментальные средства для реализации специфицированных компонентов. **Владеть:** методами разработки и документирования архитектуры web-приложений; навыками составления и согласования технических спецификаций на программные компоненты с архитектором и разработчиками; навыками формализации требований к взаимодействию клиентской и серверной частей web-приложения.	Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на экзамене
2	ОПК-6.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.		Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на экзамене
3	ОПК-6.3. Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач		Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на экзамене

4.2. Типовые задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации по дисциплине (модулю)

Задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации, необходимые для оценки образовательных достижений обучающихся.

Текущий контроль успеваемости для обучающихся по очной форме

Вопрос для собеседования: Опишите структуру технической спецификации на REST API. Какие разделы являются обязательными и почему?

Тестовое задание: Какой элемент языка UML лучше всего подходит для описания взаимодействия компонентов "Клиент" и "Сервер" во времени?

А) Диаграмма классов

Б) Диаграмма последовательности

В) Диаграмма состояний

Г) Диаграмма компонентов

Задание на соответствие: Соотнесите термин из области проектирования web-приложений (1-4) с его правильным определением (А-Д).

OpenAPI Specification (Swagger)

Спецификация

Промежуточное ПО (Middleware)

Модель "Сущность-Связь" (ER-модель)

А) Детальное описание интерфейса, например, перечень всех endpoint'ов API с их параметрами и ответами.

Б) Программный код или компонент, который обрабатывает запросы до того, как они достигнут основного обработчика.

В) Формальный документ, точно описывающий требования, свойства и методы взаимодействия компонента.

Г) Язык графического описания структур данных, используемый при проектировании баз данных.

Д) Формат для описания RESTful API, который может быть использован для генерации документации и кода.

Практико-ориентированное задание (кейс):

Условие: Вам необходимо спроектировать микросервис "Управление пользователями" для web-приложения. На вход микросервис получает HTTP-запросы, а в ответ должен отдавать данные в формате JSON. Разработайте фрагмент технической спецификации (в свободной форме), которая включает:

Назначение и цели микросервиса.

Спецификацию минимум двух endpoint'ов:

`GET /users/{id}` — получение данных пользователя.

`POST /users` — создание нового пользователя.

Для каждого endpoint'а укажите: метод, URL, обязательные и необязательные параметры заголовка/запроса, пример тела запроса (если есть) и пример успешного ответа.

Задание на разработку спецификации компонента:

Опишите на естественном языке или в виде псевдокода спецификацию для программного компонента `UserAuthenticator`. Компонент должен реализовывать метод `authenticate(login, password)`. В спецификации необходимо указать: входные параметры (их типы, допустимые значения, ограничения), возвращаемое значение (например, `true/false` или токен сессии), а также описать, какие исключительные ситуации могут возникнуть (например, "Неверный пароль", "Пользователь заблокирован").

Аналитическое задание:

Проанализируйте предложенный фрагмент технической спецификации (см. Приложение 1 к билету). Найдите в нем не менее трех ошибок или неточностей (например, не указан формат даты, не описан код ошибки, отсутствует обязательный параметр). Предложите свой исправленный вариант спецификации.

Задание на согласование (в рамках командной работы):

Представьте, что вы — системный аналитик. Архитектор проекта изменил требования к модулю обработки платежей. Ваша задача: составить формализованное

описание изменений в технической спецификации для команды разработки (в виде уточненных требований к API, новым моделям данных или бизнес-логике). Зафиксируйте эти изменения в виде документа "Предложение об изменении спецификации версии 1.2".

Шкала оценивания результатов по заданиям для проведения текущего контроля успеваемости по дисциплине

% верных решений (ответов)	Шкала оценивания
85-100	5 - отлично
71-84	4 - хорошо
50-70	3 - удовлетворительно
0-49	2 - неудовлетворительно

Текущий контроль успеваемости для обучающихся по заочной форме

Контрольная работа представляет собой систематическое, достаточно полное изложение авторского решения соответствующей проблемы и выполнение заданий в рамках дисциплины, которая является одним из видов текущего контроля успеваемости обучающихся заочной формы обучения.

Цели контрольной работы:

- проверка и оценка знаний обучающихся;
- закрепление практических навыков применения теоретических подходов и методов анализа на учебных примерах и задачах;
- получение информации об уровне самостоятельности и активности обучающегося, об эффективности форм и методов учебной работы.

Контрольные работы выполняются обучающимися в сроки, предусмотренные учебным планом и календарным учебным графиком.

Контрольная работа выполняется в рукописном или в печатном (компьютерном) варианте на листах формата А4 в 1 экземпляре с соблюдением установленного формата. Текст набирается шрифтом Times New Roman 12, через 1 интервал, абзацный отступ - 1,25 см, выравнивание по ширине страницы. Страница должна иметь следующие поля: левое - 25 мм, правое - 10 мм, верхнее и нижнее - 20 мм. Титульный лист содержит информацию об обучающемся выполнившем контрольную работу (ФИО обучающегося, направление подготовки, группа); наименование дисциплины; ФИО преподавателя, проверяющего работу.

Задания для контрольных работ разрабатываются преподавателем дисциплины по вариантам, которые содержат:

1) Задание в форме ответа на теоретический вопрос по теме (разделу) – объем не более 2-3 страниц;
2) Задания, составленные в форме тестов (2 задания открытого и закрытого типа, разработанные в фонде оценочных средств) или ответ на практическое задание (решить ситуацию, математическую задачу, расчетно-графическое задание и др.).

Готовая контрольная работа в электронном виде прикрепляется в электронную образовательную среду Moodle в профиль обучающегося выполнившего работу до начала сессии. Если работа в рукописном варианте, то она должна быть отсканирована и прикреплена.

Шкала и критерии оценивания контрольной работы

№ п/п	Критерии	Зачтено
Теоретический вопрос		
1	Глубина проработки материала	Основные теоретические положения по вопросу раскрыты. Имеются элементы обоснования выводов

2	Представление	Имеются элементы систематизации информации, факты применения профессиональной терминологии
3	Использование рекомендованной литературы	Основные источники рекомендованной литературы использованы
4	Грамотность изложения и качество оформления	Продемонстрирована культура речи. Соблюдены основные требования к оформлению
Выполнение тестовых заданий		

Если работа не отвечает названным критериям, выставляется оценка «не зачтено».

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

Теоретические вопросы к экзамену

1. Клиент-серверная архитектура веб-приложений. Роли клиента и сервера, протокол HTTP, методы запросов (GET, POST, PUT, DELETE), коды ответов (200, 301, 404, 500). Приведите пример полного цикла обработки запроса браузера к серверу.

2. Понятия URL и DNS. Структура URL (схема, домен, порт, путь, параметры, фрагмент). Как доменное имя преобразуется в IP-адрес? Что такое HTTPS и в чём его отличие от HTTP?

3. Понятие жизненного цикла программного обеспечения. Место технической спецификации в процессах разработки (Waterfall, Agile).

4. Классификация требований к программному обеспечению (функциональные, нефункциональные). Как они отражаются в спецификации на web-компонент?

5. Роль API в современной web-разработке. Спецификация как контракт между клиентом и сервером.

6. Языки и нотации для формализации требований: IDEF0, BPMN, UML. Применение каждого для описания web-систем.

7. Структура и содержание технического задания (ТЗ) на разработку web-приложения.

8. OpenAPI Specification (Swagger): назначение, структура, преимущества использования для документирования REST API. Генерация клиентских библиотек по спецификации.

9. Спецификация на уровне базы данных: ER-диаграмма, описание таблиц, связей, индексов и триггеров.

10. Подходы к специфицированию пользовательского интерфейса (UI): создание макетов, wireframes, UI-китов. Чем UI-спецификация отличается от технической?

11. Спецификация на интеграцию с внешними сервисами (платежные шлюзы, соцсети). Какие данные она должна содержать?

12. Инструменты для совместной работы над спецификациями (SwaggerHub, Postman, Confluence). Системы контроля версий (Git) для хранения и управления изменениями спецификаций.

13. Спецификация и тестирование: как документ может служить основой для разработки тестовых сценариев (модульное, интеграционное, API-тестирование).

14. Обработка ошибок в web-приложении: как спецификация должна описывать возможные коды ошибок и ответов HTTP.

15. Нефункциональные требования в спецификации: производительность (время ответа API), безопасность (аутентификация/авторизация, требования к паролям), масштабируемость.

16. Версионирование API. Как спецификация помогает управлять API версиями и обеспечивать обратную совместимость.

17. Спецификация на событийно-ориентированную архитектуру: описание структуры событий, очередей и обработчиков (на примере web-сокетов или message broker).

18. Архитектура современных веб-приложений. Модель MVC (Model-View-Controller): компоненты и их взаимодействие.

19. Язык Python в веб-разработке: основные особенности, преимущества и недостатки. Обзор экосистемы фреймворков.

20. Веб-серверы и WSGI. Принципы работы WSGI-сервера. Роль Gunicorn/uWSGI в развертывании Python-приложений.

21. Фреймворк Flask: философия, установка, структура минимального приложения. Понятие "микрофреймворк".

22. Маршрутизация во Flask. Декораторы @app.route, переменные правила, обработка HTTP-методов.

23. Объект request во Flask. Работа с GET и POST параметрами, заголовками, файлами.

24. Шаблонизатор Jinja2: синтаксис, наследование шаблонов (extends, block), управляющие конструкции и циклы.

25. Работа с формами во Flask. Использование Flask-WTF для валидации и защиты от CSRF.

26. Управление сессиями и cookies во Flask. Функции session, flash, настройка секретного ключа.

27. Аутентификация и авторизация. Библиотека Flask-Login. Разграничение прав доступа пользователей.

28. Введение в базы данных. Реляционная модель. Основы SQL (DDL, DML). СУБД MySQL, PostgreSQL, SQLite.

29. Концепция ORM (Object-Relational Mapping). Преимущества и недостатки. Библиотека SQLAlchemy.

30. Flask-SQLAlchemy: определение моделей, типы данных, миграции базы данных (Alembic).

31. CRUD-операции в Flask-SQLAlchemy: создание, чтение, обновление и удаление записей.

32. Связи между таблицами в ORM (один-к-одному, один-ко-многим, многие-ко-многим). Реализация в SQLAlchemy.

33. Фреймворк Django: философия, структура проекта и приложений. Паттерн MVT (Model-View-Template).

34. Django ORM. Определение моделей, выполнение сложных запросов, агрегация и аннотация данных.

35. Django Admin: назначение, настройка, регистрация моделей.

36. Формы в Django. Классы Form и ModelForm, валидация данных.

37. Безопасность веб-приложений на Python. Основные уязвимости (XSS, CSRF, SQL-инъекции) и методы защиты.

38. Генерация данных в форматах, отличных от HTML (JSON, XML, CSV) в Django и Flask.

39. Тестирование и отладка веб-приложений на Python. Модульное тестирование (unittest, pytest), отладочный режим Flask/Django.

40. Развертывание веб-приложений. Размещение проекта на PythonAnywhere или Heroku. Настройка переменных окружения.

Практические задания к экзамену

Задание 1. Разработка простого веб-приложения на Flask.

Требования:

Создайте приложение Flask, которое имеет три маршрута: / (возвращает "Главная страница"), /about (возвращает "О нас"), /user/<username> (возвращает приветствие с именем пользователя). Используйте HTML-шаблоны Jinja2 для вывода страниц. Настройте базовый шаблон с общим меню.

Задание 2. Обработка форм и валидация.

Требования:

Разработайте HTML-форму обратной связи (имя, email, сообщение) с методом POST. Реализуйте обработчик на Flask. Проведите серверную валидацию: все поля обязательны, email должен соответствовать формату. При ошибке выведите сообщения

об ошибке, сохранив введенные данные. При успехе перенаправьте на страницу "Спасибо".

Задание 3. Проектирование базы данных и ORM (SQLAlchemy).

Требования:

Спроектируйте базу данных для блога (таблицы User, Post, Comment). Напишите модели SQLAlchemy для Flask. Установите связи: один пользователь — много постов, один пост — много комментариев. Создайте скрипт инициализации БД и добавления тестовых данных.

Задание 4. Реализация CRUD-операций.

Требования:

На основе моделей из Задания 3 реализуйте веб-интерфейс для управления постами: страница со списком всех постов, страница создания нового поста, страница редактирования и кнопка удаления. Используйте Flask-WTF для форм. Обеспечьте защиту от CSRF.

Задание 5. Аутентификация и авторизация.

Требования:

Интегрируйте Flask-Login в приложение. Реализуйте страницы регистрации и входа. Пароли должны хешироваться с помощью werkzeug.security. Настройте защиту маршрутов: создание/редактирование постов доступно только авторизованным пользователям. Отображайте имя текущего пользователя в шапке сайта.

Задание 6. Разработка REST API на Flask.

Требования:

Создайте REST API для управления списком задач (To-Do). Реализуйте эндпоинты:

GET /api/v1/tasks — получить все задачи.

POST /api/v1/tasks — создать задачу.

GET /api/v1/tasks/{id} — получить задачу по ID.

PUT /api/v1/tasks/{id} — обновить задачу.

DELETE /api/v1/tasks/{id} — удалить задачу.

Ответы должны быть в формате JSON. Используйте соответствующие HTTP-коды (200, 201, 404 и т.д.).

Задание 7. Спецификация API (OpenAPI/Swagger).

Требования:

На основе API из Задания 6 разработайте спецификацию в формате YAML (OpenAPI 3.0). Опишите все эндпоинты, параметры запросов, структуру JSON-ответов и возможные коды ошибок. Приложите примеры запросов и ответов.

Задание 8. Знакомство с Django: создание проекта.

Требования:

Создайте новый проект Django и приложение blog. Настройте базу данных SQLite. Создайте суперпользователя. Зарегистрируйте модели Post и Comment в админ-панели Django. Настройте отображение списка постов в админке (list_display, search_fields).

Задание 9. Django Views и Templates.

Требования:

Реализуйте представления (views) для отображения списка всех постов и детальной страницы поста. Настройте URL-маршрутизацию. Создайте HTML-шаблоны с использованием наследования. Выведите заголовок, автора, дату и текст поста.

Задание 10. Сложные запросы в Django ORM.

Требования:

Используя Django ORM, напишите запросы для решения следующих задач:

Получить все посты, опубликованные за последние 7 дней.

Найти всех авторов, у которых больше 5 постов (используйте annotate и count).

Получить список постов, отсортированных по количеству комментариев (от большего к меньшему).

Задание 11. Аудит уязвимого Python-кода.

Требования:

Дан фрагмент кода на Flask:

```
@app.route('/user')
def get_user():
user_id = request.args.get('id')
user = db.execute(f"SELECT * FROM users WHERE id = {user_id}").fetchone()
return f"<h1>Привет, {user['name']}</h1>"
```

Выявите все уязвимости (SQL-инъекция, XSS). Проведите рефакторинг кода, используя параметризованные запросы и экранирование вывода. Обоснуйте внесенные изменения.

Задание 12. Спецификация на уровне базы данных (ER-диаграмма).

Требования:

Для системы "Интернет-магазин" (товары, категории, заказы, пользователи) спроектируйте ER-диаграмму. Опишите таблицы, их атрибуты, типы данных, первичные и внешние ключи. Укажите связи (1:M, M:M). Представьте результат в текстовом и графическом виде.

Задание 13. Сквозная спецификация компонента.

Требования:

Спроектируйте и документируйте модуль "Корзина" для интернет-магазина. Работа должна включать:

ER-диаграмму для хранения данных корзины.

Спецификацию REST API для добавления, изменения количества и удаления товара.

Описание модели данных (JSON) для передачи корзины на клиент.

Опишите возможные коды ошибок (например, товар не найден, недостаточно на складе).

Задание 14. Тестирование веб-приложения на Python.

Требования:

Напишите модульные тесты (с использованием pytest или unittest) для функции валидации пароля. Функция должна проверять длину (минимум 8 символов), наличие хотя бы одной цифры и одной заглавной буквы. Протестируйте граничные случаи (пустая строка, короткий пароль, пароль без цифр).

Задание 15. Загрузка и обработка файлов.

Требования:

Реализуйте форму загрузки аватара пользователя. Настройте серверную часть для проверки: тип файла (только jpg, png), размер (не более 2 МБ). Сохраняйте файл под уникальным именем в папку uploads. Запишите путь к файлу в базу данных. Обеспечьте защиту от загрузки исполняемых скриптов (например, .php или .py).

Задание 16. Развертывание проекта (Deployment).

Требования:

Опишите процесс подготовки Flask-приложения к развертыванию на PythonAnywhere. Настройте файл requirements.txt. Вынесите конфиденциальные данные (SECRET_KEY, данные БД) в переменные окружения. Настройте WSGI-файл. Перечислите 5 пунктов чек-листа безопасности перед публикацией в продакшн.

Задание 17. Интеграция с внешним API.

Требования:

Напишите скрипт на Python, который обращается к публичному API (например, JSONPlaceholder или OpenWeatherMap) с помощью библиотеки requests. Получите данные, распарсите JSON. Обработайте возможные ошибки (таймаут, 404, 500). Выведите полученные данные в консоль или в простой HTML-шаблон.

Задание 18. Пагинация и поиск.

Требования:

Реализуйте страницу со списком статей (из базы данных) с пагинацией по 5 записей. Добавьте форму поиска по заголовку статьи. Используйте SQL-оператор LIKE с экранированием специальных символов. Реализуйте подсчет общего количества страниц для навигации.

Задание 19. Анализ и критика "сырой" спецификации.

Требования:

Вам предоставлено текстовое описание от заказчика: "Нужен сайт для продажи билетов. Чтобы можно было выбрать место, оплатить картой и получить билет на почту. Еще чтобы был личный кабинет". Выявите недостатки (неполнота, противоречия, отсутствие нефункциональных требований). Перепишите это в виде формального технического задания, готового для передачи в разработку (используйте разделы: Цель, Функциональные требования, Нефункциональные требования, Интерфейсы).

Задание 20. Событийно-ориентированная архитектура (WebSockets).

Требования:

Опишите спецификацию для чата на реальном времени. Опишите структуру событий (например, message_sent, user_joined, user_left), формат передаваемых данных (JSON). Приведите пример реализации простого чата с использованием Flask-SocketIO или Django Channels (достаточно описать логику подключения и обработки событий).

Критерии оценивания промежуточной аттестации: экзамен

Оценка	Критерии оценивания по экзамену
Высокий уровень «5» (отлично)	оценку «отлично» заслуживает обучающийся, освоивший знания, умения, компетенции и теоретический материал без пробелов; выполнивший все задания, предусмотренные учебным планом на высоком качественном уровне; практические навыки профессионального применения освоенных знаний сформированы.
Средний уровень «4» (хорошо)	оценку «хорошо» заслуживает обучающийся, практически полностью освоивший знания, умения, компетенции и теоретический материал, учебные задания не оценены максимальным числом баллов, в основном сформировал практические навыки.
Пороговый уровень «3» (удовлетворительно)	оценку «удовлетворительно» заслуживает обучающийся, частично с пробелами освоивший знания, умения, компетенции и теоретический материал, многие учебные задания либо не выполнил, либо они оценены числом баллов близким к минимальному, некоторые практические навыки не сформированы.
Минимальный уровень «2» (неудовлетворительно)	оценку «неудовлетворительно» заслуживает обучающийся, не освоивший знания, умения, компетенции и теоретический материал, учебные задания не выполнил, практические навыки не сформированы.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Методические указания для обучающихся по освоению дисциплины (модуля)

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной

работы обучающихся и промежуточной аттестации.

При проведении учебных занятий по дисциплине обеспечивается развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, преподавание дисциплины в форме курса, составленного на основе результатов научных исследований, проводимых институтом, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей).

Результат обучения считается сформированным, если теоретическое содержание курса освоено полностью; при устных собеседованиях обучающийся исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, обучающийся способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий.

Результат обучения считается несформированным, если обучающийся при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет задания, не демонстрирует необходимых умений, качество выполненных заданий не соответствует установленным требованиям.

Методические указания для обучающихся по освоению дисциплины на занятиях лекционного типа

Лекционный курс предполагает систематизированное изложение основных вопросов тематического плана. В ходе лекционных занятий раскрываются базовые вопросы в рамках каждой темы дисциплины. Обозначаются ключевые аспекты тем, а также делаются акценты на наиболее сложные и важные положения изучаемого материала. Материалы лекций являются опорной основой для подготовки обучающихся к практическим занятиям и выполнения заданий самостоятельной работы, а также к мероприятиям текущего контроля успеваемости и промежуточной аттестации по дисциплине.

В ходе лекционных занятий необходимо вести конспектирование учебного материала. Возможно ведение конспекта лекций в виде интеллект-карт.

Методические указания для обучающихся по освоению дисциплины на занятиях практического типа

Практические (семинарские) занятия представляют собой детализацию лекционного теоретического материала, проводятся в целях закрепления курса и охватывают все основные разделы. Основной формой проведения семинаров и практических занятий является обсуждение наиболее проблемных и сложных вопросов по отдельным темам, а также решение задач и разбор примеров и ситуаций в аудиторных условиях.

Практические (семинарские) занятия обучающихся обеспечивают:

- проверку и уточнение знаний, полученных на лекциях;
- получение умений и навыков составления докладов и сообщений, обсуждения вопросов по учебному материалу дисциплины;
- подведение итогов занятий по рейтинговой системе, согласно технологической карте дисциплины.

Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обеспечивает подготовку обучающегося к аудиторным занятиям и мероприятиям текущего контроля и промежуточной аттестации по изучаемой дисциплине. Результаты этой подготовки проявляются в активности обучающегося на занятиях и в качестве выполненных практических заданий и других форм текущего контроля.

При выполнении заданий для самостоятельной работы рекомендуется проработка материалов лекций по каждой пройденной теме, а также изучение рекомендуемой литературы.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

6. Материально-техническое и учебно-методическое обеспечение по дисциплине (модулю)

Основная литература:

1. Лехмус, М. Ю., Базовые технологии веб-программирования : учебное пособие / М. Ю. Лехмус. — Москва : КноРус, 2025. — 86 с. — ISBN 978-5-406-14103-8. — URL: https://book.ru/book/956632— Текст : электронный.

2. Шаталова, А. Ю., Основы веб-разработки : учебное пособие / А. Ю. Шаталова, М. В. Коротеев. — Москва : КноРус, 2025. — 282 с. — ISBN 978-5-406-14458-9. — URL: https://book.ru/book/957272— Текст : электронный.

3. Полуэктова, Н. Р. Разработка веб-приложений : учебник для вузов / Н. Р. Полуэктова. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 204 с. — (Высшее образование). — ISBN 978-5-534-18645-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/567610

Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС) и базы данных

Доступ к ЭБС предоставляется из любой точки, в которой имеется доступ к сети «Интернет», как на территории Института, так и вне ее (удаленный доступ).

1. Образовательная платформа «ЮРАЙТ» - URL: https://urait.ru/
2. Электронно-библиотечная система «BOOK.ru» - URL: https://www.book.ru
3. Научная электронная библиотека eLIBRARY.RU - URL: https://elibrary.ru (крупнейшая российская база научных публикаций, доступ к рефератам и полным текстам статей).
4. КиберЛенинка - URL: https://cyberleninka.ru (научная электронная библиотека открытого доступа).

Информационные справочные системы

1. Справочная правовая система «Консультант Плюс» - доступ по локальной сети с компьютеров библиотеки и компьютерных классов (актуальная база законодательства РФ, в сфере образования и социальной защиты).

2. Справочная правовая система «Гарант» - доступ по локальной сети с компьютеров библиотеки и компьютерных классов (актуальная база законодательства РФ, включая сферы образования и социальной защиты).

Профессиональные сообщества и IT-порталы

1. Habr - URL: https://habr.com/ (крупнейшее русскоязычное сообщество IT-специалистов, статьи, новости, обсуждения).
2. Stack Overflow - URL: https://stackoverflow.com/ (международный ресурс для программистов, вопросы и ответы).
3. GitHub - URL: https://github.com/ (платформа для хостинга кода, совместной разработки и открытых проектов).
4. CodeProject - URL: https://www.codeproject.com/ (статьи, примеры кода, обсуждения для разработчиков).
5. Microsoft Learn - URL: https://learn.microsoft.com/ (бесплатные учебные материалы и документация по продуктам Microsoft, .NET, Azure и др.).
6. MDN Web Docs - URL: https://developer.mozilla.org/ (ресурс для веб-разработчиков с документацией по HTML, CSS, JavaScript и API).

Научные базы данных зарубежных издательств (открытый доступ)

7. IEEE Xplore - URL: https://ieeexplore.ieee.org/ (доступ к книгам, статьям и материалам конференций по компьютерным наукам, электротехнике и информационным технологиям).
8. ACM Digital Library - URL: https://dl.acm.org/ (библиотека статей и материалов конференций Ассоциации вычислительной техники).
9. SpringerLink - URL: https://link.springer.com/ (книги и журналы издательства Springer, включая серии по информатике).
10. Wiley Online Library - URL: https://onlinelibrary.wiley.com/ (коллекция журналов и книг по направлению «Computer Science & Information Technology»).

Образовательные платформы и онлайн-курсы

11. Национальная платформа «Открытое образование» - URL: https://openedu.ru/ (курсы ведущих российских вузов).
12. Stepik - URL: https://stepik.org/ (российская образовательная платформа с курсами по программированию и информатике).
13. Intuit - URL: https://intuit.ru/ (национальный открытый университет, курсы по информационным технологиям).

Образовательные и справочные порталы

Федеральный портал «Российское образование» - URL: http://www.edu.ru/.

Российское общество «Знание» - URL: https://znanierussia.ru/.

Справочно-информационный портал «Грамота.ру» - URL: http://gramota.ru/ (русский язык и культура речи).

Образовательный портал «Учеба» - URL: http://www.ucheba.com/.

Словари и энциклопедии на Академике - URL: https://dic.academic.ru/.

Проект Государственного института русского языка им. А.С. Пушкина «Образование на русском» - URL: https://pushkininstitute.ru/.

Официальные органы государственной власти и управления

1. Министерство науки и высшего образования Российской Федерации - URL: https://www.minobrnauki.gov.ru/.
2. Министерство просвещения Российской Федерации - URL: https://edu.gov.ru/.
3. Министерство труда и социальной защиты Российской Федерации - URL: https://mintrud.gov.ru/.
4. Федеральная служба по надзору в сфере образования и науки (Рособрнадзор) - URL: http://obrnadzor.gov.ru/.

5. Государственная система правовой информации «Законодательство России» - URL: http://pravo.gov.ru/.

Комплект лицензионного и свободно распространяемого программного обеспечения:

1. LibreOffice - офисный пакет
2. PDFedit – программа для работы с pdf
3. Yandex Browser – браузер
4. Менеджер архивов
5. Libre Base – программа для работы с БД
6. Inkscape – ПО для компьютерной графики
7. DIA – ПО для блока схем и диаграмм
8. 1С: Предприятие – Учебная версия
9. GiMP - Программа обработки изображений

Перечень материально-технического обеспечения включает:

учебные аудитории для проведения занятий лекционного типа, занятий практического (семинарского) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы оснащенные компьютерной техникой с подключением к сети "Интернет" и доступом в электронную информационно-образовательную среду Института.

Наименование помещения. Перечень основного оборудования	Адрес
Учебная аудитория № 217 Учебная аудитория для проведения занятий лекционного и практического типа, выполнения курсовых работ, групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой аттестации). Оборудование: рабочее место преподавателя (1); рабочие места обучающихся (36); ноутбук с лицензионным ПО (LibreOffice) и возможностью выхода в интернет (1); мультимедийное оборудование (1); доска учебная (1); книжный шкаф (1); сплит-система(1); учебно-наглядные пособия; доступ в электронную информационно-образовательную среду Института.	350002, Краснодарский край, г. Краснодар, Центральный внутригородской округ, ул. им. Леваневского, д. 187/1
Аудитория № 218 Помещение для самостоятельной работы обучающихся Оборудование: рабочие места обучающихся (17); персональный компьютер с лицензионным ПО и возможностью выхода в Интернет (17); книжный шкаф (1); сплит-система (1); учебно-наглядные пособия;	350002, Краснодарский край, г. Краснодар, Центральный внутригородской округ, ул. им. Леваневского, д. 187/1

доступ в электронную информационно-образовательную среду Института.	
Учебная аудитория № 304 (компьютерный класс) Учебная аудитория для проведения занятий лекционного и практического типа, выполнения курсовых работ, групповых и индивидуальных консультаций, текущего контроля, промежуточной аттестации. Оборудование: рабочее место преподавателя (1); рабочие места обучающихся (25); персональный компьютер с лицензионным ПО и возможностью выхода в интернет (26); мультимедийное оборудование (1); доска учебная (1); книжный шкаф (1); сплит-система (1); учебно-наглядные пособия; доступ в электронную информационно-образовательную среду Института.	350002, Краснодарский край, г. Краснодар, Центральный внутригородской округ, ул. им. Леваневского, д. 187/1