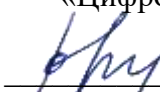


Федеральное государственное образовательное бюджетное учреждение
высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

ЦЕНТР КОМПЕТЕНЦИЙ «ЦИФРОВАЯ ЭКОНОМИКА»

УТВЕРЖДАЮ

Директор Центра компетенций
«Цифровая экономика»

 Н.Ю. Сурова

«01» августа 2026 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА

«Кибербезопасность. 10-11 класс (базовый уровень)»

Учебная нагрузка: 145 академических часа.

МОСКВА 2026

1. Пояснительная записка

Актуальность

Программа базового уровня предназначена для обучающихся 10–11 классов, знакомых с базовыми алгоритмическими конструкциями, но не имеющих систематических знаний в области программирования и информационной безопасности. Ее актуальность обусловлена стремительным ростом киберугроз в условиях цифровой трансформации – большинство старшеклассников активно используют цифровые сервисы, но не обладают практическими навыками защиты личных данных, распознавания угроз и безопасной разработки. Программа восполняет этот пробел, формируя у школьников культуру кибербезопасности и навыки программирования на Python в контексте реальных задач защиты информации.

Программа имеет выраженную профориентационную ценность: информационная безопасность является одной из наиболее динамично развивающихся сфер ИТ-индустрии, а обучение позволяет школьникам погрузиться в практические задачи ИБ-специалиста – от шифрования данных до анализа веб-уязвимостей. Программа помогает ученикам осознать, как полученные навыки соотносятся с кадровыми потребностями рынка, и служит надёжным фундаментом для выбора будущей профессии. Кроме того, содержание программы интегрирует элементы кодификатора ЕГЭ по информатике, что повышает ее практическую значимость.

Отличительные особенности программы:

- обучение через призму информационной безопасности – базовый синтаксис Python (переменные, условия, циклы, функции, ООП) осваивается на задачах валидации паролей, шифрования файлов, анализа логов и моделирования атак;
- модульная структура, что обеспечивает системность знаний;
- проектная деятельность – итогом обучения становится разработка и защита итогового проекта, что позволяет применить все полученные навыки в законченном продукте.

Новизна программы:

- использование ИИ-инструментов как объекта контроля – обучающиеся учатся проверять и исправлять код, сгенерированный нейросетями, что развивает критическое мышление, навыки отладки и понимание ограничений ИИ;
- формирование культуры кибербезопасности с первых занятий – особое внимание уделяется триаде информационной безопасности, правовым и этическим рамкам;
- связь с индустрией – в программу интегрированы кейс-задачи без единственно верного ответа, включая задачу от аккредитованной ИТ-компании, что позволяет школьникам погрузиться в реальные бизнес-контексты и актуальные проблемы отрасли, а также формирует первичное представление о профессиональной деятельности в сфере информационной безопасности.

Описание программы

Программа направлена на формирование у обучающихся системных знаний и практических навыков в области информационной безопасности и программирования на Python для защиты информации и разработки безопасных приложений. Программа включает в себя четыре модуля, охватывающих основные аспекты безопасной разработки, криптографии, анализа уязвимостей и проектной деятельности.

Формы реализации – очная без применения дистанционных образовательных технологий (в том числе, с применением средств электронного обучения).

Общий объем программы 145 академических часов.

Первый модуль посвящён основам программирования на Python и объектно-ориентированному программированию в контексте информационной безопасности. Обучающиеся систематизируют знания базового синтаксиса языка, углублённо изучают ООП – классы, объекты, наследование и инкапсуляцию, осваивают работу с файловой системой, обработку исключений и взаимодействие с API. Все темы отрабатываются на практических задачах: от создания классов для безопасной аутентификации пользователей до написания скриптов, моделирующих сценарии атак. В результате обучающиеся знают принципы объектно-ориентированного проектирования, умеют строить модульную архитектуру приложений, владеют навыками работы с внешними API и имеют навыки создания безопасного, структурированного кода.

Второй модуль – математические основы криптографии и криптографические протоколы. В рамках этого модуля обучающиеся изучают математический фундамент современной криптографии, включая модульную арифметику и теорию чисел, осваивают реализацию симметричных и асимметричных алгоритмов шифрования. Обучающиеся программируют алгоритмы RSA и AES с использованием библиотеки `cryptography`, изучают механизмы электронной подписи и инфраструктуру открытых ключей. Особое внимание уделяется построению модели угроз и выбору криптографических средств в зависимости от решаемой задачи. В результате обучающиеся понимают математические принципы, лежащие в основе криптоалгоритмов, умеют реализовывать RSA и AES на Python, владеют навыками генерации и управления ключами шифрования, а также умеют строить модель угроз для информационной системы.

Третий модуль посвящён анализу веб-уязвимостей и статическому анализу кода. Обучающиеся углублённо изучают типовые веб-уязвимости, включая SQL-инъекции, межсайтовый скриптинг и подделку межсайтовых запросов, осваивают методы их обнаружения и эксплуатации в контролируемой среде, а также принципы безопасной среды разработки. В модуле рассматриваются сетевые угрозы и создание простейших сетевых инструментов для мониторинга безопасности. Уникальной особенностью модуля является статический анализ кода: обучающиеся проверяют собственный код, анализируют предоставленные фрагменты и учатся находить уязвимости в коде, сгенерированном ИИ-инструментами. В результате обучающиеся умеют распознавать и эксплуатировать базовые веб-угрозы, владеют методами статического анализа кода, имеют навыки настройки безопасной среды разработки и осознают этические и правовые границы пентеста.

Четвёртый модуль — проектная деятельность и погружение в профессию. В этом модуле обучающиеся применяют весь спектр полученных знаний для разработки и защиты итогового проекта, что позволяет создать законченный продукт для портфолио. Модуль включает решение кейс-задач без единственно верного ответа, в том числе от аккредитованной ИТ-компании, что позволяет школьникам погрузиться в реальные бизнес-контексты и актуальные проблемы отрасли. Содержание модуля также интегрирует элементы из кодификатора ЕГЭ по информатике, что повышает его практическую значимость для подготовки к экзамену. В результате обучающиеся получают опыт завершённой инженерной деятельности от идеи до защиты, умеют презентовать свои решения, владеют навыками командной работы и распределения ролей, а также формируют осознанное представление о профессиональных треках в сфере информационной безопасности.

Программа предназначена для обучающихся 10–11 классов, знакомых с базовыми алгоритмическими конструкциями, но не имеющих систематических знаний в области программирования и информационной безопасности, и желающих получить практико-ориентированные навыки в этой востребованной сфере.

Программа обеспечена учебно-методическими ресурсами для качественного освоения содержания учебного материала, что также обеспечивает результативность освоения модулей. Обучающимся обеспечен доступ к современному практико-ориентированному содержанию

электронных образовательных и информационных ресурсов. Предусмотрены материально-технические условия – программное обеспечение и оборудование.

Преподавательский состав включает IT-специалистов, имеющих большой педагогический стаж в области преподавания языков программирования и обеспечения информационной безопасности.

Аннотация

«Кибербезопасность. 10-11 класс (базовый уровень)» – дополнительная общеобразовательная программа технической направленности, ориентированная на углублённое изучение программирования и системных принципов кибербезопасности. Программа рассчитана на базовый уровень сложности.

Дополнительная общеобразовательная программа предусматривает поэтапное освоение языка Python и ключевых концепций защиты информации с переходом к реализации криптографических алгоритмов и анализу уязвимостей.

Программа направлена на формирование системных знаний и практических навыков в области информационной безопасности и программирования на Python для защиты информации и разработки безопасных приложений.

Программа предназначена для школьников 10–11 классов, знакомых с базовыми алгоритмическими конструкциями и желающих получить углублённые практические навыки в сфере информационной безопасности.

Программа формирует следующие навыки:

- программирование на Python с применением объектно-ориентированного подхода и работой с API;
- реализация криптографических алгоритмов (RSA, AES) с использованием специализированных библиотек;
- построение модели угроз и выбор криптографических средств в зависимости от решаемой задачи;
- проведение статического анализа кода и выявление уязвимостей, в том числе в коде, сгенерированном ИИ-инструментами;
- разработка простейших сетевых инструментов для мониторинга безопасности.

Программа способствует формированию целостного понимания информационной безопасности, её этических и правовых аспектов. Обучающиеся научатся использовать цифровые технологии осознанно и ответственно, критически оценивая риски, соблюдая культуру цифровой гигиены и понимая правовую ответственность специалиста по информационной безопасности.

Входные требования к обучающимся: необходимо знание базовых алгоритмических конструкций и начальные навыки программирования. Специальной подготовки в области информационной безопасности не требуется.

Программа базового уровня сочетает глубокую техническую подготовку с развитием командных навыков и правовой грамотности. Содержание и материалы программы разработаны с учётом возрастных особенностей целевой аудитории и обеспечивают последовательное усвоение материала от базового синтаксиса Python до разработки и защиты итогового проекта. Предусмотрена система повторения: каждый модуль опирается на навыки предыдущего, а в четвёртом модуле обучающиеся интегрируют все полученные знания в итоговом проекте. Программа построена на дифференцированном подходе и предусматривает практические задания двух уровней сложности.

Программа ориентирована на обеспечение вклада в развитие дальнейшей образовательной траектории в сфере ИТ за счёт включения элементов содержания из кодификатора ЕГЭ по информатике и практических заданий, направленных на подготовку к ЕГЭ. Начиная со второго модуля, программа предусматривает вариативность образовательной

траектории, которая выражается в предоставлении обучающимся выбора контекста практических заданий.

В программу интегрированы кейс-задачи, не имеющие единственно верного ответа, включая задачу, представленную аккредитованной ИТ-организацией. В практических заданиях предусмотрены основы коллективной работы: парное программирование, командный аудит кода с распределением ролей и критериями оценки индивидуального вклада каждого участника.

Программа включает работу с использованием искусственного интеллекта (ИИ): нейросети используются как инструмент для поиска уязвимостей и анализа кода, а обучающиеся учатся проверять и исправлять сгенерированный ИИ код. Предусмотрена «Политика использования ИИ-инструментов», регламентирующая допустимые и запрещённые виды деятельности. Правовой блок программы охватывает Доктрину информационной безопасности РФ, защиту персональных данных, уголовную ответственность в сфере компьютерной информации и этику ответственного раскрытия уязвимостей.

Промежуточные аттестации более чем наполовину состоят из практико-ориентированных заданий на написание кода, а итоговая аттестация реализована в формате, устойчивом к ИИ-генерации, и включает тестирование и защиту проекта с демонстрацией результатов проектной работы.

Результаты освоения программы востребованы как для дальнейшего обучения, так и в профессиональной деятельности. Изучение криптографических алгоритмов, освоение методов статического анализа кода и практических подходов к построению модели угроз формируют прочную основу для успешного старта и уверенного продвижения в сфере информационной безопасности и смежных направлениях ИТ.

Цель программы: формирование у обучающихся 10-11 классов системных знаний и практических навыков в области информационной безопасности и программирования на Python для защиты информации и разработки безопасных приложений.

Задачи:

1. Освоить базовые конструкции Python, ООП, работу с файлами и API.
2. Изучить математические основы криптографии, реализовать алгоритмы RSA и AES.
3. Сформировать умения находить, анализировать и устранять типовые веб-уязвимости как в собственном, так и в предоставленном коде, включая код, сгенерированный ИИ-инструментами.
4. Воспитать культуру цифровой гигиены и этичного поведения.
5. Сформировать опыт завершённой инженерной деятельности: от выбора идеи проекта до его защиты.

Задачи обучения, развития, воспитания:

Задачи обучения:

сформировать отношение к компьютерной технике и современным техническим устройствам как к профессиональному инструменту для решения задач в области информационной безопасности и реализации сложных ИТ-проектов;

сформировать мотивацию к углублённому изучению программных ИТ-инструментов для создания безопасных приложений и проведения анализа защищённости;

сформировать системное представление о программных разработках с помощью языка программирования Python, включая объектно-ориентированное программирование и работу с API;

сформировать представление о математических основах криптографии, принципах работы симметричных и асимметричных алгоритмов шифрования;

сформировать представление о принципах построения модели угроз и выбора криптографических средств защиты информации;

изучить криптографическую библиотеку `cryptography` и основы реализации алгоритмов RSA и AES;

изучить методы статического анализа кода, принципы поиска и эксплуатации веб-уязвимостей, а также принципы настройки безопасной среды разработки.

Задачи воспитания:

сформировать нормы культурного проявления социальной активности и гражданской позиции в отношении тенденций развития отечественных IT-технологий в области информационной безопасности и защиты данных;

сформировать морально-этическую основу для развития профессиональной позиции в отношении этичного использования технологий искусственного интеллекта и нейронных сетей, включая вопросы ответственного раскрытия уязвимостей;

сформировать основу для развития культуры профессионального общения в IT-сфере, в том числе навыков командной работы, распределения ролей и оценки индивидуального вклада;

воспитать осознанное отношение к правовой ответственности в сфере компьютерной информации, включая знание Доктрины информационной безопасности РФ.

Задачи развития:

способствовать развитию системного и критического мышления;

развивать творческий подход к решению нестандартных задач в области кибербезопасности;

развивать профессиональный тезаурус и понимание принципов построения защищённых информационных систем;

развивать навыки самостоятельного анализа защищённости кода и принятия обоснованных решений при выборе методов защиты.

Формы и режим занятий

Формы занятий – теоретическая подготовка, практическая работа, самостоятельная работа.

Объем часов в неделю – до 6 академических часов.

Описание планируемых результатов обучения:

Сформированные у обучающихся базовые умения и навыки в области кибербезопасности.

По окончании обучения по программе базового уровня обучающиеся будут: иметь представление:

- о триаде информационной безопасности;
- об этике и правовых нормах;
- о принципах работы веб-уязвимостей.

знать:

- основы синтаксиса Python;
- ООП;
- базовые криптографические алгоритмы;
- типы сетевых и веб-угроз;
- принципы безопасной среды разработки.

уметь:

- писать скрипты для шифрования и хеширования;
- реализовывать простейшие сетевые инструменты;
- распознавать базовые веб-угрозы и применять методы их нейтрализации;
- проводить статический анализ кода.

владеть:

- инструментами криптографической библиотеки cryptography;
- методами построения модели угроз;
- навыками безопасной работы с файловой системой и данными.

У обучающегося будут сформированы:

- культура цифровой гигиены и ответственного поведения в сети;
- понимание правовой ответственности и этических границ деятельности

специалиста по ИБ;

- критическое мышление при работе с ИИ-инструментами;
- навыки командной работы;
- мотивация к дальнейшему изучению ИТ и информационной безопасности, в том

числе к участию в олимпиадах и соревнованиях.

Планируемые результаты обучения измеримы и проверяемы через систему текущего контроля, промежуточной аттестации по каждому модулю и итогового контроля по ДОП.

Личностные результаты:

ответственное отношение к учению, способность к саморазвитию и самообразованию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в профессиональных предпочтениях, с учётом устойчивых познавательных интересов и уважительного отношения к труду;

готовность и способность к самообразованию на протяжении всей жизни как условию успешной профессиональной и общественной деятельности;

формирование осознанной гражданской позиции в отношении защиты персональных данных и соблюдения норм информационной безопасности;

понимание правовой ответственности и этических границ деятельности специалиста по информационной безопасности, включая знание Доктрины информационной безопасности РФ.

Метапредметные результаты:

умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в познавательной деятельности, развивать её мотивы и интересы;

умение самостоятельно планировать альтернативные пути достижения целей, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;

умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий и требований, корректировать их в соответствии с изменяющейся ситуацией;

умение самостоятельно определять цели деятельности и составлять её планы; контролировать и корректировать деятельность; использовать все возможные ресурсы для реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

умение продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников, эффективно разрешать конфликты;

развитие системного и критического мышления при анализе защищённости кода, в том числе сгенерированного ИИ-инструментами.

Предметные результаты:

представление о математических основах криптографии и принципах работы симметричных и асимметричных алгоритмов шифрования;

представление о возможностях языка программирования Python для реализации криптографических алгоритмов и построения модели угроз;

представление о программных инструментах и специализированных библиотеках, использующихся для криптографической защиты данных и статического анализа кода;

системные навыки и умения написания программного кода на языке программирования Python с применением объектно-ориентированного подхода в соответствии с задачами информационной безопасности;

навыки и умения реализации алгоритмов RSA и AES с использованием библиотеки cryptography;

навыки и умения проведения статического анализа кода, распознавания и эксплуатации веб-уязвимостей в контролируемой среде, а также настройки безопасной среды разработки.

2. Учебный план

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
Модуль 1. Python как инструмент для кибербезопасности	36	10	18	6	2	
Тема 1.1. Знакомство с образовательной платформой, инструментами обучения и введение в кибербезопасность	2	2	0	0	0	
Тема 1.2. Ответственное использование ИТ: государственная политика ИБ, персональные данные, правовая ответственность и этика ИИ	3	2	0	1	0	
Тема 1.3. Безопасная среда разработки	6	1	3	1	1	Выполнение тестовых заданий для текущего контроля знаний и

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
						навыков по пройденному материалу
Тема 1.4. Структуры данных и модули	5	1	3	1	0	
Тема 1.5. Объектно-ориентированная логика защиты: классы и инкапсуляция	10	2	6	2	0	
Тема 1.6. Криптографически стойкая случайность и работа с байтами. Кейс-задача	9	2	6	1	0	
Промежуточная аттестация по модулю 1	1	0	0	0	1	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Модуль 2. Прикладная криптография: от математики к защищённому коду	36	7	21	6	2	
Тема 2.1. Математика криптографии: модульная арифметика и простые числа	10	2	6	2	0	
Тема 2.2. RSA с нуля: реализация асимметричного шифрования	5	1	3	1	0	
Тема 2.3. AES и современное	6	1	3	1	1	Выполнение тестовых заданий

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
симметричное шифрование						для текущего контроля знаний и навыков по пройденному материалу
Тема 2.4. НМАС и целостность данных: защита от подделки	5	1	3	1	0	
Тема 2.5. Взлом слабой криптографии: словарные атаки и радужные таблицы	5	1	3	1	0	
Тема 2.6. РКІ и цифровые подписи. Кейс-задача	4	1	3	0	0	
Промежуточная аттестация по модулю 2	1	0	0	0	1	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Модуль 3. Сетевая безопасность и веб-уязвимости: инструменты атаки и защиты	36	7	21	6	2	
Тема 3.1. Сокеты и сетевое программирование: клиент-сервер с шифрованием	10	2	6	2	0	
Тема 3.2. Сканер портов и сетевая разведка. ИИ как инструмент в ИБ	5	1	3	1	0	

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
Тема 3.3. Анализ логов и обнаружение аномалий	5	1	3	0	1	Выполнение тестовых заданий для текущего контроля знаний и навыков по пройденному материалу
Тема 3.4. Веб-уязвимости: XSS и CSRF. Проверка уязвимостей с применением ИИ	5	1	3	1	0	
Тема 3.5. Path Traversal: небезопасная работа с файловой системой	5	1	3	1	0	
Тема 3.6. Безопасная работа с файлами и сериализация данных. Кейс-задача	5	1	3	1	0	
Промежуточная аттестация по модулю 3	1	0	0	0	1	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Модуль 4. Итоговый проект: от идеи к готовому продукту	36	6	18	10	2	
Тема 4.1. Этика, закон и профессия: белые, серые и чёрные хакеры	5	1	3	1	0	
Тема 4.2. Аудит кода: статический анализ и поиск уязвимостей	10	2	6	2	0	

Наименование модулей/тем программы	Всего, час	Виды учебных занятий				Формы контроля
		Теория	Практика	Самостоятельная работа	Контроль/ аттестация, ак. час	
Тема 4.3. Проектирование итогового проекта: модель угроз (threat model) и архитектура	7	1	3	2	1	Выполнение тестовых заданий для текущего контроля знаний и навыков по пройденному материалу
Тема 4.4. Разработка, отладка и статический анализ проекта	8	1	3	4	0	
Тема 4.5. Оформление проекта	5	1	3	1	0	
Промежуточная аттестация по модулю 4	1	0	0	0	1	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Итоговый контроль	1	0	0	0	1	Комплексное задание: тест и защита проекта
Итого	145	30	78	28	9	
ИТиСИ	Объем часов ИТиСИ не входит в общую трудоемкость ДОП, но участие в ИТиСИ обязательно для всех успешно прошедших Итоговый контроль/аттестацию по ДОП					

3. Календарно-тематическое планирование

Название ДОП	№ потока	Дата начала обучения по ДОП	№ модуля ДОП	Начало обучения по модулю ДОП	Календарный период (количество дней) длительности модуля	Дата окончания обучения по ДОП
Кибербезопасность. 10-11 класс (базовый уровень)	1	2026-10-01	1	2026-10-01	61	2027-05-28
			2	2026-12-01	67	
			3	2027-02-08	54	
			4	2027-04-05	54	

4. Рабочая программа

Порядковый номер модуля	Наименование темы	Лекции. Количество академических часов	Содержание теоретической подготовки	Практические занятия. Количество академических часов	Содержание практической работы	Самостоятельная работа. Количество академических часов	Содержание самостоятельной работы
1	Python как инструмент для кибербезопасности						
1	Тема 1.1. Знакомство с образовательной платформой, инструментами обучения и введение в кибербезопасность	2	Знакомство с LMS и основными инструментами обучения: теоретические материалы, тренажёры кода, Jupyter Notebook, тесты, самостоятельная работа. Кибербезопасность как защита данных, устройств, сервисов и пользователей. CIA-триада: конфиденциальность, целостность, доступность. Основные причины инцидентов: слабые цифровые привычки, ошибки в коде, неверные настройки, действия злоумышленников. Цифровой след и защищаемые активы.	0	Практическая работа в указанной теме Дополнительной общеобразовательной программы не предусмотрена и проводиться не будет.	0	Самостоятельная работа и самоконтроль в указанной теме Дополнительной общеобразовательной программы не предусмотрены и проводиться не будут.

			Базовая модель угроз. Типовые сценарии атак: фишинг, кража паролей, вредоносные вложения, повторное использование учётных данных. Примеры нарушений CIA-триады: утечки данных, ransomware-атаки, подмена информации				
1	Тема 1.2. Ответственное использование ИТ: государственная политика ИБ, персональные данные, правовая ответственность и этика ИИ	2	Государственная политика в области информационной безопасности. Доктрина информационной безопасности РФ. Понятие персональных данных и основные положения 152-ФЗ. Статьи 272–274 УК РФ: уголовная ответственность в сфере компьютерной информации. Этические принципы специалиста по ИБ. Различие между законным исследованием уязвимостей и противоправным вмешательством. Программы bug bounty и ответственное раскрытие уязвимостей. Этика применения ИИ-	0	Практическая работа в указанной теме Дополнительной общеобразовательной программы не предусмотрена и проводиться не будет.	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.

			инструментов при написании и анализе кода.				
1	Тема 1.3. Безопасная среда разработки	1	Базовые конструкции Python: переменные, типы данных, условия, циклы и функции. Виртуальное окружение и изоляция зависимостей проекта. Переменные окружения как способ хранения конфигурации и чувствительных данных. Почему пароли, токены и API-ключи нельзя хранить в коде. Git как система контроля версий: цикл init — add — commit — push. Файл .gitignore и предотвращение публикации секретов и служебных файлов. Безопасная среда разработки как сочетание базовых навыков программирования, контроля версий и правильной работы с секретами.	3	Повторение теории, необходимой для выполнения практических заданий по безопасной среде разработки. Подробный разбор совместно с обучающимися решения задач по созданию виртуального окружения. Выполнение практических заданий по настройке среды разработки и работе с Git. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний.	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания
1	Тема 1.4. Структуры данных и модули	1	Списки, словари, множества и кортежи как инструменты для задач ИБ. Хранение попыток	3	Повторение теории, необходимой для выполнения практических заданий по	1	Повторение и систематизация пройденного материала по

			входа, чёрных списков IP-адресов и журналов событий. Алгоритмическая сложность операций и её значение для безопасности. Сравнение временных характеристик $O(n)$ и $O(1)$. Влияние выбора структуры данных на устойчивость системы к атакам перебора. Модули стандартной библиотеки Python для задач информационной безопасности.		структурам данных и модулям Python. Подробный разбор совместно с обучающимися решения задач по выбору оптимальных структур данных для хранения логинов, паролей, IP-адресов. Выполнение практических заданий по их реализации. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний.		ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
1	Тема 1.5. Объектно-ориентированная логика защиты: классы и инкапсуляция	2	Классы и объекты как основа построения безопасных систем. Атрибуты и методы класса. Инкапсуляция: сокрытие внутреннего состояния и контролируемый доступ к данным. Приватные атрибуты и их роль в защите чувствительных данных. Валидация данных через методы	6	Повторение теории, необходимой для выполнения практических заданий по объектно-ориентированной логике защиты. Подробный разбор совместно с обучающимися решения задач по созданию классов с инкапсуляцией и валидацией. Выполнение	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и

			класса. ООП как инструмент построения защищённой архитектуры приложения.		практических заданий по разработке иерархии классов для задач ИБ. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		самоконтроля/самопроверки с автоматической системой оценивания.
1	Тема 1.6. Криптографически стойкая случайность и работа с байтами. Кейс-задача	2	Псевдослучайные и криптографически стойкие генераторы случайных чисел. Почему генераторы общего назначения непригодны для паролей, токенов и ключей. Двоичные данные и работа с байтами на низком уровне. Понятие энтропии как меры устойчивости к подбору. Измерение энтропии строки. Применение	6	Повторение теории, необходимой для выполнения практических заданий по криптографически стойкой случайности и работе с байтами. Подробный разбор совместно с обучающимися решения и выполнение практических заданий по по криптографически стойкой случайности и работе с байтами. Разбор	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.

			криптографически стойкой случайности при генерации одноразовых токенов.		кейс-задачи. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний.		
2	Прикладная криптография: от математики к защищённому коду						
2	Тема 2.1. Математика криптографии: модульная арифметика и простые числа	2	Наибольший общий делитель и алгоритм Евклида. Взаимно простые числа. Арифметика по модулю как математический фундамент асимметричного шифрования. Простые числа как строительный материал криптографических алгоритмов. Вероятностные тесты простоты: тест Миллера — Рабина. Быстрое возведение в степень по модулю и его роль в криптографических вычислениях.	6	Повторение теории, необходимой для выполнения практических заданий по математике криптографии. Подробный разбор совместно с обучающимися решения задач по реализации алгоритма Евклида, теста Миллера-Рабина и быстрого возведения в степень. Выполнение практических заданий по модульной арифметике. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.

					закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
2	Тема 2.2. RSA с нуля: реализация асимметричного шифрования	1	Генерация пары открытого и закрытого ключей. Процедуры шифрования и расшифрования в RSA. Функция Эйлера и выбор показателя шифрования. Связь модульной арифметики с асимметричной криптографией. Уязвимость при малом значении открытой экспоненты. Практические последствия ошибок при выборе параметров RSA.	3	Повторение теории, необходимой для выполнения практических заданий по реализации RSA с нуля. Подробный разбор совместно с обучающимися решения задач по генерации ключей, шифрованию и расшифрованию. Выполнение практических заданий по генерации ключей, шифрованию и расшифрованию, в том числе вариативное задание. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

					знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
2	Тема 2.3. AES и современное симметричное шифрование	1	Блочное симметричное шифрование и принцип работы AES. Режим ECB и его уязвимость: сохранение структуры открытого текста в шифротексте. Режим CBC и его ограничения. Режим GCM: совмещение шифрования с проверкой целостности. Аутентифицированное шифрование как современный стандарт. Выбор режима шифрования как фактор безопасности системы.	3	Повторение теории, необходимой для выполнения практических заданий по реализации RSA с нуля. Подробный разбор совместно с обучающимися решения задач по симметричному шифрованию и расшифрованию. Выполнение практических заданий по указанным темам. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.

					«Материалы для преподавателя по проведению практических занятий».		
2	Тема 2.4. НМАС и целостность данных: защита от подделки	1	Ограничения хеш-функций при использовании без ключа. Код аутентификации сообщений НМАС. Атака удлинения сообщения и уязвимость незащищённых конструкций на основе SHA. Устранение уязвимости за счёт схемы вложенного хеширования в НМАС. Применение НМАС для проверки целостности и подлинности данных. Защита API-запросов с помощью НМАС.	3	Повторение теории, необходимой для выполнения практических заданий по реализации RSA с нуля. Подробный разбор совместно с обучающимися решения задач, выполнение практических заданий по проверке целостности и подлинности данных, в том числе вариативного задания. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

2	Тема 2.5. Взлом слабой криптографии: словарные атаки и радужные таблицы	1	Уязвимость MD5 и SHA1 без соли при хранении паролей. Принцип словарных атак. Радужные таблицы и их эффективность против незащищённых хешей. Роль соли в защите от атак по предвычисленным таблицам. Современные алгоритмы хеширования паролей, устойчивые к перебору. Вычислительная стоимость алгоритма как фактор практической защищённости.	3	Повторение теории, необходимой для выполнения практических заданий по реализации RSA с нуля. Подробный разбор совместно с обучающимися решения задач по хранению паролей. Выполнение практических заданий по построению защиты от атак. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.
2	Тема 2.6. PKI и цифровые подписи. Кейс-задача	1	Инфраструктура открытых ключей (PKI). Цифровой сертификат X.509 и его структура. Цепочка доверия и роль центров сертификации.	3	Повторение теории, необходимой для выполнения практических заданий по инфраструктуре открытых ключей.	0	Самостоятельная работа и самоконтроль в указанной теме Дополнительной общеобразовательной программы не

			Формирование и проверка цифровой подписи. Гарантии подлинности и целостности документа. Самоподписанные сертификаты и сферы их применения.		Подробный разбор совместно с обучающимися решения и выполнение практических заданий по инфраструктуре открытых ключей (PKI), формированию и проверке цифровой подписи. Разбор кейс-задачи. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		предусмотрены и проводиться не будут.
3	Сетевая безопасность и веб-уязвимости: инструменты атаки и защиты						
3	Тема 3.1. Сокеты и сетевое программирование: клиент-сервер с шифрованием	2	Сокеты TCP и UDP как основа сетевого программирования. Построение клиент-серверного взаимодействия. Подтверждение доставки	6	Повторение теории, необходимой для выполнения практических заданий по сетевым сокетам и клиент-серверному взаимодействию.	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление

			(ACK/NACK) и обнаружение потери пакетов. Надёжный протокол поверх ненадёжного канала. Интеграция криптографических механизмов в сетевой канал. Шифрование передаваемых данных на прикладном уровне.		Подробный разбор совместно с обучающимися решения задач, выполнение практических заданий по обмену сообщениями с шифрованием. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.
3	Тема 3.2. Сканер портов и сетевая разведка. ИИ как инструмент в ИБ	1	Принципы сетевой разведки и выявление доступных служб. Сканирование портов как способ обнаружения потенциальных точек входа. Многопоточность при сканировании сети. Заголовки безопасности HTTP: CSP, HSTS, X-Frame-Options, X-	3	Повторение теории, необходимой для выполнения практических заданий по сканированию портов и анализу HTTP-заголовков безопасности. Подробный разбор совместно с обучающимися решения задач по разработке	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий

			Content-Type-Options. Аудит заголовков как элемент проверки защищённости веб-приложения. Применение ИИ-инструментов для генерации и проверки кода в задачах ИБ.		скрипта для проверки заголовков. Выполнение практических заданий по созданию скрипта аудита HTTP-заголовков безопасности сайта, в том числе вариативного задания. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
3	Тема 3.3. Анализ логов и обнаружение аномалий	1	Форматы журналов событий веб-серверов и операционных систем. Регулярные выражения для извлечения данных из логов. Статистические методы обнаружения аномалий. Пороговое значение и z-оценка для выявления отклонений. Признаки брутфорс-	3	Повторение теории, необходимой для выполнения практических заданий по XSS и CSRF. Подробный разбор совместно с обучающимися решения задач по веб-уязвимостям. Выполнение	0	Самостоятельная работа и самоконтроль в указанной теме. Дополнительной общеобразовательной программы не предусмотрены и проводиться не будут.

			атаки в журналах событий. Формирование отчёта по результатам анализа логов.		практических заданий по использованию ИИ для поиска уязвимостей в коде. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
3	Тема 3.4. Веб-уязвимости: XSS и CSRF. Проверка уязвимостей с применением ИИ	1	Межсайтовый скриптинг XSS: хранимый, отражённый и DOM-based. Механизм атаки CSRF и принцип защиты токеном. Санитизация пользовательского ввода. Политика безопасности контента CSP. Применение ИИ для поиска уязвимостей в коде. Сравнение автоматического анализа с ручным code review.	3	Повторение теории, необходимой для выполнения практических заданий по реализации RSA с нуля. Подробный разбор совместно с обучающимися решения задач по хранению паролей. Выполнение практических заданий по построению защиты от атак. Подробное описание практических заданий представлено в	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.

					приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
3	Тема 3.5. Path Traversal: небезопасная работа с файловой системой	1	Атака обхода каталогов и последовательности «../». URL-кодирование как способ обхода простых проверок пути. Доступ к файлам за пределами разрешённой директории. Декодирование и нормализация пути перед проверкой. Приведение пути к абсолютному виду. Белый список допустимых директорий как мера защиты.	3	Повторение теории, необходимой для выполнения практических заданий по защите от Path Traversal. Подробный разбор совместно с обучающимися решения задач по небезопасной работе с файловой системой. Выполнение практических заданий по о нормализации путей и валидации ввода, в том числе вариативное задание. Подробное описание практических заданий представлено в приложении к ДОП. Проведение	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

					диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
3	Тема 3.6. Безопасная работа с файлами и сериализация данных. Кейс-задача	1	Опасности десериализации недоверенных данных. Выполнение произвольного кода через уязвимые механизмы сериализации. Безопасные форматы сериализации, не допускающие выполнения кода. Валидация входных данных как обязательный этап перед обработкой. Проверка типов и структуры данных до десериализации. Принцип минимального доверия к внешним данным.	3	Повторение теории, необходимой для выполнения практических заданий по безопасной сериализации. Подробный разбор совместно с обучающимися решения и выполнение практических заданий по реализации безопасной сериализации. Разбор кейс-задачи. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.

					упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		
4	Итоговый проект: от идеи к готовому продукту						
4	Тема 4.1. Этика, закон и профессия: белые, серые и чёрные хакеры	1	Классификация специалистов по этическим принципам: белые, серые и чёрные хакеры. Договор на проведение пентеста и соглашение о неразглашении (NDA). Программы вознаграждения за найденные уязвимости. Принцип ответственного раскрытия уязвимостей. Карьерные траектории в сфере информационной безопасности. Международные сертификации специалистов по ИБ.	3	Повторение теории, необходимой для выполнения практических заданий по правовым и этическим аспектам ИБ. Подробный разбор совместно с обучающимися решения задач по разбору этических дилемм и классификации хакеров. Выполнение практических заданий по анализу сценариев. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.

					методической разработке «Материалы для преподавателя по проведению практических занятий».		
4	Тема 4.2. Аудит кода: статический анализ и поиск уязвимостей	2	Статический анализ как метод выявления уязвимостей без выполнения кода. Типичные ошибки: жёстко заданные в коде секреты и ключи. Использование небезопасных источников случайности. Отсутствие валидации входных данных. Классификация уязвимостей по критичности. Составление отчёта с рекомендациями по устранению.	6	Повторение теории, необходимой для выполнения практических заданий по статическому анализу кода и поиску уязвимостей. Подробный разбор совместно с обучающимися решения задач по выявлению типовых ошибок. Выполнение практических заданий по аудиту кода, в том числе вариативное задание. Подробное описание практических заданий представлено в приложении к ДОП. Проведение диагностического опроса «Повторим главное» для повторения и закрепления полученных знаний. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.

					проведению практических занятий».		
4	Тема 4.3. Проектирование итогового проекта: модель угроз (threat model) и архитектура	1	Цели и формат финального проекта. Три направления проектов на выбор. Критерии выбора темы. Постановка задачи и определение ожидаемого результата. Планирование работы над проектом. Распределение ролей в команде. Критерии оценивания проекта. Модель угроз (threat model) как обязательный этап до написания кода. Определение защищаемых активов. Выявление источников угроз и сценариев атак. Архитектурные решения как ответ на выявленные угрозы.	3	Повторение теории, необходимой для выполнения практических заданий. Разбор с обучающимися примеров трёх направлений итогового проекта и критериев их оценки. Выбор обучающимися одного из вариантов проекта. Выполнение обучающимися практических заданий, в том числе по работе над проектом. Подробное описание практических заданий приложено к ДОП. Проведение проверочного тестирования для закрепления изученного материала. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».	2	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.
4	Тема 4.4. Разработка, отладка	1	Организация разработки проекта: итеративный	3	Повторение теории, необходимой для	4	Повторение и систематизация

	и статический анализ проекта		подход. Статический анализ собственного кода и устранение выявленных проблем. Документирование принятых решений и мер защиты. Описание модели угроз в README проекта. Парный code review как способ выявления уязвимостей. Исправление замечаний по результатам проверки.		выполнения практических заданий. Разбор с обучающимися примеров отладки кода и тестирования на граничных случаях. Выполнение обучающимися практических заданий по доработке собственного проекта, проверке обработки ошибок. Подробное описание практических заданий приложено к ДОП. Проведение проверочного тестирования для закрепления изученного материала. Дополнительные упражнения по теме представлены в методической разработке «Материалы для преподавателя по проведению практических занятий».		пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление теоретических знаний и практических умений через выполнение заданий для самостоятельной работы и самоконтроля/самопроверки с автоматической системой оценивания.
4	Тема 4.5. Оформление проекта	1	Документирование кода: структура и читаемость. Подготовка презентации проекта: архитектура, модель угроз, демонстрация. Публикация проекта с	3	Повторение теории, необходимой для выполнения практических заданий. Разбор с обучающимися примеров написания технической	1	Повторение и систематизация пройденного материала по ключевым понятиям и практическим навыкам темы. Закрепление

			использованием системы контроля версий. Файл .gitignore и исключение секретов из публичного доступа. Финальная проверка соответствия проекта требованиям безопасной разработки.		документации (README) и подготовки к публичному выступлению. Выполнение обучающимися практических заданий по финальной доработке проекта, оформлению файла README, репетиции защиты и подготовке ответов на технические вопросы комиссии. Подробное описание практических заданий приложено к ДОП. Проведение проверочного тестирования для закрепления изученного материала.		теоретических знаний и практических умений через выполнение заданий для самостоятельной работы с автоматической системой оценивания.
--	--	--	---	--	---	--	--

5. Формы аттестации и оценочные материалы, включая примеры контрольных заданий

5.1 Текущий контроль

В таблице ниже представлены по каждому из модулей:

- количество часов текущего контроля;
- формы текущего контроля с подробным описанием процедуры оценивания результатов обучения;
- диагностические инструменты с примерами (контрольные задания, материалы, промежуточные тесты и задачи);
- показатели и критерии оценивания;
- шкала оценивания.

Текущий контроль. Количество ак. часов	1	1	1	1
Текущий контроль. Формы контроля	Тестирование	Тестирование	Тестирование	Тестирование

Текущий контроль. Диагностические инструменты	Задание в виде теста. 1. Какое ключевое слово используется для объявления функции в Python? А) class Б) def В) func Г) return Правильный ответ - Б 2. Что выведет код <code>`print(2 ** 3)`</code>? А) 5 Б) 6 В) 8 Г) 9 Правильный ответ - В 3. Какой оператор используется для ветвления? А) if Б) for В) def Г) import Правильный ответ - А 4. Сколько раз выполнится цикл <code>`for i in range(4):`</code>? А) 3 Б) 4 В) 5 Г) 0 Правильный ответ - Б	Задание в виде теста. 1. Какая структура данных хранит пары ключ-значение? А) list Б) set В) dict Г) tuple Правильный ответ - В 2. Какой метод добавляет элемент в список в конец? А) append Б) add В) push Г) pop Правильный ответ - А 3. Как называется скрывание внутреннего состояния объекта? А) наследование Б) инкапсуляция В) рекурсия Г) компиляция Правильный ответ - Б 4. Какой модуль Python предназначен для криптостойкой случайности? А) random Б) math В) time Г) secrets Правильный ответ - Г	Задание в виде теста. 1. Что ищет алгоритм Евклида? А) НОК Б) НОД В) Сумму Г) Степень Правильный ответ - Б 2. Чему равен остаток от деления 29 на 6? А) 5 Б) 4 В) 3 Г) 2 Правильный ответ - А 3. Какие числа называются взаимно простыми? А) Их сумма простая Б) Их разность равна 1 В) Их НОД равен 1 Г) Они оба нечётные Правильный ответ - В 4. Что означает запись <code>`a mod n`</code>? А) Частное Б) Остаток В) Степень Г) Корень Правильный ответ - Б 5. Открытый вопрос: Найдите НОД чисел 18 и 24. Ответ: _____ Правильный ответ - 6	Задание в виде теста. 1. Что из перечисленного относится к симметричному шифрованию? А) HMAC Б) RSA В) SHA Г) AES Правильный ответ - Г 2. Что проверяет HMAC в сообщении? А) Размер шрифта Б) Целостность В) Цвет текста Г) Версию браузера Правильный ответ - Б 3. Какой ключ в RSA должен храниться в секрете? А) Закрытый Б) Открытый В) Общий Г) Временный Правильный ответ - А 4. Что общего у AES и RSA? А) Оба являются форматами файлов Б) Оба используются для защиты данных В) Оба не используют ключи Г) Оба только хэшируют Правильный ответ - Б 5. Сканер портов проверяет диапазон портов с 1 по 1024 включительно. На хосте найдено 8 открытых портов. Сколько
--	---	---	--	--

	5. Какой оператор сравнения используется для проверки равенства? Правильный ответ: ==	5. Чему равно значение выражения $17 \% 5$? Ответ: _____ Правильный ответ: 2		портов в этом диапазоне оказалось закрытым или фильтруемым? Ответ: _____ Правильный ответ: 1016
Текущий контроль. Описание процедуры оценивания Показатели и критерии оценивания	Критерий оценивания: правильность ответа на тестовый вопрос. Показатель оценивания: количество правильных ответов на тестовые задания. За каждый правильный ответ начисляется 1 балл. В случае неправильного ответа баллы не начисляются.	Критерий оценивания: правильность ответа на тестовый вопрос. Показатель оценивания: количество правильных ответов на тестовые задания. За каждый правильный ответ начисляется 1 балл. В случае неправильного ответа баллы не начисляются.	Критерий оценивания: правильность ответа на тестовый вопрос. Показатель оценивания: количество правильных ответов на тестовые задания. За каждый правильный ответ начисляется 1 балл. В случае неправильного ответа баллы не начисляются.	Критерий оценивания: правильность ответа на тестовый вопрос. Показатель оценивания: количество правильных ответов на тестовые задания. За каждый правильный ответ начисляется 1 балл. В случае неправильного ответа баллы не начисляются.
Текущий контроль. Шкала оценивания, нижнее значение	0	0	0	0
Текущий контроль. Шкала оценивания, верхнее значение	5	5	5	5
Текущий контроль. Шкала оценивания, минимальный проходной балл для успешной сдачи	3	3	3	3

5.2 Промежуточная аттестация

В таблице ниже представлены по каждому из модулей:

- количество часов промежуточной аттестации;
- формы контроля (промежуточной аттестации) с подробным описанием процедуры оценивания результатов обучения;

- диагностические инструменты с примерами (контрольные задания, материалы, промежуточные тесты и задачи);
- показатели и критерии оценивания;
- шкала оценивания.

Аттестация по итогам модуля. Количество ак. часов	1	1	1	1
Аттестация по итогам модуля. Формы контроля	Тестирование и выполнение практико-ориентированных заданий на написание программного кода	Тестирование и выполнение практико-ориентированных заданий на написание программного кода	Тестирование и выполнение практико-ориентированных заданий на написание программного кода	Тестирование и выполнение практико-ориентированных заданий на написание программного кода
Аттестация по итогам модуля. Диагностические инструменты	Тест и практико-ориентированные задания. 1. Что делает функция len()? А) Возвращает длину Б) Открывает файл В) Преобразует число Г) Создает список Правильный ответ - А 2. Какой оператор используется для присваивания? А) == Б) = В) != Г) >= Правильный ответ - Б 3. Что выведет print(2 + 3 * 4)? А) 20 Б) 14 В) 11 Г) 24 Правильный ответ - Б	Тест и практико-ориентированные задания. 1. Какой алгоритм используется для нахождения НОД двух чисел? А) Евклида Б) Дейкстры В) Прима Г) Флойда Правильный ответ - А 2. Какая криптосистема использует открытый и закрытый ключи? А) AES Б) RSA В) HMAC Г) ZIP Правильный ответ - Б 3. Что добавляют к паролю перед хешированием? А) соль Б) порт В) модуль	Тест и практико-ориентированные задания. 1. Какой протокол обеспечивает надёжную доставку данных? А) TCP Б) UDP В) ARP Г) ICMP Правильный ответ - А 2. Как называется веб-уязвимость внедрения скрипта на страницу? А) CSRF Б) XSS В) DNS Г) FTP Правильный ответ - Б 3. Как называется атака с последовательностями ../ ? А) traversal Б) spoofing В) hashing Г) flooding	Тест и практико-ориентированные задания. 1. Какой модуль Python подходит для работы с JSON? А) json Б) math В) time Г) sqlite Правильный ответ - А 2. Какой алгоритм из курса относится к асимметричной криптографии? А) HMAC Б) AES В) RSA Г) ECB Правильный ответ - В 3. Что чаще всего используют для автоматизации действий в Python? А) скрипт Б) картинку

4. Какие из перечисленных слов являются ключевыми словами Python? (выберите несколько вариантов) A) if B) while B) loop Г) def Правильный ответ - А, Б, Г 5. Какой будет результат у выражения $5 > 3$, если привести его к целочисленному типу (int)? Правильный ответ - 1 6. Какой оператор используется для получения остатка от деления? Правильный ответ - % 7. Какое число выведет вызов <code>len("id")</code>? Правильный ответ - 2 Тренажеры кода: 8. Система мониторинга получает целочисленный показатель нагрузки сервера n. На вход подаётся одно целое число n. На выход программа должна вывести квадрат этого числа (оценка квадратичной нагрузки). Пример входных данных: 4 Пример выходных данных: 16 Эталонный код решения:	Г) ключевое слово Правильный ответ - А 4. Какие алгоритмы используют ключи? A) RSA B) AES B) HMAC Г) HTML Правильный ответ - А, Б, В 5. Чему равен результат выражения <code>`17 mod 5`</code>? Правильный ответ - 2 6. Какими символами обозначается оператор вычисления остатка в Python? Правильный ответ - % 7. Сколько байтов содержит вывод функции SHA-256? Правильный ответ - 32 Тренажеры кода: 8. Криптомодуль обрабатывает два целых параметра протокола <code>`a`</code> и <code>`b`</code> и должен вычислить их наибольший общий делитель для проверки корректности ключей. На вход подаются два целых числа a и b. На выход программа должна вывести их наибольший общий делитель, найденный алгоритмом Евклида. Пример входных данных: 18	Правильный ответ - А 4. Что относится к сетевой безопасности и веб-уязвимостям? A) сокет Б) логи B) XSS Г) tuple Правильный ответ - А, Б, В 5. Какой код ответа HTTP означает успешный запрос (ОК)? Правильный ответ - 200 6. Какой оператор сравнения в Python проверяет равенство? Правильный ответ - == 7. Какой код ответа HTTP означает внутреннюю ошибку сервера? Правильный ответ - 500 Тренажеры кода: 8. В логах веб-приложения сохраняется полный текст запроса пользователя. Нужно быстро понять, попала ли запись с возможной XSS-вставкой. Напишите программу, которая считывает одну строку лога и выводит True, если в строке есть подстрока <code>`<script>`</code>, и False в противном случае. Пример входных данных:	В) архив Г) шрифт Правильный ответ - А 4. Что из перечисленного является структурой данных в Python? A) dict Б) list B) int Г) tuple Правильный ответ - А, Б, Г 5. Какой код ответа HTTP означает успешный запрос (ОК)? Правильный ответ - 200 6. Какой оператор сравнения в Python проверяет равенство? Правильный ответ - == 7. Какой код ответа HTTP означает внутреннюю ошибку сервера? Правильный ответ: 500 Тренажеры кода: 8. В панели администрирования хранятся данные профиля в формате JSON. Структура содержит поля <code>`user`</code> и <code>`role`</code>. Напишите программу, которая считывает эту JSON-строку и выводит значение поля <code>`role`</code>. Пример входных данных:
---	---	--	--

n = int(input()) print(n ** 2) 9. В журнале безопасности зафиксированы два целых значения: число срабатываний одного правила а и другого правила b. На вход подаются два целых числа, каждое с новой строки. На выход программа должна вывести большее из этих чисел (какое правило сработало чаще). Пример входных данных: 5 9 Пример выходных данных: 9 Эталонный код решения: a = int(input()) b = int(input()) print(max(a, b)) 10. Сканер безопасности должен по очереди проверить порты от 1 до n включительно. На вход подаётся одно целое число n. На выход программа должна вывести все числа от 1 до n включительно, каждое с новой строки (номера портов для проверки). Пример входных данных: 3	24 Пример выходных данных: 6 Эталонный код решения: a = int(input()) b = int(input()) while b: a, b = b, a % b print(a) 9. Система модульной арифметики получает два целых числа а и n и должна вычислить остаток от деления а на n (операция по модулю n). На вход подаются два числа. На выход программа должна вывести остаток от деления а на n. Пример входных данных: 17 5 Пример выходных данных: 2 Эталонный код решения: a = int(input()) n = int(input()) print(a % n) 10. Анализатор ключевых параметров получает два целых числа а и b и должен определить, являются ли они взаимно простыми (подходят ли для простейшей	GET /?q=<script>alert(1)</script> Пример выходных данных: True Эталонный код решения: line = input() print('<script>' in line) 9. Приложение принимает путь к файлу, который передаёт клиент. Вы хотите отфильтровать потенциальные попытки обхода директории через '..'. Напишите программу, которая считывает строку с путём и выводит True, если в пути встречается последовательность '..', и False иначе. Пример входных данных: ../etc/passwd Пример выходных данных: True Эталонный код решения: path = input() print('..' in path) 10. Конфигурация сервиса задаёт список «разрешённых» портов: 22 (SSH), 80 (HTTP), 443 (HTTPS). Вам дают номер порта, и нужно проверить, входит ли он	{"user": "ann", "role": "admin"} Пример выходных данных: Admin Эталонный код решения: import json data = json.loads(input()) print(data['role']) 9. Служба уведомлений добавляет к каждому сообщению криптографическую подпись HMAC-SHA256. Напишите программу, которая считывает текст уведомления и секретный ключ и выводит HMAC-SHA256 этого сообщения. Пример входных данных: hello key Пример выходных данных: 9307b3b915efb5171ff14d8cb55fbcc798c6c0ef1456d66ded1a6aa723a58b7b Эталонный код решения: import hmac, hashlib msg = input().encode() key = input().encode() print(hmac.new(key, msg, hashlib.sha256).hexdigest()) 10. В веб-приложение передают путь к загружаемому файлу. Перед обработкой нужно
--	---	--	--

	Пример выходных данных: 1 2 3 Эталонный код решения: n = int(input()) for i in range(1, n + 1): print(i) 11. Криптографический модуль получает два целых ключевых параметра. На вход подаются два целых числа. На выход программа должна вывести их сумму, вычисленную с помощью вызова функции (например, суммарный размер ключей). Пример входных данных: 2 7 Пример выходных данных: 9 Эталонный код решения: def add(a, b): return a + b a = int(input()) b = int(input()) print(add(a, b)) 12. Система анализа логов получает одну строку с сообщением. На вход подаётся одна строка. На выход программа должна	схемы RSA). На вход подаются числа a и b. На выход программа должна вывести True, если числа взаимно простые, и False в противном случае. Пример входных данных: 8 15 Пример выходных данных: True Эталонный код решения: a = int(input()) b = int(input()) x, y = a, b while y: x, y = y, x % y print(x == 1) 11. Учебная реализация RSA получает параметры a, b и n, где a — основание, b — показатель степени, n — модуль. На вход подаются три целых числа. На выход программа должна вывести значение выражения a в степени b по модулю n. Пример входных данных: 2 5 7 Пример выходных данных: 4 Эталонный код решения: a = int(input())	в этот список. Напишите программу, которая считывает целое число и выводит True, если порт равен 22, 80 или 443, и False иначе. Пример входных данных: 80 Пример выходных данных: True Эталонный код решения: port = int(input()) allowed = [22, 80, 443] print(port in allowed) 11. В настройках балансировщика трафика каждое backend-подключение описывается парой «IP-адрес и порт». Напишите программу, которая считывает IP-адрес (строка) и порт (целое число) и выводит словарь с ключами 'ip' и 'port'. Пример входных данных: 127.0.0.1 8080 Пример выходных данных: {'ip': '127.0.0.1', 'port': 8080} Эталонный код решения: ip = input() port = int(input()) print({'ip': ip, 'port': port})	проверить, нет ли попытки выхода из каталога через '..'. Напишите программу, которая считывает строку пути и выводит True, если в ней есть '..', иначе False. Пример входных данных: ../secret.txt Пример выходных данных: True Эталонный код решения: path = input() print('..' in path) 11. Аналитик хочет оценить, насколько разнообразен сгенерированный токен доступа. Напишите программу, которая считывает строку и выводит количество уникальных символов в ней. Пример входных данных: aba Пример выходных данных: 2 Эталонный код решения: s = input() print(len(set(s))) 12. В журнал безопасности попал HTML-фрагмент с разметкой, которая мешает чтению текста. Напишите программу, которая
--	---	---	---	---

	вывести количество символов в этой строке (длину сообщения журнала). Пример входных данных: code Пример выходных данных: 4 Эталонный код решения: s = input() print(len(s)) 13. Фильтр трафика получает целое число — размер пакета в байтах. На вход подаётся одно целое число. На выход программа должна вывести True, если число чётное, и False в противном случае. Пример входных данных: 8 Пример выходных данных: True Эталонный код решения: n = int(input()) print(n % 2 == 0) 14. Система подсчёта инцидентов получает три целых значения — количество предупреждений трёх разных типов. На вход подаются три целых числа, каждое с новой строки. На выход программа должна вывести сумму этих чисел	<pre>b = int(input()) n = int(input()) print(pow(a, b, n))</pre> 12. Система выбора рабочих модулей получает одно целое число n и должна найти все числа от 1 до n, которые взаимно просты с n (подходят как возможные экспоненты). На вход подаётся одно число. На выход программа должна вывести все числа от 1 до n, взаимно простые с n, каждое с новой строки. Пример входных данных: 6 Пример выходных данных: 1 5 Эталонный код решения: n = int(input()) for i in range(1, n + 1): a, b = i, n while b: a, b = b, a % b if a == 1: print(i) 13. Модуль проверки сложности пароля получает строку-пароль и должен проверить минимальную длину для корпоративного стандарта. На вход подаётся одна	12. Логи прокси содержат первую строку HTTP-запроса целиком. Нужно отличать запросы типа GET от остальных. Напишите программу, которая считывает одну строку HTTP-запроса и выводит True, если строка начинается с `GET`, и False в противном случае. Пример входных данных: GET /index HTTP/1.1 Пример выходных данных: True Эталонный код решения: request = input() print(request.startswith('GET')) 13. В файле log.txt лежит журнал работы сервиса за день. Вы хотите узнать, сколько раз за это время произошли ошибки, отмеченные словом `ERROR`. Напишите программу, которая открывает файл log.txt и выводит количество строк, содержащих `ERROR`. Пример выходных данных: 3 Эталонный код решения: count = 0	считывает строку и выводит её без символов `<` и `>`. Пример входных данных: hi Пример выходных данных: bhib Эталонный код решения: s = input() print(s.replace('<', '').replace('>', '')) 13. Система хранения учётных записей сохраняет от пароля только SHA-256 хэш. Напишите программу, которая считывает строку и выводит её SHA-256 хэш. Пример входных данных: pass123 Пример выходных данных: 9b8769a4a742959a2d0298c36fb70623a6a7afef2f1a8a6f6d7c6ad517b96206 Эталонный код решения: import hashlib print(hashlib.sha256(input().encode()).hexdigest()) 14. Система логирования выдаёт полный путь до файла отчёта о проверке безопасности. Напишите программу, которая считывает путь и выводит только имя
--	---	--	--	--

	(общее количество срабатываний). Пример входных данных: <pre>1 2 3</pre> Пример выходных данных: <pre>6</pre> Эталонный код решения: <pre>nums = [int(input()), int(input()), int(input())] print(sum(nums))</pre> 15. Система генерации одноразовых кодов использует модуль secrets. На вход данные не подаются. На выход программа должна вывести случайное целое число от 0 до 9 (одноразовый код). Пример выходных данных: <pre>7</pre> Эталонный код решения: <pre>import secrets print(secrets.randbelow(10))</pre>	строка. На выход программа должна вывести True, если длина пароля не меньше 8 символов, и False в противном случае. Пример входных данных: <pre>Qwerty12</pre> Пример выходных данных: <pre>True</pre> Эталонный код решения: <pre>password = input() print(len(password) >= 8)</pre> 14. Журнал аутентификации сохраняет строки событий и должен хранить их в виде SHA-256 хеша, чтобы не раскрывать исходные данные. На вход подаётся одна строка. На выход программа должна вывести её хеш SHA-256 в шестнадцатеричном виде. Пример входных данных: <pre>Abc</pre> Пример выходных данных: <pre>ba7816bf8f01cfea414140de5da e2223b00361a396177a9cb410f f61f20015ad</pre> Эталонный код решения: <pre>import hashlib s = input().encode() print(hashlib.sha256(s).hexdigest())</pre> 15. Сервис проверки целостности сообщений	with open('log.txt', 'r', encoding='utf-8') as f: <pre>for line in f: if 'ERROR' in line: count += 1 print(count)</pre> 14. Один микросервис отправляет в другой JSON-объект с адресом сервера базы данных: там есть поля `ip` и `port`. Нужен простой скрипт, который из такой JSON-строки достанет только IP-адрес. Напишите программу, которая считывает корректную JSON-строку и выводит значение поля `ip`. Пример входных данных: <pre>{ip: 10.0.0.1, port: 22}</pre> Пример выходных данных: <pre>10.0.0.1</pre> Эталонный код решения: <pre>import json data = json.loads(input()) print(data['ip'])</pre> 15. В логах иногда попадает сырая HTML-разметка с угловыми скобками, которая мешает чтению. Нужен простой фильтр, убирающий только сами скобки, оставляя текст. Напишите программу, которая	файла (часть после последнего символа `/`). Пример входных данных: <pre>/home/user/data.txt</pre> Пример выходных данных: <pre>data.txt</pre> Эталонный код решения: <pre>path = input() print(path.split('/')[-1])</pre> 15. В отчёте по сетевому сканированию некоторые строки содержат IPv4-адреса. Напишите программу, которая считывает строку и выводит True, если в ней ровно три точки. Пример входных данных: <pre>192.168.0.1</pre> Пример выходных данных: <pre>True</pre> Эталонный код решения: <pre>ip = input() print(ip.count('.') == 3)</pre>
--	--	--	--	--

		получает текст сообщения и секретный ключ и должен вычислить HMAC-SHA256 для хранения аудита. На вход подаются две строки. На выход программа должна вывести HMAC-SHA256 сообщения в шестнадцатеричном виде. Пример входных данных: hello key Пример выходных данных: 9307b3b915efb5171ff14d8cb55fbcc798c6c0ef1456d66ded1a6aa723a58b7b Эталонный код решения: import hmac, hashlib msg = input().encode() key = input().encode() print(hmac.new(key, msg, hashlib.sha256).hexdigest())	считывает строку и выводит её без символов '<' и '>'. Пример входных данных: <script> Пример выходных данных: script Эталонный код решения: s = input() print(s.replace('<', '').replace('>', '')) Правильный ответ - CSRF	
Аттестация по итогам модуля. Описание процедуры оценивания Показатели и критерии оценивания	Критерий оценивания: для тестовой части – правильность ответа на тестовый вопрос; для практико-ориентированных заданий – правильность работы программы. Показатели оценивания: для тестовой части – количество правильных ответов на тестовые задания; для практико-ориентированных	Критерий оценивания: для тестовой части – правильность ответа на тестовый вопрос; для практико-ориентированных заданий – правильность работы программы. Показатели оценивания: для тестовой части – количество правильных ответов на тестовые задания; для практико-ориентированных	Критерий оценивания: для тестовой части – правильность ответа на тестовый вопрос; для практико-ориентированных заданий – правильность работы программы. Показатели оценивания: для тестовой части – количество правильных ответов на тестовые задания; для практико-ориентированных	Критерий оценивания: для тестовой части – правильность ответа на тестовый вопрос; для практико-ориентированных заданий – правильность работы программы. Показатели оценивания: для тестовой части – количество правильных ответов на тестовые задания; для практико-ориентированных

	заданий – количество заданий, в которых программа работает правильно. За каждый правильный ответ/правильно выполненное практико-ориентированное задание начисляется 1 балл. В случае неправильного ответа/неправильно выполненное практико-ориентированное задание баллы не начисляются.	заданий – количество заданий, в которых программа работает правильно. За каждый правильный ответ/правильно выполненное практико-ориентированное задание начисляется 1 балл. В случае неправильного ответа/неправильно выполненное практико-ориентированное задание баллы не начисляются.	заданий – количество заданий, в которых программа работает правильно. За каждый правильный ответ/правильно выполненное практико-ориентированное задание начисляется 1 балл. В случае неправильного ответа/неправильно выполненное практико-ориентированное задание баллы не начисляются.	заданий – количество заданий, в которых программа работает правильно. За каждый правильный ответ/правильно выполненное практико-ориентированное задание начисляется 1 балл. В случае неправильного ответа/неправильно выполненное практико-ориентированное задание баллы не начисляются.
Аттестация по итогам модуля. Шкала оценивания, нижнее значение	0	0	0	0
Аттестация по итогам модуля. Шкала оценивания, верхнее значение	15	15	15	15
Аттестация по итогам модуля. Шкала оценивания, минимальный проходной балл для успешной сдачи	8	8	8	8

5.3 Итоговый контроль

Итоговый контроль. Количество академических часов	1
Итоговый контроль. Формы контроля	Комплексное задание: тест и защита проекта. Итоговый контроль не входит в 4 модуль (является отдельным элементом ДОП).
Итоговый контроль. Диагностические инструменты	Комплект заданий: тестовые вопросы и процедура защиты проекта
Итоговый контроль. Показатели и критерии оценивания	Показатели и критерии (с весами) оценивания: Этап 1. Тестирование Тестовая часть итогового контроля выполняется и проверяется автоматизированно на Образовательной платформе. Каждый верный ответ оценивается в 1 балл, неверный или отсутствующий — 0 баллов. Максимальный балл 1 этапа – 10 Минимальный пороговый балл – 5 Этап 2. Защита проекта (с презентацией) Критерии и веса внутри проектной части (каждый критерий оценивается членами комиссии по шкале от 0 до 10): 1. Содержательная часть (качество продукта, глубина проработки) - вес 0,7 0 – 4 - Код работает с ошибками или не запускается. Реализована только часть заявленной функциональности (менее 50% функций). Отсутствуют ключевые функции. Нарушены требования безопасности. Код нечитаем: отсутствуют комментарии, осмысленные имена переменных, логическая структура. Не обработаны ошибки и граничные случаи. Описание проекта отсутствует или содержит минимум информации. Проект не демонстрирует понимания темы. 5 – 7 – Код запускается и выполняет основные функции без критических ошибок. Реализованы все минимально необходимые функции для проекта. Соблюдены базовые требования безопасности. Код в целом читаем, но есть замечания к структуре или именованию переменных. Обработаны основные ошибки, но не все граничные случаи. Описание проекта присутствует, содержит описание проекта и примеры использования. Проект демонстрирует базовое понимание темы и способность применить изученные инструменты. Структура проекта логична, но без дополнительных улучшений. 8 – 10 - Код работает без ошибок, все функции выполняют заявленную функциональность корректно. Реализованы все заявленные функции + дополнительные улучшения. Строго соблюдены требования безопасности. Код читаемый: осмысленные имена переменных, логическая структура, отступы, комментарии к функциям и ключевым блокам. Полностью обработаны ошибки и граничные случаи (программа корректно работает при любом вводе). Описание проекта содержит инструкции по запуску, примеры использования. Проект демонстрирует глубокое понимание

	темы, творческий подход и способность применять знания на практике. Структура проекта логична и соответствует лучшим практикам разработки. Обучающийся уверенно отвечает на технические вопросы и обосновывает принятые решения. 2. Визуальное оформление презентации - вес 0,3 0 – 4 - Слайды перегружены текстом; нет единого стиля; шрифты нечитаемы; изображения низкого качества; есть ошибки в оформлении. 5 – 7 – Текст и графика сбалансированы; единый стиль прослеживается; информация структурирована; визуализация данных присутствует. 8 – 10 - Дизайн эстетичный и функциональный; все визуальные элементы работают на раскрытие темы. Максимальный балл 2 этапа – 10 Минимальный пороговый балл – 6 Алгоритм расчета оценки/балла итогового контроля по ДОП Итоговый показатель аттестации рассчитывается в 20-балльной шкале как сумма результатов обоих этапов: Максимально возможное значение: 20 баллов.
Итоговый контроль. Примеры заданий	Вариант 1 1. Тест. Ответьте на тестовые вопросы, выбрав один из предложенных вариантов ответа. В отдельных вопросах нужно будет ввести свой ответ в поле для ответа. 1. Какое ключевое слово используется для объявления функции в Python? А) class Б) def В) import Г) lambda Правильный ответ - Б 2. Какой алгоритм используется для нахождения НОД двух чисел? А) Евклида Б) Дейкстры В) Флойда Г) Прима Правильный ответ - А 3. Какой протокол обеспечивает надёжную доставку данных? А) UDP Б) TCP В) DNS Г) ICMP Правильный ответ - Б 4. Какая структура данных хранит пары ключ-значение? А) list Б) dict В) set Г) tuple Правильный ответ - Б

	5. Как называется веб-уязвимость внедрения скрипта на страницу? А) XSS Б) CSRF В) ECB Г) HMAC Правильный ответ - А 6. Какой алгоритм относится к симметричному шифрованию? А) RSA Б) AES В) HMAC Г) SHA Правильный ответ - Б 7. Какие из перечисленных объектов Python являются изменяемыми? А) list Б) dict В) tuple Г) set Правильный ответ - А, Б, Г 8. Какой код ответа HTTP означает внутреннюю ошибку сервера? Правильный ответ - 500 9. Сколько бит в ключе AES-128? Правильный ответ - 128 10. Что добавляют к паролю перед хэшированием для усиления защиты? А) соль Б) индекс В) цикл Г) комментарий Правильный ответ - А 2. Защита проекта. Представьте с использованием презентационных материалов результаты проектной работы на курсе. Полный комплект оценочных материалов (7 вариантов заданий) представлен в документе «Оценочные материалы для проведения итогового контроля» в составе УМК в приложении к ДОП.
Итоговый контроль. Шкала оценивания, нижнее значение	0
Итоговый контроль. Шкала оценивания, верхнее значение	20
Итоговый контроль. Шкала оценивания, минимальный проходной балл	11

6. Организационно-педагогические условия

6.1 Методическое обеспечение: методы, формы и технологии, применяемые при реализации программы

Методы обучения: модульный, проблемный и контекстный подходы.

Формы занятий: лекции с демонстрацией мультимедийных материалов, практические работы, самостоятельная работа.

Технологии: традиционные (объяснительно-иллюстративные) и интерактивные (групповые обсуждения, решение кейсов).

6.2 Материально-техническое обеспечение

Наименование требуемого оборудования и программного обеспечения:

Рабочее место обучающегося: оборудование и мебель, необходимые для обучения, включая расходные материалы.

Рабочее место педагога: оборудование (включая демонстрационное) и мебель.

Иное оборудование и мебель.

Сетевое оборудование и подключение к сети Интернет.

Программное обеспечение.

Средства индивидуальной защиты и охраны труда.

Зонирование площадки.

Компьютер/ноутбук

Windows/Linux (не ранее 2020)

Интернет от 2 Мбит/сек

Веб-браузер (не ранее 2020)

Микрофон, наушники и колонки

Клавиатура, мышь

Проектор/интерактивная доска

7. Учебно-методические материалы

Ниже представлен перечень методических разработок и материалов для обучения:

Методические разработки

Пакет методических разработок содержит:

- Методическую разработку «Материалы для преподавателя по проведению практических занятий».
- Методическую разработку «Материалы для преподавателя по организации самостоятельной работы и самоконтроля/самопроверки».
- Оценочные материалы текущего контроля и промежуточной аттестации.
- Оценочные материалы для проведения итогового контроля по ДОП

Материалы для обучения

Пакет материалов модуля содержит:

- Презентации для теоретических занятий.
- Презентации для практических занятий.
- Описание практических заданий.
- Задания для самоконтроля (самопроверки).
- Задания для самостоятельной работы.

- Задания системы контроля: текущего контроля, промежуточной аттестации.

8. Перечень источников информационного сопровождения (учебная литература и др.)

Учебная литература:

1. Солдатова Г. У., Чигарькова С. В., Пермякова И. Д. Кибербезопасность: учебное пособие для старших классов общеобразовательных организаций: в 2 ч. Ч. 1. — Москва: Русское слово-учебник, 2022. — 80 с. — ISBN 978-5-533-02111-1.
2. Солдатова Г. У., Чигарькова С. В., Пермякова И. Д. Кибербезопасность: учебное пособие для старших классов общеобразовательных организаций: в 2 ч. Ч. 2. — Москва: Русское слово-учебник, 2022. — 112 с. — ISBN 978-5-533-02112-8.
3. Цветкова М. С., Голубчиков С. В., Новиков В. К. Информационная безопасность. Правовые основы информационной безопасности. 10–11 класс: учебник. — Москва: Просвещение, 2021. — 112 с. — ISBN 978-5-09-085309-5.
4. Заколдаев Д. А., Петренко С. А., Уваров В. А. и др. Безопасность в цифровом пространстве. 10–11 классы: учебное пособие / под ред. В. В. Минина, С. А. Петренко. — Москва: Просвещение, 2023. — 144 с. — ISBN 978-5-09-107129-7.

Электронные информационные ресурсы:

1. Портал «КиберЗОЖ» <https://киберзож.рф>
2. Образовательная платформа по Python <https://pythonist.ru/>

Директор Центра компетенций
«Цифровая экономика»



Н.Ю. Сурова