Приложение
к ОПОП СПО 09.02.11 Разработка и управление программным обеспечением

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
РЕСПУБЛИКИ ДАГЕСТАН «ТЕХНИЧЕСКИЙ КОЛЛЕДЖ ИМЕНИ Р.Н. АШУРАЛИЕВА»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

междисциплинарного курса МДК 02.06 Безопасность программного обеспечения

Специальность 09.02.11 Разработка и управление программным обеспечением

Квалификация программист

Махачкала, 2026г.

ОДОБРЕНО

предметной (цикловой) комиссией УГС 09.00.00. Информатика и вычислительная техника и 10.00.00 Информационная безопасность

Председатель П(Ц)К

 Ш.М. Мусаева

Протокол № 8 от 30 апреля 2026 г.

Фонд оценочных средств междисциплинарного курса МДК.02.06 Безопасность программного обеспечения разработан на основе требований:

- Приказа Минпросвещения России от 24.02.2025 N 138 "Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.11 Разработка и управление программным обеспечением" (Зарегистрировано в Минюсте России 31.03.2025 N 81696);
- Рабочей программы междисциплинарного курса МДК.02.06 Безопасность программного обеспечения

с учетом:

- Примерной образовательной программы «Профессионалитет» по специальности 09.02.11 Разработка и управление программным обеспечением, утвержденная протоколом ФУМО в системе СПО по УГПС 09.00.00 Информатика и вычислительная техника от 29.08.2025 № 6/2025, зарегистрирована в государственном реестре примерных образовательных программ – регистрационный номер 18/2025. Реквизиты утверждающего документа –приказ ФГБОУ ДПО ИРПО № 01-09-581/2025 от 13.10.2025;

Разработчик:

- Азизова Лилия Насруллаховна, к.п.н., преподаватель ГБПОУ РД «Технический колледж имени Р.Н. Ашуралиева»

© Азизова Лилия Насруллаховна, 2026

© ГБПОУ РД «Технический колледж имени Р.Н. Ашуралиева», 2026

СОДЕРЖАНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ 4
2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА 4
3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ....10
4. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ МЕЖДИСЦИПЛИНАРНОГО КУРСА 13
5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ....17
5.1. Типовые задания для оценки знаний и умений....17
5.2. Критерии оценивания 28
5.2.1. Критерии оценивания устного ответа 28
5.2.2. Критерии оценивания выполнения заданий на лабораторных и практических занятиях....28
5.2.3. Критерии оценивания тестовых заданий 29
5.2.4. Общая классификация ошибок 29
6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ....29
6.1. Вопросы к экзамену по МДК....29
6.2. Критерии оценивания ответов на экзамене....31
7. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ 32

1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (далее – ФОС) междисциплинарного курса МДК 02.06 Безопасность программного обеспечения является частью программы подготовки специалистов среднего звена специальности 09.02.11 Разработка и управление программным обеспечением.

ФОС позволяет оценить достижение, запланированных по междисциплинарному курсу, результатов обучения.

ФОС включают оценочные средства для проведения текущего контроля успеваемости и промежуточной аттестации по междисциплинарному курсу.

Текущий контроль успеваемости осуществляется с целью регулярного наблюдения за ходом поэтапного освоения обучающимися междисциплинарного курса, оптимизации управления образовательной деятельностью обучающихся, своевременной корректировки персональных образовательных результатов, обучающихся педагогическими средствами.

Текущему контролю успеваемости подлежат все обучающиеся, осваивающие междисциплинарный курс.

Текущий контроль проводится в пределах учебного времени, отведенного на изучение междисциплинарного курса традиционными и инновационными методами с использованием современных технологий.

Результаты текущего контроля успеваемости обучающихся в виде оценки в балльном выражении («5», «4», «3», «2») записываются в журнале учебных занятий.

Текущий контроль освоения обучающимися программного материала междисциплинарного курса может иметь следующие виды: оперативный и рубежный контроль.

Оперативный контроль проводится с целью объективной оценки качества освоения программы междисциплинарного курса, а также стимулирования учебной работы обучающихся, мониторинга результатов образовательной деятельности, подготовки к промежуточной аттестации и обеспечения максимальной эффективности учебно-воспитательного процесса.

Рубежный контроль является контрольной точкой и проводится с целью комплексной оценки уровня освоения программного материала.

Промежуточная аттестация по междисциплинарному курсу проводится с целью оценки уровня освоения теоретических знаний, умений, приобретенного практического опыта.

Формы и периодичность промежуточной аттестации по междисциплинарному курсу определяются учебным планом образовательной программы: экзамен в 7 семестре.

Экзамен проводится непосредственно после завершения освоения междисциплинарного курса, в сроки, установленные календарным учебным графиком. Экзамен проводится в день, освобожденный от других форм учебной нагрузки.

Экзаменационные вопросы и задания составляются на основе рабочей программы междисциплинарного курса. Экзаменационные вопросы и задания должны соответствовать проверяемым результатам обучения и доводятся до сведения обучающихся в течение первых двух месяцев от начала обучения.

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА

ПМ.02 Разработка и интеграция модулей программного обеспечения направлен на формирование общих и профессиональных компетенций.

Освоение междисциплинарного курса должно способствовать формированию общих компетенций:

– ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

– ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

– ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.

– ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Освоение междисциплинарного курса должно способствовать формированию профессиональных компетенций:

– ПК 2.2. Разрабатывать модули программного обеспечения.

– ПК 2.3. Выполнять интеграцию модулей и компонентов программного обеспечения.

– ПК 2.4. Выполнять тестирование и отладку программного обеспечения.

– ПК 2.5. Осуществлять документирование программных модулей программного обеспечения.

В результате освоения междисциплинарного курса обучающийся должен овладеть навыками:

– Создания модулей программного обеспечения на различных языках программирования.

– Отладки и тестирования разработанных модулей.

– Применения структурного и объектно-ориентированного программирования.

– Оптимизации кода и алгоритмов программных модулей для увеличения производительности.

– Мониторинга и анализа производительности приложений.

– Интеграции программных модулей и компонентов в единое программное решение.

– Работы с API и веб-сервисами для взаимодействия между модулями.

– Работы с интеграционными платформами и инструментами.

– Обеспечения совместимости и стабильности системы.

– Отладки программного обеспечения на уровне программных модулей.

– Тестирования программного обеспечения.

– Формирования тестовых сценариев.

– Подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости).

– Оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения.

– Настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции.

– Формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами.

– Выполнения тестовых процедур на тестовых данных.

– Создания технической документации для модулей.

– Документирования кода, API и интерфейсов.

– Работы со специализированным ПО по документированию программного кода.

В результате освоения междисциплинарного курса обучающийся должен **уметь:**

Код умения	**Название умения**
У1	Разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий.

У2	Применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей.
У3	Анализировать требования и определять функциональность модуля.
У4	Создавать интерфейсы для взаимодействия с другими модулями и системами.
У5	Обеспечивать безопасность, производительность и масштабируемость при разработке модулей.
У6	Оптимизировать проектируемые модули для повышения их эффективности и качества.
У7	Работать с системой контроля версий.
У8	Улучшать производительность модулей, выявляя и устраняя узкие места.
У9	Проводить анализ и мониторинг производительности приложений.
У10	Применять инструменты для рефакторинга и оптимизации программного кода.
У11	Интегрировать модули и компоненты, обеспечивая их взаимодействие.
У12	Работать с API и устанавливать соединения между компонентами.
У13	Отслеживать и устранять конфликты и ошибки интеграции.
У14	Анализировать и определять зависимости между модулями и компонентами.
У15	Работать с различными форматами данных и протоколами передачи данных.
У16	Анализировать требования к программному обеспечению и составлять планы тестирования.
У17	Создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям.
У18	Выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования.
У19	Анализировать результаты тестирования и документировать найденные ошибки.
У20	Разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении.
У21	Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования.
У22	Использовать системы контроля дефектов ПО.
У23	Составлять отчет о выполнении тестирования ПО.
У24	Описывать функциональность модулей в документации.
У25	Создавать диаграммы для иллюстрации работы модулей.
У26	Программировать с использованием комментариев для документирования кода.
У27	Использовать специальные метки/теги для отметки важных частей кода в документации.
У28	Вести журнал изменений и фиксировать обновления программных модулей.
У29	Разбивать модули на логические блоки и описывать каждый блок отдельно.
У30	Включать в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки.
У31	Проводить регулярное обновление документации при изменении модулей или добавлении нового функционала.

В результате освоения междисциплинарного курса обучающийся должен **знать:**

Код знания	**Название знания**
З1	Язык программирования, основные конструкции, синтаксис.
З2	Паттерны проектирования.
З3	Структуры данных.
З4	Принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP.
З5	Работа с инструментальным программным обеспечением.
З6	Методы оптимизации кода и алгоритмов.
З7	Эффективные алгоритмы и структуры данных для повышения производительности.
З8	Многопоточность в программных модулях.
З9	Методы оптимизации сетевых протоколов для ускорения обмена данными.
З10	Кэширование данных.
З11	Управление памятью.
З12	Техники повышения производительности программного обеспечения.

313	Общих принципов функционирования аппаратных, программных и программно-аппаратных средств администрируемой информационно-коммуникационной системы.
314	Международных стандартов локальных вычислительных сетей.
315	Методы и подходы к интеграции модулей и компонентов.
316	Принципы версионирования и управления изменениями при интеграции.
317	Принципы безопасности при интеграции модулей и компонентов.
318	Принципы и методы тестирования программного обеспечения.
319	Основы программирования и архитектуры программного обеспечения.
320	Основы баз данных и SQL-запросов.
321	Инструменты для автоматизации тестирования.
322	Основы разработки и отладки программного обеспечения на разных языках программирования.
323	Понятие дефекта программного обеспечения.
324	Критерии качества ПО.
325	Виды и типы тестирования ПО.
326	Техники ручного тестирования.
327	Техники автоматизированного тестирования.
328	Жизненный цикл дефекта ПО.
329	Принципы работы в системе контроля дефектов.
330	Основные понятия о качестве ПО.
331	Стандарты технической документации.
332	Принципы документирования программного обеспечения.
333	Инструменты для создания технической документации и комментирования

Общие компетенции:

Код компетенции	**Формулировка компетенции**	**Знания, умения**
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	**Умения:** распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять ее составные части; определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; владеть актуальными методами работы в профессиональной и смежных сферах; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)
		Знания: актуальный профессиональный и социальный контекст, в котором приходится работать и жить; структура плана для решения задач, алгоритмы выполнения работ в профессиональной и смежных областях; основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте; методы работы в профессиональной и смежных сферах; порядок оценки результатов решения задач профессиональной деятельности

ОК 02	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	**Умения:** определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации; выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска; оценивать практическую значимость результатов поиска; применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение в профессиональной деятельности; использовать различные цифровые средства для решения профессиональных задач
		Знания: номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации; современные средства и устройства информатизации, порядок их применения; программное обеспечение в профессиональной деятельности, в том числе цифровые средства; психологические основы деятельности коллектива
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	**Умения:** грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке проявлять толерантность в рабочем коллективе
		Знания: правила построения устных сообщений; особенности социального и культурного контекста
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках	**Умения:** понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснять свои действия (текущие и планируемые) писать простые связные сообщения на знакомые или интересующие профессиональные темы
		Знания: правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности

Профессиональные компетенции:

Код и формулировка компетенции	**Показатели освоения компетенции**
ПК 2.2. Разрабатывать модули программного обеспечения	**Навыки:** Создания модулей программного обеспечения на различных языках программирования. Отладки и тестирования разработанных модулей. Применения структурного и объектно-ориентированного программирования. Оптимизации кода и алгоритмов программных модулей для увеличения производительности. Мониторинга и анализа производительности приложений.
	Умения: Разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий. Применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей. Анализировать требования и определять функциональность модуля. Создавать интерфейсы для взаимодействия с другими модулями и системами. Обеспечивать безопасность, производительность и масштабируемость при разработке модулей.

	Оптимизировать проектируемые модули для повышения их эффективности и качества; Работать с системой контроля версий. Улучшать производительность модулей, выявляя и устраняя узкие места. Проводить анализ и мониторинг производительности приложений. Применять инструменты для рефакторинга и оптимизации программного кода.
	Знания: Язык программирования, основные конструкции, синтаксис. Паттерны проектирования. Структуры данных. Принципы создания интерфейсов для взаимодействия с другими модулями и системами, таких как REST API, SOAP. Работу с инструментальным программным обеспечением. Методы оптимизации кода и алгоритмов. Эффективные алгоритмы и структуры данных для повышения производительности; Многопоточность в программных модулях. Методы оптимизации сетевых протоколов для ускорения обмена данными; Кэширование данных. Управление памятью. Техники повышения производительности программного обеспечения.
ПК 2.3. Выполнять интеграцию модулей и компонентов программного обеспечения	**Навыки:** Интеграции программных модулей и компонентов в единое программное решение. Работы с API и веб-сервисами для взаимодействия между модулями. Работы с интеграционными платформами и инструментами. Обеспечения совместимости и стабильности системы.
	Умения: Интегрировать модули и компоненты, обеспечивая их взаимодействие. Работать с API и устанавливать соединения между компонентами. Отслеживать и устранять конфликты и ошибки интеграции. Анализировать и определять зависимости между модулями и компонентами. Работать с различными форматами данных и протоколами передачи данных.
	Знания: Общие принципы функционирования аппаратных, программных и программно-аппаратных средств администрируемой информационно-коммуникационной системы. Международные стандарты локальных вычислительных сетей. Методы и подходы к интеграции модулей и компонентов. Принципы версионирования и управления изменениями при интеграции. Принципы безопасности при интеграции модулей и компонентов.
ПК 2.4. Выполнять тестирование и отладку программного обеспечения	**Навыки:** Отладки программного обеспечения на уровне программных модулей. Тестирования программного обеспечения. Формирования тестовых сценариев. Подготовки тестовых платформ (установка операционной системы, дополнительного ПО и другого по необходимости). Оценки объема тестирования ПО с целью определения необходимых ресурсов для его выполнения. Настройки тестовой среды и аппаратных средств для выполнения тестирования ПО в соответствии с заданием на тестирование в пределах своей компетенции. Формирования и представления отчетности о подготовке к выполнению задания на тестирование ПО в соответствии с установленными регламентами. Выполнения тестовых процедур на тестовых данных.
	Умения: Анализировать требования к программному обеспечению и составлять планы тестирования. Создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям. Выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования. Анализировать результаты тестирования и документировать найденные ошибки. Разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении. Выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования. Использовать системы контроля дефектов ПО. Составлять отчет о выполнении тестирования ПО.

	Знания: Принципы и методы тестирования программного обеспечения. Основы программирования и архитектуры программного обеспечения. Основы баз данных и SQL-запросов. Инструменты для автоматизации тестирования. Основы разработки и отладки программного обеспечения на разных языках программирования. Понятие дефекта программного обеспечения. Критерии качества ПО. Виды и типы тестирования ПО. Техники ручного тестирования. Техники автоматизированного тестирования. Жизненный цикл дефекта ПО. Принципы работы в системе контроля дефектов. Основные понятия о качестве ПО.
ПК 2.5. Осуществлять документирование программных модулей программного обеспечения.	**Навыки:** Создания технической документации для модулей. Документирования кода, API и интерфейсов. Работы со специализированным ПО по документированию программного кода.
	Умения: Описывать функциональность модулей в документации. Создавать диаграммы для иллюстрации работы модулей. Программировать с использованием комментариев для документирования кода. Использовать специальные метки/теги для отметки важных частей кода в документации. Вести журнал изменений и фиксировать обновления программных модулей. Разбивать модули на логические блоки и описывать каждый блок отдельно. Включать в документацию особенности модулей, такие как ограничения, уязвимости или оптимальные настройки. Проводить регулярное обновление документации при изменении модулей или добавлении нового функционала.
	Знания: Стандарты технической документации. Принципы документирования программного обеспечения. Инструменты для создания технической документации и комментирования кода.

3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ

Формы текущего контроля по междисциплинарному курсу:

- устный опрос (фронтальный, индивидуальный, комбинированный);
- тестирование (письменное или компьютерное);
- письменная проверка (ответы на вопросы, решение задач и примеров, составление тезисов, рефератов, выполнение схем, выполнение заданий для самостоятельной работы и др.);
- практическая проверка (при проведении практических занятий));
- самоконтроль и взаимопроверка.

Возможны и другие формы текущего контроля успеваемости, в том числе инновационные на основе информационно-коммуникационных технологий.

Преподаватель на одном учебном занятии может использовать одну или несколько форм текущего контроля.

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	**Критерии оценки результата (показатели освоенности компетенций)**	**Формы контроля и методы оценки**

ПК 2.2. Разрабатывать модули программного обеспечения	Разрабатывать модули программного обеспечения с использованием различных языков программирования и технологий применять паттерны проектирования и структуры данных для создания эффективных и масштабируемых модулей анализировать требования и определять функциональность модуля создавать интерфейсы для взаимодействия с другими модулями и системами обеспечивать безопасность, производительность и масштабируемость при разработке модулей оптимизировать проектируемые модули для повышения их эффективности и качества работать с системой контроля версий улучшать производительность модулей, выявляя и устраняя узкие места проводить анализ и мониторинг производительности приложений применять инструменты для рефакторинга и оптимизации программного кода	Экспертная оценка деятельности в ходе выполнения практических работ, практической подготовки, интерпретация результатов собеседования и наблюдения, решение производственных задач. Текущий контроль: - защита отчетов по практическим работам; - оценка заданий для самостоятельной работы Промежуточная аттестация: - экспертная оценка выполнения практических заданий на экзамене по МДК;
ПК 2.3. Выполнять интеграцию модулей и компонентов программного обеспечения	Анализировать и определять зависимости между модулями и компонентами работать с различными форматами данных и протоколами передачи данных	
ПК 2.4. Выполнять тестирование и отладку программного обеспечения	Анализировать требования к программному обеспечению и составлять планы тестирования. создавать тестовые сценарии и тест-кейсы для проверки функциональности и соответствия требованиям. выполнять тестирование программного обеспечения вручную и автоматизировать процесс тестирования. анализировать результаты тестирования и документировать найденные ошибки. разрабатывать стратегии отладки и исправлять ошибки в программном обеспечении. выполнять модульные тесты с использованием инструментов тестирования, в том числе автоматизированного тестирования использовать системы контроля дефектов ПО составлять отчет о выполнении тестирования ПО	
ПК 2.5. Осуществлять документирование программных модулей программного обеспечения	Использование выбранной системы контроля версий. Использование методов для получения кода с заданной функциональностью и степенью качества. Выявление ошибок в системных компонентах на основе спецификаций.	
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к	Обоснованность планирования учебной и профессиональной деятельности; соответствие результата выполнения профессиональных задач эталону (стандартам, образцам, алгоритму, условиям, требованиям или ожидаемому результату);	

различным контекстам	степень точности выполнения поставленных задач.	
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	полнота охвата информационных источников; скорость нахождения и достоверность информации; обновляемость и пополняемость знаний, влияющих на результаты учебной и производственной деятельности.	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	демонстрация навыков грамотно общения и оформление документации на государственном языке Российской Федерации, принимая во внимание особенности социального и культурного контекста	
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках	демонстрация умений понимать тексты на базовые и профессиональные темы; - составлять необходимую документацию на государственном и иностранном языках	

4. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ МЕЖДИСЦИПЛИНАРНОГО КУРСА

МДК 02.06 Безопасность программного обеспечения

№	Наименование темы	Результаты обучения (освоенныеумения и знаний)	ПК, ОК	Текущий контроль успеваемости	Промежуточная аттестация
	МДК 02.06 Безопасность программного обеспечения				
	Раздел 6. Безопасность программного обеспечения				
	Тема 6.1. Основы безопасности программного обеспечения				
	Введение в кибербезопасность и уязвимости ПО	323, 324, 330	ОК 01, 05, 09 ПК 2.2, 2.3, 2.4, 2.5	Устный опрос, тестирование	Экзамен
	Модели угроз и анализ рисков	315, 317		Устный опрос, тестирование	
	Уязвимости веб-приложений: OWASP Top 10	34, 317, 320		Устный опрос, тестирование	
	Безопасная аутентификация и авторизация	34, 317		Устный опрос, тестирование	
	Криптография для разработчиков	31-33, 35- 312, 322		Устный опрос, тестирование	
	Безопасность цепочки поставок (Supply Chain Security): уязвимости в зависимостях (dependency confusion, typosquatting), анализ SBOM (Software Bill of Materials), инструменты (Snyk, Dependabot).	35, 315, 316, 333		Устный опрос, тестирование	
	Безопасная работа с памятью и данными: переполнение буфера, утечки памяти, защита от инъекций на уровне ОС	33, 36, 37, 311, 312		Устный опрос, тестирование	
	Основы безопасности баз данных: SQL-инъекции, NoSQL-инъекции, шифрование на уровне столбцов, маскировка данных, аудит доступа	34, 317, 320		Устный опрос, тестирование	
	DevSecOps и безопасность CI/CD пайплайнов: внедрение SAST/DAST/IAST в пайплайн, безопасное хранение секретов, сканирование контейнеров на этапе сборки	35, 318, 321, 325-327, 333		Устный опрос, тестирование	
	Управление уязвимостями и патч-менеджмент: CVE/CPE/CVSS, приоритизация исправлений, эксплуатация vs исправление, жизненный цикл уязвимости	323, 324, 328- 330		Устный опрос, тестирование	
	Анализ кода на наличие уязвимостей - ручной review 1000 строк кода. Составление модели угроз для типового веб-приложения	317, 323, 324, 330-333; У3, У5, У16, У19, У20, У24-У31		Отчет по практическому занятию	
	SQL инъекции - эксплуатация и защита уязвимого приложения. XSS атаки - создание и предотвращение межсайтового скриптинга	34, 317, 320, 323-327, 330; У1-, У5, У15, У18, У20- У22		Отчет по практическому занятию	
	CSRF защита - реализация токенов и проверки Origin/Referer. Настройка CORS политик для веб-приложений	34, 39, 314, 317; У4, У5, У12, У15		Отчет по практическому занятию	
	Настройка безопасной аутентификации с JWT и refresh токенами. Реализация RBAC системы с разделением привилегий	32, 33, 34, 317; У4, У5, У11, У12, У14		Отчет по практическому занятию	

	Шифрование данных с использованием AES и RSA. Хэширование паролей с salt и adaptive functions (bcrypt, Argon2)	31-, 33, 36, 37, 311, 312, 317; У1, У2, У5, У6, У8, У10		Отчет по практическому занятию	
	Анализ сетевого трафика с помощью Wireshark. Сканирование уязвимостей OWASP ZAP и Burp Suite. Аудит логов безопасности и выявление подозрительной активности	35, 39, 313, 314, 318, 321, 323, 324, 330; У9, У15, У19, У22, У23		Отчет по практическому занятию	
	Настройка HTTPS и создание самоподписанных сертификатов. Защита от brute-force атак с ограничением попыток входа. Защита от DDOS атак с помощью rate limiting	39, 310, 313, 314, 317; У5, У6, У9, У12, У15		Отчет по практическому занятию	
	Безопасная работа с файлами. Реализация безопасной десериализации данных. Безопасная работа с памятью в приложениях	33, 36, 37, 311, 317, 322; У1, У5, У6, У8, У10, У20		Отчет по практическому занятию	
	Создание безопасного API с валидацией всех входных данных	32, 34, 317, 318, 320, 321; У4, У5, У12, У15, У18, У20		Отчет по практическому занятию	
	Тема 6.2. Разработка безопасного ПО и прикладная криптография				
	Принципы безопасного проектирования архитектуры	31- 33, 36-312, 319, 322, 331- 333	ОК 01, 05, 09 ПК 2.2, 2.3, 2.4, 2.5	Устный опрос, тестирование	
	Криптографические протоколы и их реализация	31- 33, 35- 312		Устный опрос, тестирование	
	Криптография в мобильных приложениях	31- 33, 38, 311		Устный опрос, тестирование	
	Криптография в веб-приложениях	31- 34, 310		Устный опрос, тестирование	
	Криптография в облачных средах	35, 313- 315		Устный опрос, тестирование	
	Безопасность REST API и GraphQL: rate limiting, validation, массовое присвоение (mass assignment), перебор ID (IDOR), безопасная документация (OpenAPI), управление версиями	34, 39, 310, 316, 317		Устный опрос, тестирование	
	Безопасность микросервисной архитектуры: mTLS между сервисами, JWT Best Practices, контроль доступа в сетке сервисов (Istio/Linkerd), обход прокси-атак	32, 34, 38, 39, 315, 317		Устный опрос, тестирование	
	Постквантовая криптография и гибридные схемы: алгоритмы NIST (Kyber, Dilithium), миграция с RSA/ECDSA, гибридные handshake в	31- 33, 37		Устный опрос, тестирование	
	Управление криптографическими ключами на практике: генерация, ротация, отзыв ключей, HSM (аппаратные модули), иерархия ключей (KEK/DEK), секреты в распределенных системах	35, 311, 313, 317		Устный опрос, тестирование	
	Безопасность контейнеров и оркестрации PodSecurity, защита etcd, сканирование образов, runtime-безопасность	35, 38, 313- 315, 317		Устный опрос, тестирование	
	Реализация end-to-end шифрования для мессенджера на Signal	31- 33, 36-38, 311, 312, 317;		Отчет по практическому занятию	

		У1, У2, У5, У6, У8, У12			
	Настройка TLS 1.3 с perfect forward secrecy и современными cipher сертификатов	36, 37, 39, 313, 314, 317, 319; У5, У6, У9, У12, У15		Отчет по практическому занятию	
	Создание secure OAuth 2.0 провайдера с PKCE и защитой от атак. Имплементация JWE (JSON Web Encryption) для защищенных токенов. Разработка secure API gateway с JWT verification и rate limiting. Разработка secure session management с защитой от hijacking	32- 35, 315, 316, 317; У4, У5, У11, У12, У14, У15		Отчет по практическому занятию	
	Разработка безопасного voting system с homomorphic encryption. Имплементация zero-knowledge proof для аутентификации без пароля. Реализация secure multi-party computation для совместных вычислений. Настройка quantum-resistant cryptography с lattice-based алгоритмами	31- 33, 36- 38, 311, 312, 317; У1, У2, У5, У6, У8, У12		Отчет по практическому занятию	
	Создание cryptocurrency wallet с ECDSA и hierarchical deterministic keys. Создание blockchain smart contract с защитой от reentrancy attacks	31- 33, 36- 38, 311, 312, 317; У1, У2, У5, У6, У8, У12		Отчет по практическому занятию	
	Реализация secure password manager с client-side encryption	31- 33, 36, 37, 311, 312, 317; У1, У2, У5, У6, У8, У12		Отчет по практическому занятию	
	Настройка HSM эмулятора для аппаратной защиты ключей. Создание	35, 311, 313, 314, 317; У1, У5, У6, У11, У12		Отчет по практическому занятию	
	Разработка secure file storage с encryption at rest и in transit	33, 36, 37, 311, 313-315, 317; У1, У5, У6, У12, У15		Отчет по практическому занятию	
	Имплементация digital signature system с timestamping. Создание cryptographically secure RNG (random number generator)	31- 33, 36, 37, 311, 312, 317; У1, У2, У5, У6, У8, У12		Отчет по практическому занятию	
	Самостоятельная работа обучающихся:				
	Изучить теоретический материал и составить тезисы (краткий конспект) по следующим темам:				
	Анализ реальных киберинцидентов за последние 3 года.	313, 314, 317, 318, 323- 325, 330	ОК 01, 05, 09 ПК 2.2, 2.3, 2.4, 2.5	Конспект, устный опрос	
	Построение модели угроз для вымышленного интернет-магазина.	34, 313, 314, 317, 319, 320, 323, 324, 330			
	Детальный разбор трех уязвимостей из OWASP Top 10 с примерами кода.	31, 34, 317, 320, 322-324, 330			

	Сравнительный анализ потоков аутентификации (OAuth 2.0 vs OpenID	34, 315, 317, 319			
	Исследование уязвимостей цепочки поставок.	35, 315- 317, 333			
	Настройка политик безопасности для CI/CD.	35, 317, 318, 321, 325- 327, 333			
	Оценка безопасности архитектуры на основе модели Zero Trust.	34, 313- 315, 317, 319			
	Сравнение реализации TLS 1.2 и TLS 1.3.	36, 37, 39, 313, 314, 317, 319			
	Анализ уязвимостей мобильной криптографии.	31- 33, 38, 311, 317, 322			
	Исследование алгоритмов постквантовой криптографии.	31- 33, 36, 37, 312, 317, 322			
	Проектирование политики безопасности для Kubernetes.	35, 313- 315, 317, 333			
	Оценка зрелости безопасной разработки по модели OWASP SAMM	35, 317, 318, 321, 323-327, 330-333			

5. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

МДК 02.06 Безопасность программного обеспечения

5.1.Типовые задания для оценки знаний и умений

Раздел 6. Безопасность программного обеспечения

Тема 6.1. Основы безопасности программного обеспечения

1. Задание для устного опроса по темам

1. Перечислите основные принципы построения защищенных систем (например, defense in depth, least privilege, zero trust). Как эти принципы влияют на выбор архитектурных решений на этапе проектирования? Приведите примеры их применения в распределенных системах.
2. Опишите жизненный цикл защищенного сеанса TLS: установление соединения, согласование параметров, аутентификация сервера и клиента, обмен ключами. Какие уязвимости могут возникнуть при некорректной реализации протокола (например, неправильная проверка сертификатов, использование слабых шифров)?
3. Какие особенности использования криптографии в мобильных приложениях (ограниченные ресурсы, защита ключей в хранилище, работа в офлайн-режиме)? Как обеспечить безопасное хранение ключей на устройствах Android и iOS (Keystore, Keychain) и какие риски существуют при извлечении ключей из откомпилированного кода?
4. Какие криптографические механизмы применяются для защиты данных в веб-приложениях на стороне клиента и сервера? Опишите использование HTTPS, шифрования cookies, подписи токенов (JWT) и шифрования данных в браузере (Web Crypto API). Какие атаки могут быть направлены на эти механизмы?
5. Какие проблемы возникают при использовании криптографии в облачных окружениях (управление ключами, доверенная среда выполнения, шифрование данных на стороне провайдера)? Опишите подходы к шифрованию данных в покое и в транзите, а также использование сервисов управления ключами (KMS) и Hardware Security Modules (HSM) в облаке.
6. Какие угрозы для REST API предотвращает ограничение частоты запросов? Как реализовать валидацию входных данных, чтобы избежать инъекций и некорректных данных? Что такое массовое присвоение (mass assignment) и как защититься от него (например, с помощью DTO или явного разрешенных полей)?
7. Что такое уязвимость IDOR и как ее предотвратить (использование косвенных ссылок, проверка прав доступа)? Какие аспекты безопасности важно учесть при публикации документации OpenAPI (скрытие чувствительных эндпоинтов, ограничение доступа к спецификации)? Как управление версиями API может влиять на безопасность (обратная совместимость, устаревшие уязвимые версии)?
8. В чем отличие безопасности GraphQL от REST? Опишите специфические уязвимости GraphQL (например, глубокие вложенные запросы, перегрузка ресурсов, интроспекция). Какие методы защиты применяются: depth limiting, query cost analysis, отключение интроспекции в production?
9. Как взаимная аутентификация по TLS (mTLS) обеспечивает безопасность

взаимодействия между микросервисами? Какие рекомендации по использованию JWT (короткоживущие токены, проверка аудитории, управление отзывом) следует соблюдать в микросервисной среде?

10. Как инструменты сервисной сетки (service mesh) усиливают безопасность микросервисов (авторизация на уровне сети, управление политиками)? Что такое атака «обход прокси» (bypass proxy) и как ее предотвратить при использовании sidecar-контейнеров?
11. Каковы основные угрозы квантовых компьютеров для современных криптосистем? Опишите финалистов NIST (Kyber для KEM, Dilithium для подписей). Как организовать миграцию с RSA/ECDSA на постквантовые алгоритмы без потери совместимости (гибридные схемы, двойные подписи)?
12. Опишите полный жизненный цикл ключей: генерация (где и как), безопасное хранение, периодическая ротация, отзыв при компрометации. Какую роль играют аппаратные модули безопасности (HSM) и облачные KMS в управлении ключами? Приведите примеры политик ротации.
13. Что такое иерархия ключей: ключ шифрования данных (DEK) и ключ шифрования ключей (KEK)? Как она применяется для масштабируемого и безопасного управления секретами в распределенных системах (например, в Kubernetes с использованием внешних провайдеров секретов)?
14. Какие риски несут привилегированные контейнеры и как их ограничить с помощью политик PodSecurity в Kubernetes? Почему важно защищать хранилище etcd (шифрование, контроль доступа) и как это реализовать?
15. Опишите подходы к обеспечению безопасности на протяжении всего жизненного цикла контейнера: сканирование образов на уязвимости (в CI/CD), использование минимальных базовых образов, мониторинг runtime-поведения (Falco, Sysdig) и защита от выполнения вредоносного кода.

Тестовые вопросы

Тема 6.1. Основы безопасности программного обеспечения

Вариант 1

1. Какой принцип безопасности предполагает, что доступ к ресурсам должен предоставляться только в минимально необходимом объеме?
А) Защита в глубину
Б) Принцип наименьших привилегий
В) Zero Trust
Г) Разделение обязанностей

2. Что из перечисленного является уязвимостью при использовании JWT?
А) Подпись токена с использованием асимметричного ключа
Б) Хранение токена в localStorage
В) Использование короткого времени жизни (exp)
Г) Проверка аудитории (aud)

3. Как называется атака, при которой злоумышленник подменяет идентификатор объекта в запросе, чтобы получить доступ к чужому ресурсу?
А) SQL-инъекция
Б) IDOR

В) XSS
Г) CSRF

4. Какой алгоритм постквантовой криптографии выбран NIST для шифрования с открытым ключом (KEM)?

А) Dilithium
Б) Falcon
В) Kyber
Г) SPHINCS+

5. Что такое mTLS?

А) Односторонняя аутентификация сервера
Б) Взаимная аутентификация клиента и сервера с использованием сертификатов
В) Протокол туннелирования данных
Г) Метод шифрования симметричным ключом

6. Какая из перечисленных мер защиты GraphQL позволяет предотвратить перегрузку сервера глубокими вложенными запросами?

А) Отключение интроспекции
Б) Ограничение глубины запроса (depth limiting)
В) Использование rate limiting
Г) Валидация входных данных

7. Что является основным недостатком использования привилегированных контейнеров в Kubernetes?

А) Увеличение потребления памяти
Б) Возможность получения доступа к хостовой системе и обхода изоляции
В) Проблемы с сетевым взаимодействием
Г) Снижение производительности дисковых операций

8. Какой компонент Kubernetes хранит все данные о кластере и требует особой защиты?

А) kube-apiserver
Б) etcd
В) kube-scheduler
Г) kubelet

9. Что такое KEK (Key Encryption Key) в иерархии ключей?

А) Ключ, используемый для шифрования данных
Б) Ключ, используемый для шифрования других ключей
В) Ключ, используемый для аутентификации
Г) Ключ, используемый для подписи

10. Какой механизм защиты REST API от массового присвоения (mass assignment) является наиболее эффективным?

А) Использование DTO (Data Transfer Objects) с явным перечислением разрешенных полей
Б) Валидация всех полей запроса
В) Ограничение размера запроса
Г) Использование Content-Type: application/json

11. Какая проблема возникает при использовании криптографии в мобильных приложениях, связанная с извлечением ключей?

А) Ключи могут быть извлечены из откомпилированного кода или памяти, если не используются аппаратные хранилища
Б) Ключи невозможно надежно сгенерировать на мобильном устройстве
В) Ключи автоматически удаляются при обновлении приложения
Г) Ключи не поддерживают алгоритмы с открытым ключом

12. Какая атака может быть предотвращена с помощью ограничения частоты запросов (rate limiting)?

А) XSS
Б) Brute-force паролей
В) SQL-инъекция
Г) CSRF

13. Что такое гибридная схема в контексте постквантовой криптографии?

А) Использование одного алгоритма для всех целей
Б) Комбинация классического и постквантового алгоритмов для обеспечения совместимости и

безопасности
В) Использование только постквантовых алгоритмов
Г) Использование симметричного и асимметричного шифрования одновременно
14. Какой инструмент используется для runtime-безопасности контейнеров (мониторинг аномального поведения)?
А) Docker Bench
Б) Falco
В) Trivy
Г) Kaniko
15. Какое требование безопасности при проектировании архитектуры подразумевает, что каждая операция должна запрашивать разрешение на доступ к ресурсам, а не полагаться на предыдущие проверки?
А) Defense in depth
Б) Least privilege
В) Zero Trust
Г) Separation of duties

Вариант 2

1. Какой принцип безопасности предполагает использование нескольких уровней защиты, чтобы компенсировать возможные уязвимости?
А) Минимальные привилегии
Б) Защита в глубину
В) Zero Trust
Г) Разделение обязанностей
2. Какая из перечисленных практик является рекомендацией по безопасному использованию JWT?
А) Хранить токен в localStorage без защиты
Б) Использовать короткое время жизни (exp) и реализовать механизм обновления
В) Передавать токен в URL-параметрах
Г) Использовать алгоритм шифрования без подписи
3. Как защититься от атаки IDOR при проектировании REST API?
А) Использовать прямые ссылки на объекты, полученные от клиента
Б) Проверять права доступа пользователя к запрашиваемому объекту на сервере
В) Шифровать все идентификаторы
Г) Использовать только POST-запросы
4. Какой алгоритм постквантовой криптографии выбран NIST для цифровых подписей (из представленных)?
А) Kyber
Б) Dilithium
В) NTRU
Г) McEliece
5. Что такое JWT (JSON Web Token) и как он обычно используется в микросервисах?
А) Для хранения сессионных данных на сервере
Б) Для безсессионной аутентификации, содержащей утверждения о пользователе
В) Для шифрования данных между сервисами
Г) Для обмена симметричными ключами
6. Какую угрозу для GraphQL представляет возможность запрашивать информацию о схеме (интроспекция) в production?
А) Утечка структуры данных, что упрощает злоумышленнику поиск уязвимостей
Б) Перегрузка сервера
В) SQL-инъекции
Г) Mass assignment
7. Какое из перечисленных действий помогает защитить etcd в Kubernetes?
А) Шифрование данных и использование TLS для связи
Б) Размещение etcd на том же узле, что и kube-apiserver

В) Отключение аутентификации для быстродействия
Г) Регулярное удаление старых ключей

8. Какой подход к безопасности контейнеров подразумевает сканирование образов на наличие известных уязвимостей?

А) Runtime security
Б) Image scanning (например, Trivy, Clair)
В) Pod Security Policies
Г) Network policies

9. Что такое HSM (Hardware Security Module) и его основное назначение?

А) Программное хранилище ключей
Б) Аппаратное устройство для безопасного хранения и выполнения криптографических операций с ключами
В) Система управления трафиком
Г) Балансировщик нагрузки

10. Как называется процесс периодической замены криптографических ключей для минимизации риска компрометации?

А) Генерация
Б) Ротация
В) Отзыв
Г) Архивация

11. Какая особенность криптографии в облачных средах связана с управлением ключами?

А) Ключи всегда генерируются на стороне клиента
Б) Использование облачных KMS позволяет централизованно управлять ключами и политиками доступа
В) Ключи нельзя хранить в облаке по соображениям безопасности
Г) Облачные провайдеры не предоставляют HSM

12. Что такое массовое присвоение (mass assignment) и почему оно опасно?

А) Это возможность записать несколько полей объекта одновременно, что может привести к перезаписи защищенных полей (например, роли пользователя)
Б) Это атака на базу данных
В) Это метод оптимизации запросов
Г) Это способ валидации данных

13. Как обеспечить безопасную документацию API (OpenAPI) без раскрытия чувствительной информации?

А) Публиковать документацию только внутри сети, ограничить доступ, исключить эндпоинты администратора
Б) Включать все эндпоинты в документацию
В) Не использовать OpenAPI
Г) Шифровать документацию

14. Какое преимущество дает использование сервисной сетки (service mesh) в безопасности микросервисов?

А) Автоматическое масштабирование
Б) Централизованное управление политиками безопасности, включая mTLS и авторизацию
В) Ускорение сетевых запросов
Г) Отказ от аутентификации

15. Какие риски связаны с использованием устаревших версий API?

А) Устаревшие версии могут содержать неисправленные уязвимости и быть менее защищенными
Б) Они работают быстрее
В) Они не требуют аутентификации
Г) Они используют новые алгоритмы шифрования

КЛЮЧ К ТЕСТУ

Вариант 1		Вариант 2	
№ вопроса	Правильный ответ	№ вопроса	Правильный ответ
1	Б	1	Б
2	Б	2	Б

3	Б	3	Б
4	В	4	Б
5	Б	5	Б
6	Б	6	А
7	Б	7	А
8	Б	8	Б
9	Б	9	Б
10	А	10	Б
11	А	11	Б
12	Б	12	А
13	Б	13	А
14	Б	14	Б
15	В	15	А

Практические занятия № 1-9:

1. Анализ кода на наличие уязвимостей - ручной review 1000 строк кода. Составление модели угроз для типового веб-приложения
2. SQL инъекции - эксплуатация и защита уязвимого приложения. XSS атаки - создание и предотвращение межсайтового скриптинга
3. CSRF защита - реализация токенов и проверки Origin/Referer. Настройка CORS политик для веб-приложений
4. Настройка безопасной аутентификации с JWT и refresh токенами. Реализация RBAC системы с разделением привилегий
5. Шифрование данных с использованием AES и RSA. Хэширование паролей с salt и adaptive functions (bcrypt, Argon2)
6. Анализ сетевого трафика с помощью Wireshark. Сканирование уязвимостей OWASP ZAP и Burp Suite. Аудит логов безопасности и выявление подозрительной активности
7. Настройка HTTPS и создание самоподписанных сертификатов. Защита от brute-force атак с ограничением попыток входа. Защита от DDOS атак с помощью rate limiting
8. Безопасная работа с файлами. Реализация безопасной десериализации данных. Безопасная работа с памятью в приложениях
9. Создание безопасного API с валидацией всех входных данных

Тема 6.2. Разработка безопасного ПО и прикладная криптография

1. Задание для устного опроса по темам

1. Какие архитектурные решения помогают реализовать принцип "защита в глубину" (defense in depth) в распределенной системе? Приведите примеры сочетания сетевых, прикладных и криптографических мер защиты.
2. Опишите процесс согласования параметров (cipher suite negotiation) в TLS 1.3. В чем отличие от TLS 1.2? Какие атаки на протокол были устранены в новой версии?
3. Какие риски возникают при использовании самоподписанных сертификатов в мобильных приложениях? Как обеспечить безопасную проверку сертификатов при работе с собственными серверами?
4. Для каких целей в веб-приложениях используется шифрование на стороне клиента (Web Crypto API)? Какие сценарии (например, шифрование сообщений перед отправкой, генерация ключей) оправданы, а в каких случаях оно избыточно?
5. Как обеспечить конфиденциальность данных при использовании облачных баз данных (Database as a Service)? Опишите варианты шифрования на уровне приложения, на уровне базы данных и на уровне диска, а также их компромиссы.

6. Как реализовать адаптивное ограничение частоты запросов (adaptive rate limiting), учитывающее поведение пользователя (например, повышение лимита для авторизованных пользователей)? Как это влияет на безопасность и UX?
7. Предложите стратегию управления версиями API, которая позволяет одновременно поддерживать несколько версий, при этом безопасно прекращая поддержку устаревших (с учетом уязвимостей). Как информировать клиентов об изменениях без раскрытия избыточной информации?
8. Что такое атака "GraphQL introspection abuse" и как ограничить доступ к схеме без полного отключения интроспекции (например, для внутренних сред)? Какие еще GraphQL-специфичные угрозы существуют, помимо глубины запросов?
9. Как организовать управление сертификатами для mTLS в динамической среде с частыми перезапусками сервисов (например, с использованием автоматического ротации через SPIRE/Vault)? Как это влияет на архитектуру?
10. Что такое "pod-to-pod bypass" в Kubernetes при использовании сервисной сетки? Как настроить политики сети (NetworkPolicy) и sidecar-перехват, чтобы предотвратить прямой доступ к контейнерам в обход прокси?
11. Какие проблемы возникают при переходе на постквантовые алгоритмы с точки зрения производительности (размер ключей, время подписи, пропускная способность) и как их можно решить в реальных системах?
12. Опишите механизм гибридного обмена ключами в TLS. Как он обеспечивает совместимость с существующими реализациями и защиту от атак "захват сейчас, расшифровать позже" (harvest now, decrypt later)?
13. Как реализовать автоматическую ротацию ключей без прерывания работы сервиса? Приведите пример стратегии ротации для DEK (ключ шифрования данных) и KEK (ключ шифрования ключей) в распределенной системе.
14. Сравните подходы к управлению секретами с использованием внешних провайдеров (HashiCorp Vault, AWS Secrets Manager) и встроенных механизмов Kubernetes (Secrets). Какие преимущества и недостатки у каждого, особенно в контексте безопасности и удобства ротации?
15. Опишите комплексный подход к обеспечению безопасности контейнеров на всех этапах: сборка (сканирование образа), развертывание (политики PodSecurity, ограничения ресурсов) и эксплуатация (runtime security, мониторинг поведения). Как эти меры связаны между собой и что делать при обнаружении инцидента?

Тестовые вопросы

Тема 6.2. Разработка безопасного ПО и прикладная криптография

Вариант 1

1. Принцип «никогда не доверяй, всегда проверяй» в безопасности называется:
А) Защита в глубину
Б) Zero Trust
В) Наименьшие привилегии
Г) Разделение обязанностей

2. Какой протокол обмена ключами используется в TLS 1.3 для обеспечения совершенной прямой секретности (PFS)?

А) RSA
Б) ECDHE
В) DH
Г) PSK

3. Для безопасного хранения криптографических ключей на iOS используется:
А) Keystore
Б) Keychain
В) Secure Enclave
Г) Android Keystore

4. Какой заголовок HTTP позволяет ограничить источники загрузки ресурсов для предотвращения XSS?
А) Content-Security-Policy
Б) HSTS
В) X-Frame-Options
Г) Referrer-Policy

5. Шифрование на стороне клиента в облачных сервисах означает:
А) Данные шифруются на сервере облачного провайдера
Б) Данные шифруются на стороне клиента перед отправкой, и ключи не передаются провайдеру
В) Используется только SSL/TLS для защиты канала
Г) Шифрование выполняется с использованием симметричного ключа, хранящегося в облаке

6. Какой способ предотвращает массовое присвоение (mass assignment) в REST API?
А) Использование DTO (Data Transfer Objects) с явными полями
Б) Валидация всех полей через аннотации
В) Ограничение размера запроса
Г) Использование только GET-запросов

7. Какой метод защиты от IDOR является правильным?
А) Использование прямых ссылок на объекты
Б) Проверка прав доступа пользователя к объекту на сервере
В) Шифрование идентификаторов
Г) Скрытие идентификаторов в URL

8. Для защиты GraphQL от атак перегрузки необходимо:
А) Использовать только мутации
Б) Ограничить глубину запроса
В) Отключить все запросы
Г) Использовать только rate limiting

9. Какой протокол обеспечивает взаимную аутентификацию между микросервисами в сетке сервисов?
А) HTTPS
Б) mTLS
В) JWT
Г) OAuth2

10. Какой алгоритм постквантовой криптографии выбран NIST для шифрования с открытым ключом (KEM)?
А) Dilithium
Б) Kyber

В) Falcon
Г) SPHINCS+

11. DEK (Data Encryption Key) в иерархии ключей – это:
А) Ключ для шифрования данных
Б) Ключ для шифрования других ключей
В) Ключ для аутентификации
Г) Мастер-ключ

12. Какой инструмент сканирует образы контейнеров на известные уязвимости?
А) Docker
Б) Trivy
В) Kubernetes
Г) Istio

13. Runtime-безопасность контейнеров включает:
А) Проверку подписей образов перед запуском
Б) Мониторинг поведения контейнеров во время выполнения и обнаружение аномалий
В) Сканирование исходного кода
Г) Управление секретами

14. HSM (Hardware Security Module) предназначен для:
А) Хранения больших объемов данных
Б) Выполнения криптографических операций с ключами в защищенной аппаратной среде
В) Маршрутизации сетевых пакетов
Г) Балансировки нагрузки

15. В Kubernetes для ограничения сетевого доступа к подам используются:
А) PodSecurityPolicy
Б) NetworkPolicy
В) Service
Г) Ingress

Вариант 2

1. Принцип, при котором разные задачи выполняются разными субъектами, чтобы предотвратить злоупотребление полномочиями, называется:
А) Наименьшие привилегии
Б) Разделение обязанностей
В) Защита в глубину
Г) Zero Trust

2. Какое изменение в TLS 1.3 повысило безопасность по сравнению с TLS 1.2?
А) Добавление поддержки RSA для обмена ключами
Б) Удаление алгоритмов с несовершенной прямой секретностью и использование только ECDHE
В) Увеличение длины ключа до 4096 бит
Г) Добавление поддержки AES-256-GCM

3. На Android для безопасного хранения ключей используется:
А) Keychain
Б) Android Keystore

В) Secure Enclave
Г) SharedPreferences

4. Заголовок HSTS (HTTP Strict Transport Security) заставляет браузер:
А) Использовать только HTTPS для всех запросов к сайту
Б) Отправлять куки только через HTTPS
В) Запрещать смешанный контент
Г) Проверять сертификаты сервера

5. В облачных средах KMS (Key Management Service) предоставляет:
А) Хранилище ключей с централизованным управлением и контролем доступа
Б) Сервис шифрования дисков
В) Балансировку нагрузки
Г) Мониторинг производительности

6. Для предотвращения SQL-инъекций наиболее эффективным способом является:
А) Экранирование кавычек в строке
Б) Использование параметризованных запросов
В) Использование ORM
Г) Валидация длины строк

7. Какое преимущество управления версиями API с точки зрения безопасности?
А) Позволяет быстро отключать устаревшие уязвимые версии после миграции клиентов
Б) Увеличивает производительность
В) Упрощает тестирование
Г) Снижает требования к валидации

8. Как ограничить доступ к интроспекции GraphQL в production?
А) Отключить интроспекцию полностью
Б) Оставить только для авторизованных пользователей
В) Использовать rate limiting
Г) Ограничить глубину запроса

9. Какая из перечисленных практик является наилучшей для безопасного использования JWT?
А) Использовать симметричное шифрование для подписи
Б) Устанавливать короткое время жизни (exp) и использовать refresh-токены
В) Хранить JWT в localStorage для быстрого доступа
Г) Передавать JWT в URL-параметрах

10. Какой алгоритм постквантовой криптографии выбран NIST для цифровых подписей?
А) Kyber
Б) Dilithium
В) Falcon
Г) SPHINCS+

11. KEK (Key Encryption Key) используется для:
А) Шифрования данных
Б) Шифрования DEK (Data Encryption Key)
В) Аутентификации пользователей
Г) Генерации случайных чисел

12. В Kubernetes для ограничения привилегий контейнера используются:
А) Pod Security Standards (ранее PodSecurityPolicy)
Б) NetworkPolicy
В) ServiceAccount
Г) Ingress

13. Как защитить etcd от несанкционированного доступа?
А) Шифровать данные и использовать TLS для клиентских и серверных соединений
Б) Разместить etcd в отдельной сети без доступа
В) Отключить аутентификацию для ускорения
Г) Регулярно удалять данные

14. Гибридный handshake в TLS сочетает:
А) Классический обмен ключами (например, ECDHE) с постквантовым обменом (например, Kyber) для защиты от квантовых атак
Б) Использование симметричного и асимметричного шифрования
В) Обмен ключами с использованием RSA и DSA
Г) Использование двух разных центров сертификации

15. Инструмент Falco используется для:
А) Сканирования образов
Б) Мониторинга runtime-безопасности контейнеров
В) Управления секретами
Г) Балансировки нагрузки

Ключ к тестам

Вариант 1		Вариант 2	
№ вопроса	Правильный ответ	№ вопроса	Правильный ответ
1	Б	1	Б
2	Б	2	Б
3	Б	3	Б
4	А	4	А
5	Б	5	А
6	А	6	Б
7	Б	7	А
8	Б	8	А
9	Б	9	Б
10	Б	10	Б
11	А	11	Б
12	Б	12	А
13	Б	13	А
14	Б	14	А
15	Б	15	Б

Пракические занятия № 10-18:

1. Реализация end-to-end шифрования для мессенджера на Signal Protocol
2. Настройка TLS 1.3 с perfect forward secrecy и современными cipher suites. Настройка certificate transparency logs для мониторинга SSL сертификатов
3. Создание secure OAuth 2.0 провайдера с PKCE и защитой от атак. Имплементация JWE (JSON Web Encryption) для защищенных токенов. Разработка secure API gateway с JWT

verification и rate limiting. Разработка secure session management с защитой от hijacking
4. Разработка безопасного voting system с homomorphic encryption. Имплементация zero-knowledge proof для аутентификации без пароля. Реализация secure multi-party computation для совместных вычислений. Настройка quantum-resistant cryptography с lattice-based алгоритмами
5. Создание cryptocurrency wallet с ECDSA и hierarchical deterministic keys. Создание blockchain smart contract с защитой от reentrancy attacks
6. Реализация secure password manager с client-side encryption
7. Настройка HSM эмулятора для аппаратной защиты ключей. Создание hardware-backed key storage для мобильного приложения
8. Разработка secure file storage с encryption at rest и in transit
9. Имплементация digital signature system с timestamping. Создание cryptographically secure RNG (random number generator)

5.2. Критерии оценивания

5.2.1. Критерии оценивания устного ответа

При оценке устного ответа, обучающегося учитывается:

1) полнота и правильность ответа;
2) степень осознанности, понимания изученного;

Отметка «5»: ответ правильный, полный в соответствии с изученным материалом; материал изложен в определенной логической последовательности, литературным языком.

Отметка «4»: ответ правильный, полный в соответствии с изученным материалом; материал изложен в определенной логической последовательности; возможны отдельные затруднения в формулировке выводов.

Отметка «3»: ответ, в котором в основном правильно, но схематично или с отклонениями от последовательности изложения раскрыт материал или неполный, несвязный ответ, изложенный нелогично

Отметка «2»: при ответе обнаружено непонимание обучающимся основного содержания учебного материала, неумение его анализировать допущены существенные ошибки, которые обучающийся не смог исправить при наводящих вопросах преподавателя, отсутствует логика в изложении материала, нет необходимых обобщений и самостоятельной оценки фактов; недостаточно сформированы навыки устной речи.

5.2.2. Критерии оценивания выполнения заданий на практических занятиях

Отметка «5»: работа выполнена полностью и правильно; сделаны правильные выводы.

Отметка «4»: работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя.

Отметка «3»: работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки.

Отметка «2»: допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя.

Задания к практическим занятиям представлены в методических указаниях к практическим занятиям по МДК.02.06 Безопасность программного обеспечения.

Методы оценки:

– Защита отчетов по выполненному заданию на практических занятиях.

– Интерпретация результатов наблюдений за деятельностью обучающегося на практических занятиях.

5.2.3. Критерии оценивания тестовых заданий

Оценка в баллах	Степень выполнения задания
Неуд.	Выполнено от 0 до 49,9 % предложенных заданий
Удов.	Выполнено от 50 до 69,9% предложенных заданий
Хор.	Выполнено от 70 до 89,9% предложенных заданий
Отл.	Выполнено от 90 до 100% предложенных заданий

5.2.4. Общая классификация ошибок

При оценке знаний и умений учитываются ошибки и недочеты в работе.
Грубыми считаются ошибки:

- незнание определения основных понятий, законов;
- неумение выделить в ответе главное;
- обобщить результаты изучения;
- неумение применить знания для решения задач;
- неумение использовать полученные данные для выводов;
- неумение пользоваться первоисточниками, учебником, справочником;
- нарушение техники безопасности, небрежное отношение к оборудованию

Негрубыми считаются ошибки:

- неточность формулировок, определений, понятий, законов, вызванная неполнотой охвата основных признаков определяемого понятия или заменой 1-3 из этих признаков второстепенными;
- ошибки, вызванные несоблюдением условий проведения практических занятий;
- недостаточно продуманный план устного ответа (нарушение логики изложения, подмена отдельных основных вопросов второстепенными);
- нерациональные методы работы со справочной литературой;
- неумение выполнять практические задания в общем виде.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ МДК 02.06 Безопасность программного обеспечения

Промежуточная аттестация по междисциплинарному курсу МДК 02.06 Безопасность программного обеспечения проводится в форме экзамена в 7 семестре.

6.1. Вопросы к экзамену по МДК

1. Опишите архитектуру с реализацией принципа "защита в глубину" (defense in depth) для веб-приложения, включая сетевой, прикладной и криптографический уровни. Приведите конкретные технологии и их взаимодействие.
2. Сравните подходы к управлению аутентификацией и авторизацией на основе сессий и на основе JWT. В каких сценариях предпочтителен каждый из них и как обеспечить безопасность в обоих случаях?
3. Объясните механизм согласования параметров (cipher suite negotiation) в TLS 1.3. Какие улучшения по сравнению с TLS 1.2 делают его более безопасным и быстрым?
4. Какие риски возникают при использовании самоподписанных сертификатов в мобильных приложениях и как их минимизировать с помощью pinning сертификатов? Опишите процесс внедрения и потенциальные проблемы.

5. Опишите сценарии использования Web Crypto API в браузере. Какие операции безопасно выполнять на клиенте, а какие должны выполняться на сервере? Почему?
6. Как обеспечить шифрование данных в облачной базе данных (DBaaS) с сохранением возможности выполнения запросов и индексирования? Обсудите компромиссы между производительностью и безопасностью.
7. Разработайте стратегию адаптивного rate limiting для REST API, учитывающую уровень доверия пользователя (аноним, авторизован, премиум) и тип операции. Как предотвратить атаки типа DoS, не нарушая работу легитимных пользователей?
8. Что такое IDOR и как ее предотвратить в REST API на уровне бизнес-логики? Приведите пример безопасной реализации доступа к ресурсам с использованием политик доступа на основе атрибутов (ABAC).
9. Опишите процесс безопасного версионирования REST API с учетом необходимости вывода из эксплуатации устаревших версий. Как обеспечить плавный переход клиентов, не раскрывая уязвимости старых версий?
10. Для GraphQL перечислите специфические уязвимости (глубокие запросы, перегрузка, интроспекция, инъекции в полях) и опишите методы защиты для каждой из них.
11. Как организовать взаимную аутентификацию (mTLS) между микросервисами в динамической среде с автоматическим масштабированием? Опишите использование сервисов типа Vault/SPIRE для управления сертификатами.
12. Сформулируйте рекомендации по безопасному использованию JWT в микросервисной архитектуре: срок действия, аудитория, алгоритм подписи, хранение, обновление (refresh) и отзыв.
13. Что такое атака "обход прокси" (bypass proxy) в Kubernetes при использовании сервисной сетки? Как настроить сетевые политики и перехват трафика для предотвращения такого обхода?
14. Какие угрозы квантовых компьютеров существуют для современных криптосистем (RSA, ECDSA, ECDH)? В какой временной перспективе эти угрозы могут стать реальными и какие стандарты уже готовятся?
15. Опишите алгоритмы Kyber и Dilithium, выбранные NIST в качестве постквантовых стандартов. Для каких целей они предназначены и каковы их основные параметры (размеры ключей, производительность).
16. Объясните концепцию гибридных схем в постквантовой криптографии. Как гибридный handshake в TLS может обеспечить защиту от атак "собрать сейчас, расшифровать позже", сохраняя совместимость с существующими клиентами?
17. Разработайте политику ротации ключей для системы, использующей иерархию ключей (KEK/DEK). Учитывайте частоту ротации, автоматизацию, безопасную доставку новых ключей и минимизацию простоев.
18. Сравните использование аппаратных модулей безопасности (HSM) и облачных KMS для управления ключами. Какие преимущества и ограничения у каждого подхода в контексте распределенных систем?
19. Опишите архитектуру безопасного управления секретами в Kubernetes с использованием внешнего провайдера (например, HashiCorp Vault). Как реализовать динамическую генерацию секретов и их автоматическую ротацию без перезапуска подов?
20. Что такое привилегированные контейнеры и почему они опасны? Какие альтернативы существуют для выполнения задач, требующих повышенных привилегий (например, доступ к /dev, системным вызовам)?
21. Опишите политики безопасности для контейнеров в Kubernetes: Pod Security Standards (разрешающие, базовые, ограничительные). Приведите пример ограничительной политики и объясните ее влияние на работу приложений.

22. Как защитить etcd – хранилище состояний Kubernetes? Какие меры (шифрование, аутентификация, изоляция сети, аудит) критичны для предотвращения компрометации кластера?
23. Разработайте процесс сканирования контейнерных образов на уязвимости в CI/CD. Как интегрировать этот процесс с политиками развертывания, чтобы блокировать запуск образов с критическими уязвимостями?
24. Опишите подходы к runtime-безопасности контейнеров: мониторинг системных вызовов, обнаружение аномалий, ограничение возможностей ядра (capabilities). Приведите примеры используемых инструментов (Falco, AppArmor, seccomp) и их конфигураций.
25. Как обеспечить безопасность данных в REST API при работе с файлами (загрузка, скачивание)? Рассмотрите угрозы (например, атаки на пути, file inclusion, вредоносный контент) и защитные меры.
26. Опишите процесс безопасной обработки платежей в веб-приложении: какие криптографические требования, как обеспечить PCI DSS compliance на уровне передачи и хранения данных.
27. В чем отличие между аутентификацией с использованием OAuth 2.0 и OpenID Connect? Как они взаимодействуют с JWT и где применяются в микросервисной архитектуре?
28. Какие проблемы возникают при передаче конфиденциальных данных через GraphQL (например, в полях с запросами) и как их избежать с помощью правил авторизации на уровне разрешений (resolver-level)?
29. Объясните, как работает шифрование с помощью эллиптических кривых (ECC) по сравнению с RSA. Почему ECC предпочтительнее для мобильных и IoT-устройств, и какие угрозы несет квантовый прорыв для ECC?
30. Спроектируйте систему для миграции с RSA на постквантовую криптографию в крупной корпоративной среде с тысячами клиентов. Какие этапы, тестирование, обратная совместимость и резервное планирование вы предусмотрите?

6.2.Критерии оценивания ответов на экзамене

– оценка **«отлично»**, если студент обладает глубокими и прочными знаниями программного материала; при ответе на вопросы продемонстрировал исчерпывающее, последовательное и логически стройное изложение; правильно сформулировал понятия и закономерности по вопросам; сделал вывод по излагаемому материалу;

– оценка **«хорошо»**, если студент обладает достаточно полным знанием программного материала; его ответ представляет грамотное изложение учебного материала; но имеются существенные неточности в формулировании понятий и закономерностей по вопросам; не полностью сделаны выводы по излагаемому материалу;

– оценка **«удовлетворительно»**, если студент имеет общие знания основного материала без усвоения некоторых существенных положений; формулирует основные понятия с некоторой неточностью; затрудняется в приведении примеров, подтверждающих теоретические положения;

– оценка **«неудовлетворительно»**, если студент не знает значительную часть программного материала; допустил существенные ошибки в процессе изложения; не умеет выделить главное и сделать вывод; приводит ошибочные определения; ни один вопрос не рассмотрен до конца, наводящие вопросы не помогают.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Основные печатные издания

1. Емелина, Е. И., Поддержка и тестирование программных модулей : учебник / Е. И. Емелина. - Москва: КноРус, 2025. — 267 с.

2. Зубкова Т.М. Технология разработки программного обеспечения. – Санкт-Петербург: Лань, 2021. – 252 с.

3. Шитов, В. Н., Проектирование и разработка интерфейсов пользователя: учебное пособие / В. Н. Шитов, К. Е. Успенский. — Москва: КноРус, 2023. — 294 с.

Основные электронные издания:

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва: ФОРУМ: ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст: электронный. - URL: https://znanium.com/catalog/product/1794453

2. Зараменских, Е. П. Проектирование информационных систем: учебник и практикум для среднего профессионального образования / Е. П. Зараменских. — 2-е изд. — Москва: Издательство Юрайт, 2025. — 119 с. — (Профессиональное образование). — ISBN 978-5-534-21417-8. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/571330