МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Колледж информационных и креативных технологий

РАБОЧАЯ ПРОГРАММА

МДК.04.03 Модульное тестирование веб-приложений

ПМ.04 Разработка программных решений с применением технологий цифровой экономики

программы подготовки специалистов среднего звена
специальность
09.02.11 Разработка и управление программным обеспечением

Форма обучения - очная

Владивосток 2026

Рабочая программа МДК.04.03 Модульное тестирование веб-приложений разработана на основе федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением, квалификация программист, утвержденного приказом Министерства просвещения Российской Федерации от 24 февраля 2025 г. № 138

Разработчик(и): Д.А. Трофимов, преподаватель

Рассмотрено и одобрено на заседании цикловой методической комиссии
Протокол № 9 от «20» мая 2026 г.

Председатель ПЦК _____ П.В. Калашников

СОДЕРЖАНИЕ

1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ

4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ МЕЖДИСЦИПЛИНАРНОГО КУРСА

1.1. Цель и место междисциплинарного курса в структуре образовательной программы

Междисциплинарный курс МДК.04.03 Модульное тестирование веб-приложений входит в профессиональный цикл учебного плана образовательной программы в структуру профессионального модуля ПМ.04 «Разработка программных решений с применением технологий цифровой экономики».

Цель: освоение принципов и инструментов модульного (Unit) и компонентного тестирования, методологии разработки через тестирование (TDD), методов аудита безопасности исходного кода и программных модулей, а также технологий технического сопровождения и аварийного восстановления для создания надежных и защищённых веб-приложений.

1.2. Планируемые результаты освоения междисциплинарного курса

По итогам освоения курса, обучающиеся должны продемонстрировать результаты обучения, соотнесённые с результатами освоения ООП СПО, приведенные в таблице.

Код	Наименование видов деятельности и профессиональных компетенций
ВД 4	Разработка программных решений с применением технологий цифровой экономики
ПК 4.2	Разрабатывать полнофункциональные веб-интерфейсы с применением современных фреймворков и принципов цифровой доступности
ПК 4.4	Обеспечивать соответствие веб-приложений требованиям информационной безопасности и регуляторным стандартам цифровой экономики РФ
ПК 4.5	Осуществлять командную разработку веб-решений с использованием цифровых инструментов кооперации и управления жизненным циклом ПО

В результате освоения междисциплинарного курса обучающийся должен:

Код ПК	Уметь	Знать	Владеть навыками
ПК 4.2	Собирать, очищать и преобразовывать большие объёмы данных. Проектировать и реализовывать ML-модели в составе ПО. Интегрировать модели в программные системы (через API, встраивание и др.). Оценивать качество и интерпретируемость моделей. Выбирать подходящие алгоритмы под задачу. Документировать данные и модели.	Основные концепции Big Data (объёмы, скорости, разнообразие, достоверность). Архитектуры обработки данных (Hadoop, Spark и др.). Этапы жизненного цикла данных. Принципы машинного обучения (обучение с учителем/без учителя, валидация, переобучение). Типы моделей (регрессия, классификация, кластеризация и др.). Популярные библиотеки (scikit-learn, TensorFlow, PyTorch, pandas). Методы предобработки и визуализации данных, этические и правовые аспекты работы с данными.	Программирования на Python (или другом языке, применимом к анализу данных). Работы с библиотеками машинного обучения и анализа данных. Использования систем распределённой обработки, визуализации результатов, отладки и оптимизации ML-решений в составе программных продуктов.
ПК 4.4	Интегрировать механизмы защиты в	Принципы обеспечения информационной	Применения инструментов тестирования на

	архитектуру и код программных продуктов. Проводить анализ уязвимостей и рисков. Настраивать и применять инструменты кибербезопасности. Обеспечивать соответствие требованиям законодательства в области персональных данных и защиты информации. Реализовывать политики управления доступом и шифрования.	безопасности (конфиденциальность, целостность, доступность). Основные угрозы и атаки в цифровой среде, нормативно-правовую базу (ФЗ-152, GDPR и др.). Методы и средства защиты данных (шифрование, аутентификация, авторизация, DLP, WAF, IDS/IPS). Подходы безопасной разработки (DevSecOps, Secure SDLC). Стандарты и фреймворки (ISO/IEC 27001, NIST, OWASP).	проникновение и анализа защищённости. Работы с сертификатами и ключами. Настройки безопасных API и сетевых взаимодействий, безопасного хранения и передачи данных, а также внедрения практик DevSecOps в процесс разработки ПО.
ПК 4.5	Выбирать подходящие облачные сервисы под задачи ПО. Разворачивать и настраивать облачную инфраструктуру. Автоматизировать сборку, тестирование и деплой приложений с использованием облачных CI/CD-инструментов. Управлять масштабированием и производительностью. Обеспечивать безопасность и соответствие в облаке.	Модели облачных сервисов (IaaS, PaaS, SaaS). Архитектурные принципы облачных решений (масштабируемость, отказоустойчивость, серверлесс). Основные облачные платформы (AWS, Microsoft Azure, Google Cloud, Yandex Cloud и др.). Сервисы для хранения, вычислений, CI/CD, мониторинга и управления инфраструктурой. Базовые понятия виртуализации и контейнеризации (Docker, Kubernetes).	Работы с облачными консолями и CLI. Написания инфраструктуры как кода (Terraform, CloudFormation). Управления контейнерами и оркестрацией (Kubernetes). Настройки сетей. Мониторинга и логирования в облаке, а также оптимизации затрат на облачные ресурсы.

2. СТРУКТУРА И СОДЕРЖАНИЕ МЕЖДИСЦИПЛИНАРНОГО КУРСА

2.1 Объем МДК и виды учебной работы

Вид учебной работы	Объем в часах	В т.ч. в форме практ. подготовки
Учебные занятия	60	40
Самостоятельная работа	26	
Консультации		
Промежуточная аттестация		
Всего	**86**	**40**

2.2. Тематический план и содержание МДК

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенций, формированию

			которых способствует элемент программы
Раздел 1. Основы модульного тестирования и контроль качества в жизненном цикле веб-разработки		**30**	**ПК 4.2, ПК 4.5**
Тема 1.1. Принципы модульного тестирования (Unit Testing) и методология TDD	**Содержание учебных материалов**	**4**	
	Роль модульного тестирования в пирамиде тестирования веб-приложений, принципы FIRST и структура модульного теста по паттерну AAA (Arrange-Act-Assert).	1	ПК 4.2, ПК 4.5
	Среда выполнения и фреймворки модульного тестирования (Jest / Vitest / Mocha): тестовые наборы (describe), тест-кейсы (it/test) и утверждения (Assertions / Matchers).	1	ПК 4.2
	Методология разработки через тестирование (TDD: цикл Red-Green-Refactor) и поведенческий подход (BDD), проектирование чистых функций и тестируемой архитектуры кода.	2	ПК 4.2, ПК 4.5
	В том числе практических занятий	**8**	
	Настройка тестового окружения и разработка модульных тестов для чистых функций бизнес-логики с проверкой позитивных, негативных и граничных сценариев.	4	ПК 4.2
	Разработка программного модуля валидации и расчёта данных по техническому заданию с использованием цикла TDD (Red-Green-Refactor).	4	ПК 4.2, ПК 4.5
Тема 1.2. Изоляция модулей, тестовые двойники (Mocks) и метрики покрытия кода	**Содержание учебных материалов**	**4**	
	Проблема связности модулей и методы инверсии зависимостей (Dependency Injection) для обеспечения изолированного модульного тестирования.	1	ПК 4.2
	Классификация и применение тестовых двойников (Test Doubles): пустышки (Dummy), стабы (Stubs), шпионы (Spies), моки (Mocks) и фейки (Fakes).	2	ПК 4.2
	Метрики качества модульного тестирования: покрытие операторов (Statement), ветвлений (Branch), функций (Function) и строк (Line Coverage), мутационное тестирование.	1	ПК 4.2, ПК 4.5
	В том числе практических занятий	**8**	
	Разработка модульных тестов с использованием шпионов (Spies) и моков (Mocks) для изоляции внешних сервисов, файловой системы и базы	4	ПК 4.2

	данных.		
	Анализ отчёта о покрытии кода тестами (Code Coverage Report) и доработка модульных тестов для достижения заданного порога покрытия по ТЗ.	4	ПК 4.2, ПК 4.5
	В том числе самостоятельная работа обучающихся Рефакторинг сильно связанного программного модуля веб-приложения и покрытие его изолированными Unit-тестами.	**6**	**ПК 4.2, ПК 4.5**
Раздел 2. Тестирование веб-интерфейсов и цифровой доступности, безопасность по стандартам РФ и командная интеграция		**56**	**ПК 4.2, ПК 4.4, ПК 4.5**
Тема 2.1. Модульное тестирование современных веб-интерфейсов, API-сервисов и цифровой доступности (a11y)	**Содержание учебных материалов**	**6**	
	Особенности модульного тестирования асинхронного кода: тестирование промисов (Promises), конструкций async/await, таймеров (Fake Timers) и клиентских API-запросов (fetch/axios mocks).	1	ПК 4.2
	Модульное и компонентное тестирование полнофункциональных веб-интерфейсов на современных фреймворках (React / Vue, Testing Library, рендеринг в виртуальном DOM jsdom).	1	ПК 4.2
	Тестирование пользовательских событий (клики, ввод в формы, навигация с клавиатуры), входных свойств (Props), хуков состояния и снимочное тестирование (Snapshot Testing).	1	ПК 4.2
	Принципы цифровой доступности веб-интерфейсов (Accessibility / a11y, стандарты WCAG и ГОСТ Р 52872): семантическая разметка, атрибуты WAI-ARIA и поддержка скринридеров.	1	ПК 4.2
	Автоматизированное модульное тестирование цифровой доступности компонентов интерфейса (поиск элементов по ARIA-ролям getByRole, проверка контрастности, фокуса и правил jest-axe).	2	ПК 4.2
	В том числе практических занятий	**8**	
	Разработка модульных тестов для асинхронных сервисов работы с REST API с использованием моков сетевых запросов и проверкой обработки ошибок сервера.	4	ПК 4.2
	Модульное тестирование UI-компонентов и интерактивных форм современного фреймворка с проверкой критериев цифровой доступности (WAI-ARIA, jest-axe, управление с	4	ПК 4.2

	клавиатуры).		
	В том числе самостоятельная работа обучающихся Разработка и покрытие тестами полнофункционального веб-интерфейса с соблюдением принципов цифровой доступности для пользователей с ОВЗ.	**8**	**ПК 4.2**
Тема 2.2. Обеспечение и тестирование информационной безопасности и регуляторных стандартов РФ	**Содержание учебных материалов**	**3**	
	Требования информационной безопасности и регуляторные стандарты цифровой экономики РФ (152-ФЗ «О персональных данных», требования ФСТЭК, безопасная обработка ПДн) и уязвимости OWASP Top 10.	1	ПК 4.4
	Проектирование модульных тестов безопасности (Security Unit Tests): проверка функций санитизации, экранирования XSS, защиты от инъекций, маскирования ПДн и верификации токенов доступа.	1	ПК 4.4
	Статический анализ безопасности кода (SAST-линтеры), контроль отсутствия утечек секретов и аудит сторонних пакетов (npm audit) с учётом требований импортозамещения и регуляторики РФ.	1	ПК 4.4
	В том числе практических занятий	**8**	
	Проведение аудита безопасности модулей авторизации и обработки персональных данных с написанием модульных тестов на устойчивость к XSS, инъекциям и соблюдение 152-ФЗ.	4	ПК 4.4
	Автоматизированный аудит безопасности зависимостей и исходного кода веб-приложения с устранением выявленных уязвимостей по регуляторным стандартам.	4	ПК 4.4
	В том числе самостоятельная работа обучающихся Составление отчёта о соответствии модулей веб-приложения требованиям информационной безопасности и регуляторным стандартам цифровой экономики РФ.	**6**	**ПК 4.4**
Тема 2.3. Командная разработка веб-решений, CI/CD-автоматизация и управление жизненным циклом ПО	**Содержание учебных материалов**	**3**	
	Организация командной разработки веб-решений: стратегии ветвления Git (GitFlow / Trunk-Based), коллективная работа с репозиториями, Pull/Merge Requests и проведение Code Review тестов.	1	ПК 4.5
	Интеграция модульного тестирования, линтеров и проверок безопасности в	1	ПК 4.5

	конвейеры управления жизненным 1циклом ПО и CI/CD (GitHub Actions / GitLab CI / отечественные платформы).		
	Цифровые инструменты кооперации команды (таск-трекеры, системы управления тестированием, документация), контроль качества релизов (Quality Gates) и откат сбойных сборок (Rollback).	1	ПК 4.5
	В том числе практических занятий	**8**	
	Командная разработка и ревью (Code Review) модульных тестов через Pull/Merge Request с разрешением конфликтов слияния в системе контроля версий Git.	4	ПК 4.5
	Настройка конвейера CI/CD для автоматического запуска модульных тестов, проверки порога покрытия (Quality Gate) и блокировки слияния при ошибках.	4	ПК 4.5
	В том числе самостоятельная работа обучающихся Регламентация процесса командной разработки, ревью кода и автоматизированной проверки качества в системе управления жизненным циклом ПО.	**6**	**ПК 4.5**
Всего		**86**	**ПК 4.2, ПК 4.4, ПК 4.5**

3. УСЛОВИЯ РЕАЛИЗАЦИИ МЕЖДИСЦИПЛИНАРНОГО КУРСА

3.1. Материально-техническое обеспечение

Для реализации программы МДК.04.03 «Модульное тестирование веб-приложений» предусмотрено наличие следующих помещений:

Основное оборудование: Компьютеризированные рабочие места обучающихся с выходом в сеть "Интернет" - 20 шт., стулья - 20 шт., комплект мебели (стол и стул) для преподавателя - 1 шт., графические станции ПК №2 WorkStation WS 917 20 шт., автоматизированное рабочее место преподавателя ПК №2 WorkStation WS 917, графический планшет Wacom - 20 шт., мультимедийное оборудование - 1 шт., телевизор, экран, многофункциональное устройство (МФУ) формат А3 -1 шт., доска маркерная - 1 шт.

Программное обеспечение: PyCharm, Visual Studio Code, IntelliJ IDEA, JetBrains Rider, Python (Anaconda3) с TensorFlow/PyTorch, Microsoft Visio, Draw.io, Figma, Figma, Adobe Illustrator, MySQL Workbench, HeidiSQL, MySQL, PostgreSQL, Git, Docker Desktop, Microsoft Project, Draw.io, Microsoft SQL Server Management Studio, Microsoft Office.

Основное оборудование: Рабочие места на базе компьютерной техники с возможностью подключения к информационно-телекоммуникационной сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду ВВГУ: комплекты учебной мебели (столы и стулья) – 19 шт., персональные компьютеры (облачные мониторы) - 19 шт; доска маркерная - 1шт., лазерный копир-принтер-сканер с 2 лотками Xerox WorkCentre 3345 DNI.

Программное обеспечение: Microsoft Windows 7 Professional; Microsoft Office ProPlus 2010 Russian Acdmc; СПС КонсультантЮрист: Версия Проф; Adobe Acrobat Reader; Google Chrome; Adobe Flash Player; 7-Zip 18.01 (x64).

3.2. Учебно-методическое обеспечение

Для реализации программы учебной дисциплины библиотечный фонд ВВГУ укомплектован печатными и электронными изданиями.

Обучающиеся из числа инвалидов и лиц с ограниченными возможностями здоровья обеспечены печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Основные источники

1. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 273 с. — (Профессиональное образование). — ISBN 978-5-534-20362-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/562355

2. Грекул, В. И. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 404 с. — (Профессиональное образование).
— ISBN 978-5-534-19506-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/566739

3. Соколова, В. В. Разработка мобильных приложений : учебное пособие для среднего профессионального образования / В. В. Соколова. — Москва : Издательство Юрайт, 2024. —160 с. — (Профессиональное образование). — ISBN 978-5-534-16868-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/542342

Дополнительные источники

1. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 196 с. — (Профессиональное образование). — ISBN 978-5-534- 18760-1. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/561922

2. Зыков, С. В. Объектно-ориентированное программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2024. — 151 с. — (Высшее образование). — ISBN 978-5-534-16941-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/537385

3. Станкевич, Л. А. Интеллектуальные системы и технологии : учебник и практикум для среднего профессионального образования / Л. А. Станкевич. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 478 с. — (Профессиональное образование). — ISBN 978-5-534-20364-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/566524

Электронные ресурсы

1. Информационно-справочная система «Консультант Плюс» – http://www.consultant.ru/

2. Профессиональная база данных: "Открытая база ГОСТов"/ Режим доступа: http://standartgost.ru/, доступ свободный

3. Электронная библиотечная система «РУКОНТ» - Режим доступа: https://lib.rucont.ru/

4. Электронная библиотечная система ZNANIUM.COM - Режим доступа: https://znanium.com/

5. Электронно-библиотечная система издательства "Юрайт" - Режим доступа: https://urait.ru/

6. Электронно-библиотечная система издательства "Лань" - Режим доступа: https://e.lanbook.com/

7. Президентская библиотека им. Б.Н. Ельцина (база данных различных профессиональных областей) Режим доступа: https://www.prlib.ru/, доступ свободный

4 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ИЗУЧЕНИЮ И РЕАЛИЗАЦИИ МЕЖДИСЦИПЛИНАРНОГО КУРСА

Начиная изучение дисциплины, студенту необходимо:

- ознакомиться с программой, изучить список рекомендуемой литературы;
- внимательно разобраться в структуре курса, в системе распределения учебного материала по видам занятий, формам контроля, чтобы иметь представление о курсе в целом.

Самостоятельная работа студентов (СРС) — это деятельность учащихся, которую они совершают без непосредственной помощи и указаний преподавателя, руководствуясь сформировавшимися ранее представлениями о порядке и правильности выполнения операций. Цель СРС в процессе обучения заключается, как в усвоении знаний, так и в формировании умений и навыков по их использованию в новых условиях на новом учебном материале. Самостоятельная работа призвана обеспечивать возможность осуществления студентами самостоятельной познавательной деятельности в обучении, и является видом учебного труда, способствующего формированию у студентов самостоятельности. Кроме того, для расширения и углубления знаний по дисциплине целесообразно использовать: публикации в тематических журналах; полнотекстовые базы данных библиотеки; имеющиеся в библиотеках вуза. Успешное освоение дисциплины предполагает активную работу студентов на всех занятиях аудиторной формы: лекций и практических занятий, выполнение аттестационный мероприятий, эффективную самостоятельную работу.

4.1 Методические рекомендации обучающимся по обеспечению самостоятельной работы

Самостоятельная работа проводится с целью: систематизации, закрепления, углубления и расширения теоретических знаний и практических умений, приобретаемых студентами в ходе аудиторных занятий; формирования умений использовать специальную литературу; развития познавательных способностей и активности обучающихся; формирования самостоятельности мышления, способности к саморазвитию, самосовершенствованию и самореализации; развития исследовательских умений.

Для лучшего освоения материала и систематизации знаний по дисциплине необходимо отвечать на вопросы для самоконтроля. Во время самостоятельной проработки лекционного материала особое внимание следует уделять возникшим вопросам, непонятным терминам, спорным точкам зрения. В случае необходимости обращаться к преподавателю за консультацией. Студент должен четко уяснить, что именно с лекции начинается его подготовка к практическому занятию. Вместе с тем, лекция лишь организует мыслительную деятельность, но не обеспечивает глубину усвоения программного материала.

При подготовке к практическому занятию особое внимание необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. В процессе подготовки рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретается практика в изложении и разъяснении полученных знаний, развивается речь.

При необходимости следует обращаться за консультацией к преподавателю. Перед консультацией, необходимо хорошо продумать вопросы, которые требуют разъяснения.

В процессе изучения дисциплины студенту необходимо ориентироваться на самостоятельную проработку лекционного материала, подготовку к практическим

занятиям, выполнение тестов, кейсовых заданий, самостоятельное изучение некоторых разделов курса. Для проведения занятий лекционного типа используются учебно-наглядные пособия в форме презентационных материалов, обеспечивающих тематические иллюстрации, соответствующие темам лекций.

4.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

5 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ КУРСА

Код ПК	Критерии оценки результата (показатели освоенности компетенций)	Формы контроля и методы оценки
ПК.4.2	Оценка «отлично»: веб-интерфейсы и модульные/компонентные тесты разработаны на современном фреймворке в полном соответствии с ТЗ и стандартами цифровой доступности. Оценка «хорошо»: веб-интерфейсы и тесты к ним разработаны корректно, но имеются мелкие недочёты в атрибутах доступности или покрытии граничных состояний. Оценка «удовлетворительно»: веб-интерфейсы и тесты разработаны частично, требования цифровой доступности соблюдены не в полном объёме.	Защита практических работ, проверка прохождения компонентных тестов и тестов цифровой доступности (jest-axe).
ПК.4.4	Оценка «отлично»: обеспечено полное соответствие модулей веб-приложения требованиям информационной безопасности (OWASP) и регуляторным стандартам РФ (152-ФЗ), все защитные тесты проходят успешно. Оценка «хорошо»: требования безопасности и регуляторики выполнены, но имеются незначительные неточности в отчёте об аудите или настройках статического анализатора. Оценка «удовлетворительно»: требования безопасности и нормативных стандартов РФ реализованы и протестированы частично.	Проверка практических заданий по тестированию безопасности и комплаенса РФ, защита отчёта об аудите безопасности.
ПК.4.5	Оценка «отлично»: командная разработка, работа с Git, Code Review и автоматический прогон модульных тестов в конвейере CI/CD настроены и выполняются без ошибок. Оценка «хорошо»: инструменты кооперации, контроля версий и CI/CD используются корректно, допускаются мелкие недочёты в оформлении Pull Request. Оценка «удовлетворительно»: цифровые инструменты командной разработки и управления жизненным циклом ПО используются частично или с ошибками.	Проверка истории коммитов, Pull Requests, результатов Code Review и работы настроенного конвейера CI/CD.

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Колледж информационных и креативных технологий

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущего контроля и промежуточной аттестации

по **МДК 04.03 Модульное тестирование веб-приложений**

ПМ.03 Разработка программных решений с применением технологий цифровой экономики

программы подготовки специалистов среднего звена
09.02.11 Разработка и управление программным обеспечением

Форма обучения: очная

Владивосток 2026

1. Общие сведения

Фонд оценочных средств (далее – ФОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу МДК.04.03 «Модульное тестирование веб-приложений» в соответствии с техническим заданием и оценка их качества

ФОС включают в себя контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине, которая проводится в форме дифференцированного зачёта (с использованием оценочного средства - устный опрос в форме ответов на вопросы билетов, устный опрос в форме собеседования, выполнение письменных заданий, тестирование и т.д.).

2. Планируемые результаты обучения, обеспечивающие результаты освоения образовательной программы

Код ПК	Код результата обучения	Наименование результата обучения
ПК 4.2	З1	Регламенты технического сопровождения, мониторинга работоспособности и поддержки программных модулей веб-приложений в соответствии с техническим заданием.
	З2	Методы логирования работы модулей, диагностики сбоев, стратегии резервного копирования (Backup) и аварийного восстановления (Disaster Recovery) веб-приложений.
	З3	Правила безопасного выпуска обновлений (патчей, хотфиксов), интеграции модульных тестов в конвейеры CI/CD и отката версий (Rollback) при сбоях.
	У1	Осуществлять мониторинг работоспособности веб-приложения, анализировать логи ошибок модулей и диагностировать причины технических сбоев.
	У2	Выполнять резервное копирование и полное восстановление работоспособности веб-приложения и его баз данных после сбоев в соответствии с ТЗ.
	У3	Проводить техническое обслуживание, установку обновлений и автоматизированную регрессионную проверку модулей после восстановления.
	П1	Технического сопровождения, мониторинга работоспособности и диагностики сбоев в модулях веб-приложений в соответствии с техническим заданием.
	П2	Резервного копирования, восстановления работоспособности и безошибочного обновления веб-приложений после нештатных ситуаций.
ПК 4.4	З1	Принципы модульного тестирования (Unit Testing), структуру модульного теста (паттерн AAA), методологии TDD/BDD и требования к тестируемости кода.
	З2	Фреймворки модульного и компонентного тестирования (Jest, Vitest, Mocha, React Testing Library), виды тестовых двойников (Mocks, Stubs, Spies, Fakes) и метрики покрытия кода (Code Coverage).
	З3	Методики тестирования асинхронных функций, клиент-серверных модулей (API), пользовательских компонентов и правила документирования дефектов.
	У1	Проектировать и разрабатывать модульные (Unit) и компонентные тесты для проверки бизнес-логики, граничных значений и обработки исключений по ТЗ.
	У2	Изолировать тестируемые модули от внешних зависимостей (БД, сетевых запросов, таймеров) с помощью заглушек и моков (Mocks, Stubs, Spies).

Код ПК	Код результата обучения	Наименование результата обучения
	У3	Запускать тестовые наборы, анализировать отчёты о покрытии кода (Code Coverage), проводить отладку модулей и оформлять отчёты о дефектах (баг-репорты).
	П1	Разработки и выполнения модульных, компонентных и интеграционных тестов для проверки разработанного веб-приложения.
	П2	Использования фреймворков автоматизированного тестирования, изоляции зависимостей (Mocking) и оценки метрик покрытия кода тестами.
ПК 4.5	З1	Регламенты обеспечения безопасности веб-приложений и классификацию уязвимостей программных модулей (OWASP Top 10: XSS, инъекции, небезопасная десериализация, ошибки авторизации).
	З2	Методики статического анализа безопасности кода (SAST), написания модульных тестов безопасности (Security Unit Tests) и аудита сторонних зависимостей (npm audit).
	У1	Проводить аудит безопасности программных модулей, функций валидации, криптографии и сторонних пакетов веб-приложения в соответствии с регламентом.
	У2	Разрабатывать модульные тесты для проверки устойчивости функций к вредоносным входным данным (инъекциям, XSS-векторам, переполнению) и составлять отчёт об аудите.
	П1	Проведения аудита безопасности исходного кода, программных модулей и зависимостей веб-приложения в соответствии с регламентом по безопасности.
	П2	Тестирования защищённости модулей веб-приложения от типовых атак и документирования результатов аудита безопасности.

3. Описание процедуры оценивания

Результаты обучения по модулю, уровень сформированности компетенций оцениваются по четырёх бальной шкале оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» (по бальной системе. Максимальная сумма баллов по дисциплине равна 100 баллов.)

Текущая аттестация по модулю проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине).

При проведении промежуточной аттестации оценивается достижение студентом запланированных по модулю результатов обучения, обеспечивающих результаты освоения образовательной программы в целом. Оценка на зачете / экзамене выставляется с учетом оценок, полученных при прохождении текущей аттестации.

Критерии оценивания устного ответа

(оценочные средства: контрольная работа, собеседование, устное сообщение.)

5 баллов - ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

4 балла - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

3 балла – ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

2 балла – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

Критерии оценивания письменной работы

(оценочные средства: реферат, конспект, контрольная работа, письменный отчет по практической работе, портфолио, доклад (сообщение), в том числе выполненный в форме презентации, творческое задание, курсовая работа).

5 баллов - студент выразил своё мнение по сформулированной проблеме, аргументировал его, точно определив ее содержание и составляющие. Проблема раскрыта полностью, выводы обоснованы. Приведены данные отечественной и зарубежной литературы, статистические сведения, информация нормативно-правового характера. Студент владеет навыком самостоятельной работы по заданной теме; методами и приемами анализа теоретических и/или практических аспектов изучаемой области. Фактических ошибок, связанных с пониманием проблемы, нет; графически работа оформлена правильно.

4 балла - работа характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Проблема раскрыта. Не все выводы сделаны и/или обоснованы. Для аргументации приводятся данные отечественных и зарубежных авторов. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет. Допущены одна-две ошибки в оформлении работы.

3 балла – студент проводит достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимает базовые основы и теоретическое обоснование выбранной темы. Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы Проведен анализ проблемы без привлечения дополнительной литературы. Допущено не более 2 ошибок в смысле или содержании проблемы, оформлении работы.

2 балла - работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Проблема не раскрыта. Выводы отсутствуют. Допущено три или более трех ошибок в смысловом содержании раскрываемой проблемы, в оформлении работы.

Критерии оценивания тестового задания

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Количество правильных ответов	91 % и ≥	от 81% до 90,9 %	не менее 70%	менее 70%

Критерии выставления оценки студенту на зачете/ экзамене

(оценочные средства: устный опрос в форме ответов на вопросы билетов, устный опрос в форме собеседования, выполнение письменных разноуровневых задач и заданий, комплексная расчетно-графическая работа, творческое задание, кейс-задача, портфолио, проект и т.п.)

Оценка по промежуточной аттестации	Характеристика качества сформированности компетенций
«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации.
«не зачтено» / «неудовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы.

4. Примеры оценочных средств для проведения промежуточной аттестации

Вопросы для дифференцированного зачета

1. Что такое модульное тестирование (Unit Testing) и какое место оно занимает в общей пирамиде тестирования веб-приложений?
2. Что означает аббревиатура принципов FIRST (Fast, Isolated, Repeatable, Self-Validating, Timely) в модульном тестировании?
3. Как устроена структура модульного теста по паттерну AAA (Arrange, Act, Assert) или Given-When-Then?
4. Что такое чистая функция (Pure Function) и почему детерминированный код значительно проще покрывать модульными тестами?

5. В чём суть методологии разработки через тестирование (TDD — Test-Driven Development)? Опишите цикл Red-Green-Refactor.
6. Чем поведенческая разработка (BDD — Behavior-Driven Development) отличается от классического TDD?
7. Как устроены современные фреймворки модульного тестирования (Jest, Vitest, Mocha)? Назначение блоков describe, test/it.
8. Что такое утверждения (Assertions) и матчи (Matchers) в модульных тестах? Чем отличается проверка строгого равенства (toBe) от глубокого сравнения объектов (toEqual)?
9. Зачем в модульном тестировании используются хуки жизненного цикла тестового набора (beforeEach, afterEach, beforeAll, afterAll)?
10. Как с помощью техник классов эквивалентности и анализа граничных значений подобрать входные данные для модульного теста функции?
11. Как в модульных тестах проверяется корректная генерация и перехват исключений (ошибок throw Error)?
12. Что такое параметризованные тесты (test.each) и как они помогают избежать дублирования кода при проверке множества наборов данных?
13. Почему при модульном тестировании необходимо изолировать тестируемый модуль от базы данных, файловой системы и сторонних сетевых сервисов?
14. Что такое инверсия зависимостей (Dependency Injection) и как она повышает тестируемость модулей веб-приложения?
15. Что такое тестовые двойники (Test Doubles)? Объясните разницу между пустышкой (Dummy), стабом (Stub), шпионом (Spy), моком (Mock) и фейком (Fake).
16. Как с помощью шпионов (Spies) проверить количество вызовов функции, переданные ей аргументы и возвращённое значение?
17. Как полностью замокать (заменить имитацией) внешний программный модуль или библиотеку в тестовом фреймворке?
18. Почему важно очищать состояние моков и шпионов (clearAllMocks, resetAllMocks, restoreAllMocks) между тестами?
19. В чём особенности модульного тестирования асинхронного кода (Promises, async/await, колбэки)? Как тестировать отклонение промиса (rejects)?
20. Как тестировать функции, зависящие от времени (setTimeout, setInterval, Date.now), с помощью фейковых таймеров (Fake Timers)?
21. Как организовать модульное тестирование клиентских сервисов взаимодействия с REST API путём имитации fetch или Axios?
22. Как проводится модульное и компонентное тестирование пользовательского интерфейса в виртуальной среде jsdom (на примере React Testing Library)?
23. Как в компонентных тестах имитировать действия пользователя (клики, ввод текста, отправку формы) и проверять изменение состояния интерфейса?
24. Что такое тестирование снимками (Snapshot Testing), в каких задачах оно полезно и какие у него есть недостатки?
25. Что такое метрики покрытия кода тестами (Code Coverage)? Опишите разницу между покрытием строк (Line), операторов (Statement), функций (Function) и ветвлений (Branch).
26. Почему 100% покрытие кода строками (Line Coverage) не гарантирует полного отсутствия ошибок в модуле? Что такое мутационное тестирование?
27. Что такое «хрупкие тесты» (Fragile / Flaky Tests), почему они возникают при привязке к деталям реализации и как от них избавиться?
28. Каков жизненный цикл дефекта и как правильно задокументировать ошибку, обнаруженную при тестировании модуля (баг-репорт, Severity, Priority)?
29. Что представляет собой регламент по безопасности веб-приложения и какие этапы включает аудит безопасности программных модулей?

30. Какие категории уязвимостей входят в стандарт OWASP Top 10 и какие из них можно выявить на уровне модульного тестирования кода?
31. Как спроектировать модульные тесты безопасности (Security Unit Tests) для проверки функций фильтрации и экранирования от XSS-атак?
32. Как проверить устойчивость модулей работы с данными к SQL- и NoSQL-инъекциям, а также к атакам переполнения и подмены типов параметров?
33. Что такое уязвимость отказа в обслуживании через регулярные выражения (ReDoS) и как тестировать регулярные выражения валидаторов на безопасность?
34. Как проводить аудит и модульное тестирование функций криптографии, хэширования паролей (bcrypt/argon2) и генерации/проверки JWT-токенов?
35. Как тестировать серверные модули и промежуточное ПО (Middleware) на корректность разграничения ролевого доступа (RBAC) и защиту от IDOR-уязвимостей?
36. Что такое статический анализ безопасности исходного кода (SAST) и как настроить линтеры безопасности для автоматической проверки модулей?
37. Как проводится аудит безопасности сторонних пакетов и дерева зависимостей проекта (npm audit, анализ CVE) и как устранять уязвимости в зависимостях?
38. Какова структура итогового отчёта по результатам аудита безопасности веб-приложения и как классифицируются выявленные уязвимости по уровню риска?
39. Что включает в себя техническое сопровождение веб-приложения в соответствии с техническим заданием и как модульные тесты помогают при рефакторинге и обновлениях?
40. Как интегрировать автоматический запуск модульных тестов и проверку безопасности в конвейер непрерывной интеграции (CI/CD) при каждом коммите?
41. Как организовать структурированное логирование ошибок в программных модулях веб-приложения для быстрой диагностики сбоев в эксплуатации?
42. Почему при исправлении любого бага в процессе технического сопровождения необходимо сначала написать падающий модульный тест, воспроизводящий этот баг?
43. Сравните стратегии резервного копирования (полное, инкрементальное, дифференциальное) и объясните показатели восстановления RPO и RTO.
44. Каков пошаговый алгоритм аварийного восстановления (Disaster Recovery) веб-приложения и его базы данных после критического сбоя?
45. Как выполняется безопасный выпуск хотфиксов (Hotfixes), быстрый откат сбойного обновления (Rollback) и пост-аварийное дымовое тестирование (Smoke Test)?

Экзаменационные билеты

Билет 1

1. Принципы модульного тестирования (FIRST), структура модульного теста (Arrange-Act-Assert) и правила проектирования тестируемых чистых функций.
2. Разработайте набор модульных тестов (позитивные, негативные и граничные случаи) для функции расчёта стоимости доставки заказа в зависимости от веса, расстояния и статуса VIP-клиента.

Билет 2

1. Методология разработки через тестирование (TDD): этапы цикла Red-Green-Refactor, преимущества и ограничения подхода.

2. Используя подход TDD, спроектируйте набор модульных тестов и напишите реализацию функции валидации пароля по требованиям ТЗ (длина от 8 до 32 символов, наличие заглавной буквы, цифры и спецсимвола).

Билет 3

1. Утверждения (Matchers) в модульном тестировании: проверка примитивов, глубокое сравнение объектов, работа с массивами, строками и числами с плавающей точкой.
2. Напишите модульные тесты для функции фильтрации и сортировки массива объектов каталога товаров, проверив корректность структуры возвращаемых объектов и граничные случаи (пустой массив, отсутствие совпадений).

Билет 4

1. Тестирование обработки ошибок и исключений в программных модулях. Параметризованные модульные тесты (test.each).
2. Разработайте параметризованный модульный тест для функции банковского перевода между счетами, проверяющий корректное выбрасывание исключений при недостатке средств, отрицательной сумме и заблокированном счёте.

Билет 5

1. Проблема связности кода и изоляция модулей при тестировании. Классификация тестовых двойников: Dummy, Stub, Spy, Mock, Fake.
2. Проведите рефакторинг предложенного модуля оформления заказа (жестко привязанного к внешнему сервису отправки SMS и БД) с применением внедрения зависимостей (Dependency Injection) и напишите для него изолированный Unit-тест.

Билет 6

1. Использование шпионов (Spies) и моков (Mocks) для контроля вызовов внешних функций, их аргументов и возвращаемых значений.
2. Напишите модульный тест для сервиса регистрации пользователя, замокав модуль отправки Email-уведомлений и проверив, что функция отправки вызывается ровно один раз с корректным адресом получателя.

Билет 7

1. Модульное тестирование асинхронного кода: работа с промисами (Promises), синтаксисом async/await и перехват асинхронных ошибок.
2. Разработайте модульные тесты для асинхронного модуля получения профиля пользователя по REST API, замокав сетевой клиент (fetch/axios) для сценариев успешного ответа (200 OK), ошибки 404 и сбоя сети.

Билет 8

1. Управление временем в модульных тестах: тестирование отложенных функций, таймеров (setTimeout, setInterval) и дат с помощью Fake Timers.
2. Напишите модульный тест для функции автосохранения черновика формы с задержкой (Debounce 2000 мс), используя имитацию таймеров без реального ожидания времени.

Билет 9

1. Метрики покрытия кода тестами (Code Coverage): Statement, Branch, Function, Line Coverage. Анализ непокрытых ветвлений.
2. Проанализируйте предложенный программный модуль со сложной условной логикой (if/else, switch, тернарные операторы), рассчитайте количество тестов для достижения 100% покрытия ветвлений (Branch Coverage) и напишите эти тесты.

Билет 10

1. Модульное и компонентное тестирование пользовательских интерфейсов веб-приложений (рендеринг в виртуальном DOM, поиск элементов, проверка состояния).
2. Напишите компонентный тест для интерактивного модуля «Счётчик корзины товаров», проверяющий начальное отображение, увеличение значения при клике на «+» и блокировку кнопки «-» при значении 0.

Билет 11

1. Тестирование веб-форм и пользовательского ввода на уровне компонентов. Снимочное тестирование (Snapshot Testing).
2. Разработайте компонентный тест для формы входа в систему, имитирующий ввод некорректного Email, отправку формы и проверяющий появление сообщения об ошибке валидации и блокировку кнопки отправки.

Билет 12

1. Регламент по безопасности веб-приложений и методика проведения аудита безопасности программных модулей по стандарту OWASP Top 10.
2. Проведите аудит безопасности предложенного модуля генерации HTML-шаблона профиля пользователя, выявите уязвимость межсайтового скриптинга (XSS), исправьте код и напишите модульный тест безопасности, подтверждающий экранирование вредоносных тегов.

Билет 13

1. Проектирование модульных тестов безопасности (Security Unit Tests) для защиты от инъекций (SQL/NoSQL) и уязвимостей регулярных выражений (ReDoS).
2. Проанализируйте модуль парсинга и валидации входных параметров API-запроса, составьте набор вредоносных тестовых векторов (Payloads) и напишите модульные тесты, проверяющие отклонение инъекций и некорректных типов данных.

Билет 14

1. Аудит безопасности модулей аутентификации, хэширования паролей, работы с JWT-токенами и проверки прав доступа (RBAC / IDOR).
2. Напишите набор модульных тестов для серверного Middleware-модуля проверки авторизации, тестирующий доступ с валидным JWT-токеном, просроченным токеном, подделанной подписью и недостаточной ролью пользователя (403 Forbidden).

Билет 15

1. Статический анализ безопасности кода (SAST) и аудит сторонних зависимостей веб-приложения (npm audit, CVE-уязвимости) в соответствии с регламентом.
2. Проанализируйте отчёт сканера безопасности по зависимостям и исходному коду веб-приложения, классифицируйте найденные уязвимости по критичности и составьте регламентный отчёт с планом их устранения.

Билет 16

1. Организация технического сопровождения веб-приложений по ТЗ: интеграция модульных тестов и проверок безопасности в конвейер CI/CD.
2. Разработайте схему и конфигурацию этапа тестирования в CI-пайплайне (запуск линтера, модульных тестов, проверки порога Coverage и npm audit), блокирующего слияние кода при падении хотя бы одной проверки.

Билет 17

1. Диагностика сбоев при техническом сопровождении веб-приложений: анализ логов ошибок, локализация дефекта и написание воспроизводящих регрессионных тестов.
2. По приведённому логу критической ошибки (Stack Trace) в модуле оплаты локализуйте строку сбоя (TypeError: Cannot read properties of undefined), напишите модульный тест, воспроизводящий сбой, и исправьте код модуля.

Билет 18

1. Стратегии резервного копирования веб-приложений (полное, инкрементальное, дифференциальное) и целевые показатели восстановления (RPO и RTO).
2. Спроектируйте регламент и расписание автоматического резервного копирования кода, конфигурационных файлов и базы данных веб-приложения в соответствии с требованиями ТЗ.

Билет 19

1. Процедура аварийного восстановления (Disaster Recovery) веб-приложения после критического сбоя и проверка целостности системы.
2. Составьте пошаговый план аварийного восстановления веб-приложения из резервной копии на чистом сервере и чек-лист пост-аварийного тестирования работоспособности модулей.

Билет 20

1. Управление обновлениями при сопровождении веб-приложений: выпуск хотфиксов (Hotfixes), стратегии безопасного развёртывания и процедура отката версий (Rollback).
2. Разработайте регламент действий инженера сопровождения при обнаружении критической регрессии после установки обновления на продуктивный сервер: алгоритм отката версии (Rollback) и последующего разбора инцидента.