



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И.  
Носова»



УТВЕРЖДАЮ  
Директор ИЭиАС  
В.Р. Храмшин

03.02.2026 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

***ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ***

Направление подготовки (специальность)

10.05.03 Информационная безопасность автоматизированных систем

Направленность (профиль/специализация) программы

10.05.03 специализация N 8 "Разработка автоматизированных систем в защищенном  
исполнении"

Уровень высшего образования - специалитет

Форма обучения  
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Информатики и информационной безопасности
Курс	2
Семестр	4

Магнитогорск  
2026 год

Рабочая программа составлена на основе ФГОС ВО - специалитет по специальности 10.05.03 Информационная безопасность автоматизированных систем (приказ Минобрнауки России от 26.11.2020 г. № 1457)

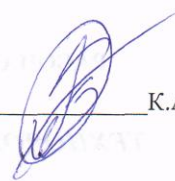
Рабочая программа рассмотрена и одобрена на заседании кафедры Информатики и информационной безопасности  
22.01.2026, протокол № 5

Зав. кафедрой  И.И. Баранкова

Рабочая программа одобрена методической комиссией ИЭиАС  
03.02.2026 г. протокол № 5

Председатель  В.Р. Храмшин

Рабочая программа составлена:  
ст. преподаватель кафедры ИиИБ,  Ю.А. Мазнина

Рецензент:  
проректор по цифровизации, канд. техн. наук  К.А. Рубан

## Лист актуализации рабочей программы

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Информатики и информационной безопасности

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ И.И. Баранкова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Информатики и информационной безопасности

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ И.И. Баранкова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2029 - 2030 учебном году на заседании кафедры Информатики и информационной безопасности

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ И.И. Баранкова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2030 - 2031 учебном году на заседании кафедры Информатики и информационной безопасности

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ И.И. Баранкова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2031 - 2032 учебном году на заседании кафедры Информатики и информационной безопасности

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ И.И. Баранкова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2032 - 2033 учебном году на заседании кафедры Информатики и информационной безопасности

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ И.И. Баранкова

## 1 Цели освоения дисциплины (модуля)

Целью дисциплины «Технологии и методы программирования» является изучение основ современных методов и средств программирования и формировании у обучающихся навыков их практического применения в соответствии с требованиями ФГОС ВО для специальности «Информационная безопасность автоматизированных систем».

Дисциплина «Технологии и методы программирования» рассматривает основные подходы к проектированию программных средств, предполагает освоение методологий структурного, процедурного, функционального, объектно-ориентированного программирования, веб-программирования, а также методов тестирования и отладки программ.

## 2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Технологии и методы программирования входит в обязательную часть учебного плана образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Организация ЭВМ и вычислительных систем

Информатика

Языки программирования

Информационные технологии. Базы данных

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Технология построения защищенных распределенных приложений

Разработка и эксплуатация автоматизированных систем в защищенном исполнении

Разработка Siem систем

Основы Data инжиниринга

Защита программного обеспечения

Методы искусственного интеллекта

Защита электронного документооборота

Пентестинг

## 3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Технологии и методы программирования» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ОПК-7	Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ;
ОПК-7.1	Реализует базовые алгоритмы средствами языков программирования.
ОПК-7.2	Проектирует программное обеспечение с использованием средств автоматизации
ОПК-7.3	Проводит комплексное тестирование и отладку программных систем

#### 4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 академических часов, в том числе:

- контактная работа – 71,15 академических часов;
- аудиторная – 68 академических часов;
- внеаудиторная – 3,15 академических часов;
- самостоятельная работа – 37,15 академических часов;
- в форме практической подготовки – 0 академических часов;
- подготовка к экзамену – 35,7 академических часов

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Методы программирования на Python								
1.1 Языки и парадигмы программирования. Инструментальные средства для разработки программного обеспечения. Компиляторы и интерпретаторы. Исходный код, машинный код, байт-код	4	1				Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка к тестированию	Тестирование	ОПК-7.1
1.2 Работа с файлами и файловой системой в Python. Исключения в Python. Генерирование и перехват исключений		1		2	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала.	Практические работы	ОПК-7.1

						Подготовка и выполнение практических работ		
1.3 Коллекции в языке программирования Python. Стандартные коллекции: списки, кортежи, множества, словари. Модуль collections. Специализированные структуры данных: стек, очередь, дек, разреженный массив		1		2	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю	Практические работы. Индивидуальное домашнее задание. Рубежный контроль	ОПК-7.1, ОПК-7.2, ОПК-7.3
1.4 Функции в языке программирования Python. Область видимости переменных. Именованные параметры функций. Значения параметров по умолчанию. Функции с переменным числом аргументов, лямбда-функции	4	1		2	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ	Индивидуальное домашнее задание	ОПК-7.1
1.5 Функциональное программирование в Python: функции как объект, декораторы функций, списковые включения, встроенные функции высших порядков, замыкания, функторы, итераторы и итерируемые объекты, функции-генераторы, оператор yield		1		2	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками.	Практические работы. Индивидуальное домашнее задание. Рубежный контроль	ОПК-7.1

						Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю		
1.6 Особенности объектно-ориентированного подхода в Python: реализация инкапсуляции, наследования и полиморфизма средствами языка Python. Создание собственных классов исключений, менеджеров контекста	4	1		2	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю	Практические работы. Индивидуальное домашнее задание. Рубежный контроль	ОПК-7.1
Итого по разделу		6		10	10			
2. Обеспечение качества программного обеспечения								
2.1 Основные принципы обеспечения качества программного обеспечения. Стандарты ISO 9000	4	0,5				Самостоятельное изучение учебной и научной литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию	Тестирование	ОПК-7.2, ОПК-7.3
2.2 Тестирование и отладка программного обеспечения. Разработка через тестирование		0,5		2	2	Самостоятельное изучение учебной и научной литературы,	Тестирование	ОПК-7.2, ОПК-7.3

						спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка к тестированию		
2.3 Практики качественной и безопасной разработки ПО: логирование, конфигурационные файлы, управление секретами, статический анализ кода, динамический анализ безопасности, управление зависимостями (виртуальное окружение, фиксация версий библиотек, аудит зависимостей), code review, рефакторинг кода	4	1		4	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю	Практические работы. Индивидуальное домашнее задание. Рубежный контроль	ОПК-7.3
Итого по разделу		2		6	4			
3. Введение в технологию разработки программного обеспечения								
3.1 Типы архитектур программного обеспечения: монолитная архитектура, клиент-серверная архитектура, сервис-ориентированная архитектура, микросервисная архитектура программного обеспечения, облачная архитектура. Асинхронность в приложениях. Методы параллельного выполнения программ	4	0,5				Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка к тестированию	Тестирование	ОПК-7.2

3.2 Жизненный цикл разработки программного обеспечения. Модели жизненного цикла. Коллективная разработка ПО		0,5				Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка к тестированию	Тестирование	ОПК-7.2
3.3 Анализ требований. Разработка технического задания. Проектирование программного обеспечения: проектирование архитектуры, устройства компонент, пользовательского интерфейса	4	1		4	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю	Практические работы. Индивидуальное домашнее задание. Рубежный контроль	ОПК-7.2
Итого по разделу	2		4	2				
4. Архитектура и средства разработки web-приложений								
4.1 Основные виды веб-приложений и их архитектура. Смена подходов к разработке веб-приложений. Концепция Web 2.0. Понятие веб-сервиса. Построение веб-сервисов с использованием протокола SOAP и архитектуры REST	4	0,5				Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного	Тестирование	ОПК-7.2

						о портала. Подготовка к тестированию		
4.2 Протокол HTTP: назначение, принцип работы, виды HTTP-запросов. API как средство интеграции приложений. Работа с протоколом HTTP и API средствами языка программирования Python (библиотека requests)		0,5		2	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка к тестированию	Тестирование	ОПК-7.1, ОПК-7.2, ОПК-7.3
4.3 Основные инструменты для разработки веб-приложений: HTML, CSS, JavaScript, библиотека jQuery. Подход AJAX	4	0,5		4	2	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка к тестированию	Тестирование	ОПК-7.1, ОПК-7.2, ОПК-7.3
4.4 Фреймворки для веб-разработки. Модель MVC		0,5				Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка к тестированию	Тестирование	ОПК-7.2
Итого по разделу		2		6	4			
5. Разработка веб-приложений с использованием фреймворка FastAPI								
5.1 Особенности	4	0,5		4	2	Самостоятельно	Практические	ОПК-7.2,

<p>фреймворка FastAPI. Создание и настройка проекта FastAPI. Работа с конечными точками и параметрами в FastAPI. Обзор автоматической документации FastAPI</p>						<p>е изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю</p>	<p>работы. Индивидуальное домашнее задание. Рубежный контроль</p>	<p>ОПК-7.3</p>
<p>5.2 Структурирование проекта FastAPI. Маршрутизация и обработка запросов. Асинхронность и middleware</p>	<p>4</p>	<p>1</p>		<p>4</p>	<p>2</p>	<p>Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю</p>	<p>Практические работы. Индивидуальное домашнее задание. Рубежный контроль</p>	<p>ОПК-7.2, ОПК-7.3</p>
<p>5.3 Модели Pydantic, валидация данных</p>		<p>0,5</p>		<p>4</p>	<p>2</p>	<p>Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с</p>	<p>Практические работы. Индивидуальное домашнее задание. Рубежный контроль</p>	<p>ОПК-7.2, ОПК-7.3</p>

						ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю		
5.4 Интеграция баз данных	4	1		4	3	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ. Выполнение индивидуального домашнего задания. Подготовка к рубежному контролю	Практические работы. Индивидуальное домашнее задание. Рубежный контроль	ОПК-7.2, ОПК-7.3
5.5 Обработка ошибок. Тестирование приложений FastAPI		0,5		4	3	Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала. Подготовка и выполнение практических работ.	Практические работы. Индивидуальное домашнее задание. Рубежный контроль	ОПК-7.2, ОПК-7.3

						<p>работ.</p> <p>Выполнение индивидуального домашнего задания.</p> <p>Подготовка к рубежному контролю</p>		
<p>5.6 Принципы безопасной разработки в проектах FastAPI. Аутентификация и авторизация в проектах FastAPI. Внедрение зависимостей</p>		1		4	3	<p>Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками.</p> <p>Работа с материалами образовательного портала.</p> <p>Подготовка и выполнение практических работ.</p> <p>Выполнение индивидуального домашнего задания.</p> <p>Подготовка к рубежному контролю</p>	<p>Практические работы.</p> <p>Индивидуальное домашнее задание.</p> <p>Рубежный контроль</p>	<p>ОПК-7.2,</p> <p>ОПК-7.3</p>
<p>5.7 Развертывание приложений FastAPI</p>	4	0,5		1	2,15	<p>Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками.</p> <p>Работа с материалами образовательного портала.</p> <p>Подготовка и выполнение практических работ.</p> <p>Выполнение индивидуального домашнего задания.</p> <p>Подготовка к рубежному контролю</p>	<p>Практические работы.</p> <p>Индивидуальное домашнее задание.</p> <p>Рубежный контроль</p>	<p>ОПК-7.2,</p> <p>ОПК-7.3</p>

Итого по разделу	5		25	17,15			
6. Аттестация							
6.1 Экзамен	4				Самостоятельное изучение учебной и научной литературы, спецификации и документации языка программирования, работа с ЭБС и интернет-источниками. Работа с материалами образовательного портала	Экзамен	ОПК-7.1, ОПК-7.2, ОПК-7.3
Итого по разделу							
Итого за семестр	17		51	37,15		экзамен	
Итого по дисциплине	17		51	37,15		экзамен	

## 5 Образовательные технологии

Для реализации предусмотренных видов учебной работы в качестве образовательных технологий в преподавании дисциплины «Базы данных» используются традиционная и модульно-компетентностная технологии.

Реализация компетентностного подхода предусматривает использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

При проведении учебных занятий преподаватель обеспечивает развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств посредством проведения интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализа ситуаций, учета особенностей профессиональной деятельности выпускников и потребностей работодателей.

Формы учебных занятий с использованием традиционных технологий:

- обзорные лекции – для рассмотрения общих вопросов информатики и информационных технологий, для систематизации и закрепления знаний;
- информационные – для ознакомления с техническими средствами реализации информационных процессов, со стандартами организации сетей, основными приемами защиты информации, и другой справочной информацией;
- лекции-визуализации – для наглядного представления способов решения алгоритмических и функциональных задач, визуализации результатов решения задач;
- семинар – практическое занятие, посвященное освоению конкретных умений и навыков по предложенному алгоритму.

Формы учебных занятий с использованием технологий проблемного обучения:

- проблемная лекция – изложение материала, предполагающее постановку проблемных и дискуссионных вопросов, освещение различных научных подходов, авторские комментарии, связанные с различными моделями интерпретации изучаемого материала (для развития исследовательских навыков и изучения способов решения задач);
- лекции с заранее запланированными ошибками – направленные на поиск обучающимися синтаксических и алгоритмических ошибок при решении алгоритмических и функциональных задач, с последующей диагностикой слушателей и разбором сделанных ошибок;
- практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от обучающегося применения как научно-теоретических знаний, так и практических навыков;
- практическое занятие на основе кейс-метода – обучение в контексте моделируемой ситуации, воспроизводящей реальные условия научной, производственной, общественной деятельности; обучающиеся должны проанализировать ситуацию, разобраться в сути проблем, предложить возможные решения и выбрать лучшее из них; кейсы базируются на реальном фактическом материале или же приближены к реальной ситуации;
- подготовка тематических рефератов, содержащих разделы, частично или полностью выносимые на самостоятельное изучение.

Формы учебных занятий с использованием игровых технологий:

- учебная игра – форма воссоздания предметного и социального содержания будущей профессиональной деятельности специалиста, моделирования таких систем отношений, которые характерны для этой деятельности как целого;

– деловая игра – моделирование различных ситуаций, связанных с выработкой и принятием совместных решений, обсуждением вопросов в режиме «мозгового штурма», реконструкцией функционального взаимодействия в коллективе и т.п.

Технологии проектного обучения:

– творческий проект – учебно-познавательная деятельность обучающихся осуществляется в рамках рамочного задания, подчиняясь логике и интересам участников проекта, жанру конечного результата (газета, фильм, праздник, издание, экскурсия, подготовка заданий конкурсов и т.п.);

– информационный проект – учебно-познавательная деятельность с ярко выраженной эвристической направленностью (поиск, отбор и систематизация информации о каком-то объекте, ознакомление участников проекта с этой информацией, ее анализ и обобщение для презентации более широкой аудитории).

## **6 Учебно-методическое обеспечение самостоятельной работы обучающихся**

Представлено в приложении 1.

## **7 Оценочные средства для проведения промежуточной аттестации**

Представлены в приложении 2.

## **8 Учебно-методическое и информационное обеспечение дисциплины**

### **а) Основная литература:**

1. Гагарина, Л. Г. Основы проектирования и разработки информационных систем : учебное пособие / Л.Г. Гагарина, Ю.С. Шевнина. — Москва : ИНФРА-М, 2024. — 211 с. — (Высшее образование). — DOI 10.12737/1872684. - ISBN 978-5-16-017759-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/1872684> (дата обращения: 10.03.2026). – Режим доступа: по подписке.
2. Гниденко, И. Г. Технологии и методы программирования : учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 241 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/583784> (дата обращения: 10.03.2026).
3. Карякин, М. И. Технологии программирования и компьютерный практикум на языке Python : учебное пособие / М. И. Карякин, К. А. Ватульян, Р. М. Мнухин ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2022. - 242 с. - ISBN 978-5-9275-4108-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2057604> (дата обращения: 10.03.2025). – Режим доступа: по подписке.
4. Конструирование программного обеспечения : учебное пособие / под ред. Л.Г. Гагариной. — Москва : ИНФРА-М, 2024. — 319 с. — (Высшее образование). — DOI 10.12737/1893880. - ISBN 978-5-16-017861-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1893880> (дата обращения: 10.03.2026). – Режим доступа: по подписке
5. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561885> (дата обращения: 10.03.2026).
6. Маркин, А. В. Программирование на SQL : учебник и практикум для вузов / А. В. Маркин. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 805 с. — (Высшее образование). — ISBN 978-5-534-18371-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/589589> (дата обращения: 10.03.2026).

7. Федоров, Д. Ю. Программирование на Python : учебное пособие для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 187 с. — (Высшее образование). — ISBN 978-5-534-19666-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/585806> (дата обращения: 10.03.2026).

8. Черткова, Е. А. Компьютерные технологии обучения : учебник для вузов / Е. А. Черткова. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2026. — 245 с. — (Высшее образование). — ISBN 978-5-534-12532-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/584740> (дата обращения: 10.03.2026).

#### **б) Дополнительная литература:**

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, 2024. — 400 с. — (Высшее образование). - ISBN 978-5-8199-0707-8. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2149181> (дата обращения: 10.03.2026). – Режим доступа: по подписке.

2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2026. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2216924> (дата обращения: 10.03.2026). – Режим доступа: по подписке

3. Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2026. — 216 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-16-015638-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2216925> (дата обращения: 10.03.2026). – Режим доступа: по подписке.

4. Зыков, С. В. Объектно-ориентированное программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 151 с. — (Высшее образование). — ISBN 978-5-534-16941-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/584131> (дата обращения: 10.03.2026).

5. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 150 с. — (Высшее образование). — ISBN 978-5-534-16942-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/584399> (дата обращения: 10.03.2026).

6. Нестеров, С. А. Базы данных : учебник и практикум для вузов / С. А. Нестеров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 258 с. — (Высшее образование). — ISBN 978-5-534-18107-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/583591> (дата обращения: 10.03.2026).

7. Чернышев, С. А. Основы программирования на Python : учебник для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 349 с. — (Высшее образование). — ISBN 978-5-534-17139-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588667> (дата обращения: 10.03.2026).

#### **в) Методические указания:**

Методические указания по выполнению практических работ представлены в приложении 3.

Методические указания по выполнению внеаудиторных самостоятельных работ представлены в приложении 4.

**г) Программное обеспечение и Интернет-ресурсы:**

**Программное обеспечение**

Наименование ПО	№ договора	Срок действия лицензии
7Zip	свободно распространяемое ПО	бессрочно
Adobe Dreamweaver CS 5 Academic Edition	К-113-11 от 11.04.2011	бессрочно
Anaconda Python	свободно распространяемое ПО	бессрочно
MS SQL Server Management Studio	свободно распространяемое ПО	бессрочно
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно
Adobe Reader	свободно распространяемое ПО	бессрочно
Браузер Mozilla Firefox	свободно распространяемое ПО	бессрочно
Браузер Yandex	свободно распространяемое ПО	бессрочно
MS Office 2003 Professional	№ 135 от 17.09.2007	бессрочно
Git	свободно распространяемое ПО	бессрочно
MS Visual Studio Code	свободно распространяемое ПО	бессрочно
JetBrains PyCharm Community Edition	свободно распространяемое ПО	бессрочно
NotePad++	свободно распространяемое ПО	бессрочно
FAR Manager	свободно распространяемое ПО	бессрочно
Double Commander	свободно распространяемое ПО	бессрочно
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно

**Профессиональные базы данных и информационные справочные системы**

Название курса	Ссылка
Информационная система - Банк данных угроз безопасности информации ФСТЭК России	<a href="https://bdu.fstec.ru/?ysclid=lujkqy7cnw630508962">https://bdu.fstec.ru/?ysclid=lujkqy7cnw630508962</a>
Информационная система - Нормативные правовые акты, организационно-распорядительные документы, нормативные и методические документы и подготовленные проекты документов по технической защите информации ФСТЭК России	<a href="https://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-tzi?ysclid=lujknksfy724757053">https://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty-tzi?ysclid=lujknksfy724757053</a>
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: <a href="https://elibrary.ru/project_risc.asp">https://elibrary.ru/project_risc.asp</a>

Электронная база периодических изданий ООО «ИВИС»	<a href="https://eivis.ru/">https://eivis.ru/</a>
Российская Государственная библиотека. Каталоги	<a href="https://www.rsl.ru/ru/4readers/catalogues/">https://www.rsl.ru/ru/4readers/catalogues/</a>
Электронные ресурсы библиотеки МГТУ им. Г.И. Носова	<a href="https://host.megaprolib.net/MP0109/Web">https://host.megaprolib.net/MP0109/Web</a>
Международная реферативная и полнотекстовая справочная база данных научных изданий «Springer Nature»	<a href="https://www.nature.com/siteindex">https://www.nature.com/siteindex</a>

### **9 Материально-техническое обеспечение дисциплины (модуля)**

Материально-техническое обеспечение дисциплины включает:

Материально-техническое обеспечение дисциплины включает:

Лекционные аудитории:

- мультимедийные средства хранения, передачи и представления информации.

Учебные аудитории для проведения практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации:

- персональные компьютеры с ПО, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

Помещения для самостоятельной работы обучающихся:

- персональные компьютеры с ПО, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.

### Учебно-методическое обеспечение самостоятельной работы обучающихся

Аудиторная самостоятельная работа обучающихся на практических занятиях осуществляется под контролем преподавателя в виде решения задач и выполнения упражнений, которые определяет преподаватель для обучающегося с использованием методов ИТ .

Внеаудиторная самостоятельная работа обучающихся осуществляется в виде чтения литературы по соответствующему разделу с проработкой материала и выполнения домашних заданий с консультациями преподавателя, а также с применением кейс-технологий.

#### Примеры заданий для практических работ

**Задание.** Реализуйте сложение и скалярное умножение векторов на языке Python, используя функции высшего порядка zip, map, reduce и др. Если длина векторов не совпадает, должно возбуждаться исключение.

**Задание.** Используя фреймворк Flask, создайте веб-приложение, которая будет выводить на экран полный адрес и координаты краеведческого музея г. Магнитогорска, используя API Яндекс Карт.

#### Примеры заданий к рубежному контролю и индивидуальных домашних заданий (ИДЗ)

**Задание.** Вам предстоит реализовать несколько классов, вычисляющих суммы последовательностей натуральных чисел от 1 до N:

$$1 + 2 + \dots + N$$

$$1 + 2^2 + \dots + N^2$$

$$1 + 2^3 + \dots + N^3$$

Используйте наследование. Базовым классом будет Сумматор, который реализует методы:

1. преобразование элементов последовательности (какое это преобразование можно заметить из последовательности последовательностей);
2. вычисление суммы последовательности (хранить элементы последовательности не нужно, используйте метод из п.1);
3. проверка вычисленной суммы последовательности по формулам:

$$1 + 2 + \dots + N = \frac{N(N - 1)}{2}$$

$$1 + 2^2 + \dots + N^2 = \frac{N(N - 1)(2N + 1)}{6}$$

$$1 + 2^3 + \dots + N^3 = \left(\frac{N(N + 1)}{2}\right)^2$$

В производных классах допускается переопределить методы 1 и 3. Метод 2 переопределять не нужно.

Подсказка: метод 1 в базовом классе может осуществлять тождественное преобразование, то есть modify(n) = n.

В основной программе создайте несколько экземпляров созданных классов, вычислите сумму и сделайте проверку.

**Задание.** Создайте клиент-серверное веб-приложение для отгадывания загадок. Предусмотрите 2 вида пользователей: у администратора будет возможность загружать загадки, а у пользователей – возможность их отгадывать. Для реализации бэкенда используйте фреймворк FastAPI. Технологии реализации фронтенда выберите самостоятельно.

## Вопросы к экзамену

### Раздел 1. Методы программирования на Python

1. Основные парадигмы программирования.
2. Эволюция языков программирования.
3. Компиляторы и интерпретаторы. Виртуальные машины для исполнения байт-кода. JIT-компиляторы.
4. Инструментальные средства разработки программного обеспечения: назначение, разновидности, краткая характеристика, примеры.
5. Интегрированная среда разработки: область применения, состав, функции, примеры.
6. Особенности языка программирования Python. Соглашения по оформлению кода PEP8, организация кода в модели и пакеты.
7. Коллекции в языках программирования: характеристики и особенности, действия с коллекциями: сортировка и обратный порядок, присваивание, копирование, глубокое копирование. Преобразования из одной коллекции в другую. Обзор стандартных коллекций языка программирования Python.
8. Стандартные коллекции в языке программирования Python: списки. Особенности, методы списков и операции над ними, срезы, сортировка и обратный порядок в списках, копирование списков.
9. Стандартные коллекции в языке программирования Python: кортежи. Особенности, методы кортежей и операции над ними, сортировка и обратный порядок в кортежах, копирование кортежей.
10. Стандартные коллекции в языке программирования Python: множества. Особенности, методы множеств и операции над ними, сортировка и обратный порядок во множествах, копирование множеств.
11. Стандартные коллекции в языке программирования Python: замороженные множества. Особенности, методы замороженных множеств и операции над ними, сортировка и обратный порядок в замороженных множествах, копирование замороженных множеств.
12. Стандартные коллекции в языке программирования Python: словари. Особенности, методы словарей и операции над ними, сортировка и обратный порядок в словарях, копирование словарей.
13. Коллекции в языке программирования Python: модуль collections.
14. Специализированные структуры данных: стек, очередь, дек, разреженный массив. Их реализация средствами языка программирования Python.
15. Функции в языке программирования Python. Область видимости переменных. Именованные параметры функций. Значения параметров по умолчанию. Функции с переменным числом аргументов.
16. Функциональное программирование в Python: итераторы и итерируемые объекты, функции-генераторы, генераторы, оператор yield.
17. Функциональное программирование в Python: списковые включения, встроенные функции высших порядков, лямбда-функции.
18. Функциональное программирование в Python: функции как объект, декораторы функций, замыкания, функторы.

19. Особенности объектно-ориентированного подхода в Python: реализация инкапсуляции, наследования и полиморфизма средствами языка Python.

20. Исключения в Python. Генерирование и перехват исключений. Создание собственных классов исключений.

21. Работа с протоколом HTTP и API средствами языка программирования Python (библиотека requests).

22. Работа с файлами в Python: основные операции с файлами, режимы доступа, менеджер контекста. Создание собственных классов менеджеров контекста.

23. Работа с файловой системой: модули os, shutil.

24. Работа с различными форматами файлов: csv-формат.

25. Работа с различными форматами файлов: json-формат.

26. Работа с различными форматами файлов: xml-формат.

27. Работа с различными форматами файлов: zip-архивы.

28. Общие принципы тестирования программного обеспечения. Тестирование кода на Python: библиотеки unittest, pytest.

29. Документирование кода в Python. Библиотека doctest.

## **Раздел 2. Обеспечение качества программного обеспечения**

30. Основные принципы обеспечения качества программного обеспечения. Стандарты ISO 9000 в области качества программного обеспечения.

31. Тестирование и отладка программного обеспечения.

32. Разработка через тестирование.

33. Практики качественной и безопасной разработки ПО: логирование.

34. Практики качественной и безопасной разработки ПО: конфигурационные файлы, управление секретами

35. Практики качественной и безопасной разработки ПО: code review, рефакторинг кода.

36. Практики качественной и безопасной разработки ПО: управление зависимостями (виртуальное окружение, фиксация версий библиотек, аудит зависимостей).

37. Практики качественной и безопасной разработки ПО: статический анализ кода, динамический анализ безопасности.

## **Раздел 3. Введение в технологию разработки программного обеспечения**

38. Структурное программирование. Декомпозиция задачи и проектирование «сверху-вниз».

39. Модульное программирование: принципы, сферы применения. Связность и сцепление модулей.

40. Объектно-ориентированное программирование: основные принципы, сфера применения.

41. Асинхронность в программировании.

42. Методы параллельного выполнения программ.

43. Типы архитектур программного обеспечения: монолитная архитектура, сервис-ориентированная архитектура, микросервисная архитектура. Привести примеры.

44. Типы архитектур программного обеспечения: архитектура клиент-сервер, двухзвенная архитектура, трехзвенная архитектура. Привести примеры.

45. Архитектура распределенных систем, архитектура облачных систем.

46. API как средство интеграции приложений.

47. Технологии OLE, DDE, COM, DCOM, CORBA: принцип работы, область использования, примеры.

48. Жизненный цикл разработки программного обеспечения. Стандарты ISO 9000 в области разработки программного обеспечения.

49. Модели жизненного цикла разработки программного обеспечения.

50. Коллективная разработка программного обеспечения.

51. Анализ требований к программному обеспечению.
52. Разработка технического задания.
53. Проектирование программного обеспечения: проектирование архитектуры, устройства компонент, пользовательского интерфейса.
54. Документирование программного обеспечения. Средства автоматизации документирования программного обеспечения.

#### **Раздел 4. Архитектура и средства разработки web-приложений**

55. Основные виды веб приложений и их архитектура.
56. Смена подходов к разработке веб-приложений. Концепция Web 2.0.
57. Понятие веб-сервиса. Построение веб-сервисов с использованием протокола SOAP и архитектуры REST.
58. Фреймворки для веб-разработки. Привести примеры. Модель MVC. Привести примеры.
59. Протокол HTTP: назначение, принцип работы, виды HTTP-запросов.
60. Основные инструменты для разработки веб-приложений: JavaScript, библиотека jQuery.
61. Браузерное окружение, DOM, браузерные события.
62. Подход AJAX при разработке веб-приложений.

#### **Раздел 5. Разработка веб-приложений с использованием фреймворка FastAPI**

63. Назначение и особенности фреймворка FastAPI. Стандарты WSGI и ASGI. Отличие фреймворка FastAPI от фреймворков Flask Django. Чем обеспечивается высокая производительность FastAPI?
64. Типовая архитектура проекта FastAPI, в котором используется разделение на модули. Какие компоненты обычно выносятся в отдельные директории (какую роль они выполняют)?
65. Маршрутизация и запросы в фреймворке FastAPI: базовый эндпоинт, обработка параметров пути и параметры запроса, передача данных в теле запроса, обработка загрузки файлов, отправка куки и заголовков. В чем разница между синхронными и асинхронными функциями-обработчиками в FastAPI?
66. Модель обработки запроса в цепочке middleware в проектах FastAPI. Как порядок добавления middleware влияет на обработку запросов?
67. Способы передачи конфигурации (переменных окружения) в приложение FastAPI. Как организовать централизованное управление настройками?
68. Валидация данных в фреймворке FastAPI: валидация на уровне поля и валидация на уровне модели, Pydantic-модели, обработка ошибок валидации. Настройка ответа от API для скрытия чувствительных данных. В чем разница между response\_model и прямой сериализацией данных в словарь? Какие преимущества даёт использование response\_model с точки зрения документации и безопасности?
69. Интеграция баз данных в фреймворке FastAPI: понятие ORM, подключение ORM к FastAPI, реализация синхронных и асинхронных запросов к базе данных, сравнение подходов, использование Pydantic-моделей вместе с моделями базы данных. Применение миграции при изменении моделей? Как управлять сессиями БД с помощью механизма зависимостей, чтобы избежать утечек соединений?
70. Обработка ошибок, тестирование приложений FastAPI: принципы централизованной обработки исключений в FastAPI, обработка исключений на уровне приложения, обработка исключений внутри эндпоинта.
71. Автоматическая генерация документации в формате OpenAPI в проектах FastAPI: Интерактивные интерфейсы Swagger UI / ReDoc. Какие декораторы или аннотации позволяют дополнить эту документацию описаниями эндпоинтов?
72. Принципы безопасной разработки в проектах FastAPI: внедрение зависимостей с точки зрения архитектуры приложения, использование зависимостей для

переиспользования логики. Влияние внедрение зависимостей на повышение тестируемости, переиспользуемости и разделения ответственности между слоями проекта FastAPI.

73. Принципы безопасной разработки в проектах FastAPI: аутентификация и авторизация, модель безопасности OAuth2 с использованием JWT, преимущества подхода, вариант реализации через зависимости.

Принципы безопасной разработки в проектах FastAPI. Какие уязвимости могут возникнуть при неправильной настройке CORS в FastAPI? Как правильно сконфигурировать CORS middleware для продакшен-окружения?

74. Принципы безопасной разработки в проектах FastAPI. Какие уязвимости, характерные для веб-приложений (например, CSRF, XSS, инъекции), не решаются автоматически FastAPI, и какие практики безопасной разработки должны применяться разработчиком дополнительно?

75. Тестирование проектов FastAPI. Использование mock для внешних сервисов.

76. Развертывание приложений FastAPI. Какие ASGI-серверы можно использовать для запуска FastAPI в продакшене? Разница между однопроцессным запуском, использованием нескольких рабочих процессов и многопоточностью с точки зрения масштабирования и стабильности.

77. В чем особенности развертывания FastAPI в бессерверной среде? Какие ограничения накладывают облачные платформы на использование асинхронности, длительности запросов и WebSocket?

78. Какие аспекты работы FastAPI-приложения необходимо учитывать при упаковке в Docker? Как выбор базового образа, управление переменными окружения и обработка сигналов завершения влияют на надежность контейнеризованного решения?

Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	<p>ОПК-7: Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ</p> <ul style="list-style-type: none"> <li>– ОПК-7.1: Реализует базовые алгоритмы средствами языков программирования.</li> <li>– ОПК-7.2: Проектирует программное обеспечение с использованием средств автоматизации</li> <li>– ОПК-7.3: Проводит комплексное тестирование и отладку программных систем</li> </ul>	
<p>ОПК-7.1: Реализует базовые алгоритмы средствами языков программирования</p>	<p><b>Теоретические вопросы к экзамену</b> Раздел 1. Методы программирования на Python</p> <ol style="list-style-type: none"> <li>1. Основные парадигмы программирования.</li> <li>2. Эволюция языков программирования.</li> <li>3. Компиляторы и интерпретаторы. Виртуальные машины для исполнения байт-кода. JIT-компиляторы.</li> <li>4. Инструментальные средства разработки программного обеспечения: назначение, разновидности, краткая характеристика, примеры.</li> <li>5. Интегрированная среда разработки: область применения, состав, функции, примеры.</li> <li>6. Особенности языка программирования Python. Соглашения по оформлению кода PEP8, организация кода в модели и пакеты.</li> <li>7. Коллекции в языках программирования: характеристики и особенности, действия с коллекциями: сортировка и обратный порядок, присваивание, копирование, глубокое копирование. Преобразования из одной коллекции в другую. Обзор стандартных коллекций языка программирования Python.</li> <li>8. Стандартные коллекции в языке программирования Python: списки. Особенности, методы списков и операции над ними, срезы, сортировка и обратный порядок в списках, копирование списков.</li> <li>9. Стандартные коллекции в языке программирования Python: кортежи. Особенности, методы кортежей и операции над ними, сортировка и обратный порядок в кортежах, копирование кортежей.</li> <li>10. Стандартные коллекции в языке программирования Python: множества. Особенности, методы множеств и операции над ними, сортировка и обратный порядок во множествах, копирование множеств.</li> <li>11. Стандартные коллекции в языке программирования Python: замороженные множества. Особенности, методы замороженных множеств и операции над ними, сортировка и обратный порядок в замороженных множествах, копирование замороженных множеств.</li> <li>12. Стандартные коллекции в языке программирования Python: словари. Особенности, методы словарей и операции над ними, сортировка и обратный порядок в словарях, копирование словарей.</li> <li>13. Коллекции в языке программирования Python: модуль collections.</li> <li>14. Специализированные структуры данных: стек, очередь, дек, разреженный массив. Их реализация средствами языка программирования Python.</li> <li>15. Функции в языке программирования Python. Область видимости переменных. Именованные параметры функций. Значения параметров по умолчанию. Функции с переменным числом аргументов.</li> <li>16. Функциональное программирование в Python: итераторы и итерируемые объекты, функции-генераторы, генераторы, оператор yield.</li> <li>17. Функциональное программирование в Python: списковые включения, встроенные функции высших порядков, лямбда-функции.</li> </ol>	

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	<p>18. Функциональное программирование в Python: функции как объект, декораторы функций, замыкания, функторы.</p> <p>19. Особенности объектно-ориентированного подхода в Python: реализация инкапсуляции, наследования и полиморфизма средствами языка Python.</p> <p>20. Исключения в Python. Генерирование и перехват исключений. Создание собственных классов исключений.</p> <p>21. Работа с протоколом HTTP и API средствами языка программирования Python (библиотека requests).</p> <p>22. Работа с файлами в Python: основные операции с файлами, режимы доступа, менеджер контекста. Создание собственных классов менеджеров контекста.</p> <p>23. Работа с файловой системой: модули os, shutil.</p> <p>24. Работа с различными форматами файлов: csv-формат.</p> <p>25. Работа с различными форматами файлов: json-формат.</p> <p>26. Работа с различными форматами файлов: xml-формат.</p> <p>27. Работа с различными форматами файлов: zip-архивы.</p> <p>28. Общие принципы тестирования программного обеспечения. Тестирование кода на Python: библиотеки unittest, pytest.</p> <p>29. Документирование кода в Python. Библиотека doctest.</p> <p>Раздел 5. Разработка веб-приложений с использованием фреймворка FastAPI</p> <p>1. Назначение и особенности фреймворка FastAPI. Стандарты WSGI и ASGI. Отличие фреймворка FastAPI от фреймворков Flask Django. Чем обеспечивается высокая производительность FastAPI?</p> <p>2. Типовая архитектура проекта FastAPI, в котором используется разделение на модули. Какие компоненты обычно выносятся в отдельные директории (какую роль они выполняют)?</p> <p>3. Маршрутизация и запросы в фреймворке FastAPI: базовый эндпоинт, обработка параметров пути и параметры запроса, передача данных в теле запроса, обработка загрузки файлов, отправка куки и заголовков. В чем разница между синхронными и асинхронными функциями-обработчиками в FastAPI?</p> <p>4. Модель обработки запроса в цепочке middleware в проектах FastAPI. Как порядок добавления middleware влияет на обработку запросов?</p> <p>5. Способы передачи конфигурации (переменных окружения) в приложение FastAPI. Как организовать централизованное управление настройками?</p> <p>6. Валидация данных в фреймворке FastAPI: валидация на уровне поля и валидация на уровне модели, Pydantic-модели, обработка ошибок валидации. Настройка ответа от API для скрытия чувствительных данных. В чем разница между response_model и прямой сериализацией данных в словарь? Какие преимущества даёт использование response_model с точки зрения документации и безопасности?</p> <p>7. Интеграция баз данных в фреймворке FastAPI: понятие ORM, подключение ORM к FastAPI, реализация синхронных и асинхронных запросов к базе данных, сравнение подходов, использование Pydantic-моделей вместе с моделями базы данных. Применение миграции при изменении моделей? Как управлять сессиями БД с помощью механизма зависимостей, чтобы избежать утечек соединений?</p> <p>8. Обработка ошибок, тестирование приложений FastAPI: принципы централизованной обработки исключений в FastAPI, обработка исключений на уровне приложения, обработка исключений внутри эндпоинта.</p> <p>9. Автоматическая генерация документации в формате OpenAPI в проектах FastAPI: Интерактивные интерфейсы Swagger UI / ReDoc. Какие декораторы или аннотации позволяют дополнить эту документацию описаниями эндпоинтов?</p> <p>10. Принципы безопасной разработки в проектах FastAPI: внедрение зависимостей с точки зрения архитектуры приложения, использование зависимостей для переиспользования логики. Влияние внедрение зависимостей на повышение тестируемости, переиспользуемости и разделения ответственности между слоями проекта FastAPI.</p>	

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	<p>11. Принципы безопасной разработки в проектах FastAPI: аутентификация и авторизация, модель безопасности OAuth2 с использованием JWT, преимущества подхода, вариант реализации через зависимости.</p> <p>Принципы безопасной разработки в проектах FastAPI. Какие уязвимости могут возникнуть при неправильной настройке CORS в FastAPI? Как правильно сконфигурировать CORS middleware для продакшен-окружения?</p> <p>12. Принципы безопасной разработки в проектах FastAPI. Какие уязвимости, характерные для веб-приложений (например, CSRF, XSS, инъекции), не решаются автоматически FastAPI, и какие практики безопасной разработки должны применяться разработчиком дополнительно?</p> <p>13. Тестирование проектов FastAPI. Использование mock для внешних сервисов.</p> <p>14. Развертывание приложений FastAPI. Какие ASGI-серверы можно использовать для запуска FastAPI в продакшене? Разница между однопроцессным запуском, использованием нескольких рабочих процессов и многопоточностью с точки зрения масштабирования и стабильности.</p> <p>15. В чем особенности развертывания FastAPI в бессерверной среде? Какие ограничения накладывают облачные платформы на использование асинхронности, длительности запросов и WebSocket?</p> <p>16. Какие аспекты работы FastAPI-приложения необходимо учитывать при упаковке в Docker? Как выбор базового образа, управление переменными окружения и обработка сигналов завершения влияют на надежность контейнеризированного решения?</p> <p><b>Примеры практических заданий для экзамена:</b></p> <p>1. Реализуйте сложение и скалярное умножение векторов на языке Python, используя функции высшего порядка zip, map, reduce и др. Если длина векторов не совпадает, должно генерироваться исключение.</p> <p>2. Используя язык Python, реализуйте генератор factorials(n), генерирующий последовательность факториалов натуральных чисел.</p> <p>3. На языке программирования Python объявите класс Автомобиль, содержащий не менее 4-х полей и 2-х методов. Создайте список из объектов класса Автомобиль.</p> <p>5. Используя механизм шаблонов фреймворка Flask, создайте веб-сайт для научного общества. На главной странице разместите небольшой текст и ссылку на страницу со статьями. Для генерации страницы со списком статей используйте циклы в шаблонах, добавьте ссылку на главную страницу. Для оформления предусмотрите простейшие стили.</p>	
<p>ОПК-7.2: Проектирует программное обеспечение с использованием средств автоматизации</p>	<p><b>Теоретические вопросы к экзамену:</b></p> <p>Раздел 3. Введение в технологию разработки программного обеспечения</p> <p>1. Структурное программирование. Декомпозиция задачи и проектирование «сверху-вниз».</p> <p>2. Модульное программирование: принципы, сферы применения. Связность и сцепление модулей.</p> <p>3. Объектно-ориентированное программирование: основные принципы, сфера применения.</p> <p>4. Асинхронность в программировании.</p> <p>5. Методы параллельного выполнения программ.</p> <p>6. Типы архитектур программного обеспечения: монолитная архитектура, сервис-ориентированная архитектура, микросервисная архитектура. Привести примеры.</p> <p>7. Типы архитектур программного обеспечения: архитектура клиент-сервер, двухзвенная архитектура, трехзвенная архитектура. Привести примеры.</p> <p>8. Архитектура распределенных систем, архитектура облачных систем.</p> <p>9. API как средство интеграции приложений.</p> <p>10. Технологии OLE, DDE, COM, DCOM, CORBA: принцип работы, область использования, примеры.</p> <p>11. Жизненный цикл разработки программного обеспечения. Стандарты ISO 9000 в области разработки программного обеспечения.</p> <p>12. Модели жизненного цикла разработки программного обеспечения.</p> <p>13. Коллективная разработка программного обеспечения.</p>	

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	<p>14. Анализ требований к программному обеспечению.</p> <p>15. Разработка технического задания.</p> <p>16. Проектирование программного обеспечения: проектирование архитектуры, устройства компонент, пользовательского интерфейса.</p> <p>17. Документирование программного обеспечения. Средства автоматизации документирования программного обеспечения.</p> <p>Раздел 4. Архитектура и средства разработки web-приложений</p> <p>18. Основные виды веб приложений и их архитектура.</p> <p>19. Смена подходов к разработке веб-приложений. Концепция Web 2.0.</p> <p>20. Понятие веб-сервиса. Построение веб-сервисов с использованием протокола SOAP и архитектуры REST.</p> <p>21. Фреймворки для веб-разработки. Привести примеры. Модель MVC. Привести примеры.</p> <p>22. Протокол HTTP: назначение, принцип работы, виды HTTP-запросов.</p> <p>23. Основные инструменты для разработки веб-приложений: JavaScript, библиотека jQuery.</p> <p>24. Браузерное окружение, DOM, браузерные события.</p> <p>25. Подход AJAX при разработке веб-приложений.</p>	
<p><b>ОПК-7.3:</b> Проводит комплексное тестирование и отладку программных систем</p>	<p><b>Теоретические вопросы к экзамену:</b></p> <p>Раздел 2. Обеспечение качества программного обеспечения</p> <p>26. Основные принципы обеспечения качества программного обеспечения. Стандарты ISO 9000 в области качества программного обеспечения.</p> <p>27. Тестирование и отладка программного обеспечения.</p> <p>28. Разработка через тестирование.</p> <p>29. Практики качественной и безопасной разработки ПО: логирование.</p> <p>30. Практики качественной и безопасной разработки ПО: конфигурационные файлы, управление секретами</p> <p>31. Практики качественной и безопасной разработки ПО: code review, рефакторинг кода.</p> <p>32. Практики качественной и безопасной разработки ПО: управление зависимостями (виртуальное окружение, фиксация версий библиотек, аудит зависимостей).</p> <p>33. Практики качественной и безопасной разработки ПО: статический анализ кода, динамический анализ безопасности.</p> <p>34. Общие принципы тестирования программного обеспечения. Тестирование кода на Python: библиотеки unittest, pytest.</p> <p><b>Примеры практических заданий для экзамена:</b></p> <p>1. Реализуйте на языке Python функцию time2min(h, m), которая переводит часы и минуты в минуты с начала суток (00:00). Проведите тестирование функции, используя библиотеки unittest или pytest.</p>	

**б) Примерный перечень задания для рубежного контроля и индивидуальных домашних заданий**

1. Создание инструмента обнаружения хостов UDP.
2. Создание сниффера для захвата учетных данных SMTP, POP3, IMAP.
3. Создание тестовых примеров фаззинга на основе захваченного сетевого трафика.
4. Создание инструмента определения поверхности атак Web-приложений. Брутфорс формы аутентификации.
5. Создание Github-Aware Trojan.
6. Создание динамической программы статистики критических уязвимостей в ПО различных производителей.
7. Создание системы анализа сетевого трафика на предмет наличия признаков сетевой атаки
8. Разработка генератора поддельных IP-пакетов для проведения smurf-атак.
9. Разработка плагина для браузера с функцией межсетевого экрана.

10. Разработка плагина для браузера, позволяющего обнаружить XSS-атаки.
11. Разработка приложения для проверки защищенности сайта от SQL-инъекций.
12. Построение топологии сети по ARP-таблицам.
13. Разработка утилиты для групповой обработки файлов на языке программирования Python.
14. Разработка MP3-плеера на языке программирования Python.
15. Разработка утилиты командной строки Site Connectivity Checker для проверки доступности сайта на языке программирования Python.
16. Разработка клиент-серверного приложения для приема численных метрик - характеристик ПК, используя возможности языка программирования Python.
17. Разработка клиент-серверного приложения для выполнения арифметических действия над двоичными или шестнадцатеричными числами, используя возможности языка программирования Python.
18. Разработать клиент-серверную систему для сжатия текстовой информации на русском языке и последующей распаковки.
19. Разработка комплекта тестов для клиент-серверного приложения.
20. Разработка REST-сервиса и веб-клиента для ведения заметок.
21. Разработка REST-сервиса и веб-клиента для поиска в тексте на основе регулярных выражений.
22. Разработка REST-сервиса и клиента для учета и отслеживания ежедневных расходов.
23. Разработка REST-сервиса для аптечного склада n
24. Разработка REST-сервиса для проверки совместимости деталей ПК на языке программирования Python
25. Разработка утилиты для работы с изображениями с использованием библиотек NumPy, SciPy.
26. Сравнение возможностей библиотек для работы с изображениями для языка программирования Python.
27. Разработка агрегатора контента.
28. Разработка виртуального помощника для проведения тестирования в сфере информационной безопасности.
29. Разработка виртуального помощника для подбора вакансий в сфере информационной безопасности (с функцией искусственного интеллекта).
30. Создание виртуального помощника (чат-бота или голосового помощника) для подбора вакансий в сфере информационной безопасности.
31. Разработка виртуального помощника для сети аптек (с функцией искусственного интеллекта).
32. Разработка виртуального помощника для создания целевых страниц (с функцией искусственного интеллекта).

**в) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:**

Промежуточная аттестация по дисциплине включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по данной дисциплине проводится в компьютерном классе по экзаменационным билетам, каждый из которых включает 1 теоретический вопрос и 2 практических задания.

**Показатели и критерии оценивания экзамена:**

– на оценку «отлично» (5 баллов) – обучающийся успешно прошел запланированные рубежные контроли и в ходе промежуточной аттестации демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует

знаниями, умениями, применяет их в ситуациях повышенной сложности;

– на оценку «хорошо» (4 балла) – обучающийся успешно прошел запланированные рубежные контроли и в ходе промежуточной аттестации демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации;

– на оценку «удовлетворительно» (3 балла) – обучающийся прошел запланированные рубежные контроли и в ходе промежуточной аттестации демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации;

– на оценку «неудовлетворительно» (2 балла) – обучающийся прошел запланированные рубежные контроли, но в ходе промежуточной аттестации демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку «неудовлетворительно» (1 балл) – обучающийся не прошел запланированные рубежные контроли, обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.

### **Методические указания по выполнению практических работ**

Практические работы проводятся в компьютерных классах целью получения практических умений для формирования и развития профессиональных навыков и соответствующих компетенций по дисциплине. При подготовке к выполнению заданий практической работы используйте лекции, справочный материал программного обеспечения, рекомендованную литературу и цифровые образовательные ресурсы соответствующих методических материалов, размещенных в сети Интернет или локальной сети университета. Перед выполнением практической работы необходимо получить свой вариант индивидуального задания у преподавателя. Прежде чем приступить к выполнению практической работы, внимательно прочтите рекомендации к ее выполнению. Ознакомьтесь с перечнем рекомендуемой литературы, повторите теоретический материал, относящийся к теме работы. Ответьте на контрольные вопросы, выполните задания для самостоятельного выполнения. По результатам практической работы предоставляется отчет. Отчет к практическим работам должен содержать:

- название практической работы;
- цель и задачи работы;
- краткие теоретические сведения;
- задания по практической работе;
- ход работы - описание последовательности действий при выполнении работы;
- выводы или результаты.

Результаты выполнения практической работы могут быть представлены в электронном варианте или распечатанные. Результаты выполнения заданий практической работы можно сохранить на образовательном портале в личном кабинете и использовать при подготовке к экзамену.

### **Защита работы и результаты оценивания**

Защита проводится в два этапа:

1. Демонстрируются результаты выполнения задания. В случае выполнения практической работы, предусматривающей разработку программы, при помощи тестового примера доказывается, что результат, получаемый при выполнении программы, является правильным.

2. Для защиты работы студенту необходимо ответить на дополнительные вопросы преподавателя. Каждая практическая работа оценивается определенным количеством баллов исходя из 5-бальной системы оценок.

Практическая работа считается выполненной и защищенной, если выполнены все задания и даны правильные ответы преподавателю на заданные вопросы. Практическая работа считается выполненной и незащищенной, если выполнены все задания, но полученные результаты являются неверными или не даны правильные ответы преподавателю на заданные вопросы и ответы были не полные. Обучающемуся, не выполнившему в полном объеме все задания практической работы, или пропустившему по уважительной причине практическую работу, необходимо выполнить ее самостоятельно в компьютерном классе, результаты выполненной работы сохранить на съемном накопителе или на образовательном портале. Результаты предоставить в сроки, указанные преподавателем вместе с отчетом, демонстрацией полученных результатов в компьютерном классе или предоставлением материалов на электронном образовательном ресурсе.

## **Правила по технике безопасности для обучающихся при проведении практических работ**

1. Практические работы проводятся под наблюдением преподавателя. К выполнению практических работ студенты допускаются только после прослушивания инструктажа по технике безопасности и противопожарным мерам.

2. Обучающийся должен строго выполнять правила техники безопасности и санитарно-гигиенические нормы при работе в компьютерных классах или специализированных лабораториях университета.

## Методические указания по выполнению внеаудиторных самостоятельных работ

### Общие положения

Настоящие методические указания предназначены для организации внеаудиторной самостоятельной работы студентов и оказания помощи в самостоятельном изучении теоретического и реализации компетенций обучаемых.

Данные методические указания не являются учебным пособием, поэтому перед началом выполнения самостоятельного задания следует изучить соответствующие разделы лекционных занятий, материалов образовательного портала, разделов основной и дополнительной литературы, представленных в пункте 8. «Учебно-методическое и информационное обеспечение дисциплины (модуля)» данной РПД.

### Цели и задачи самостоятельной работы

Цель самостоятельной работы – содействие оптимальному усвоению учебного материала обучающимися, развитие их познавательной активности, готовности и потребности в самообразовании.

Задачи самостоятельной работы:

- повышение исходного уровня владения информационными технологиями;
- углубление и систематизация знаний;
- постановка и решение стандартных задач профессиональной деятельности;
- развитие работы с различной по объему и виду информацией, учебной и научной литературой;
- практическое применение знаний, умений;
- самостоятельно использование стандартных программных средств сбора, обработки, хранения и защиты информации
- развитие навыков организации самостоятельного учебного труда и контроля за его эффективностью.

Виды внеаудиторной самостоятельной работы и формы контроля и время на выполнение каждого вида самостоятельной работы указаны в пункте 4. «Структура и содержание дисциплины» данной РПД.

### Порядок выполнения

При выполнении текущей внеаудиторной самостоятельной работы обучающемуся следует придерживаться следующего порядка действий:

- 1) внимательно изучить соответствующие теоретические разделы дисциплины, пользуясь материалами (лекционными, презентационными, аудио-визуальными):
  - предоставляемыми преподавателем на лекционных занятиях;
  - предоставляемыми преподавателем в рамках электронных образовательных курсов;
  - содержащимися в учебниках и учебных пособиях ЭБС (электронно-библиотечных систем), электронных каталогов университета и интернет-ресурсов.
- 2) подробно разобрать типовые примеры решения задач, рассмотренные в рамках аудиторной контактной работы с преподавателем;
- 3) применить полученные теоретические знания и практические навыки к решению индивидуальных заданий, к прохождению компьютерных тестирований;
- 4) при необходимости, сформировать перечень вопросов, вызвавших затруднения в процессе самостоятельной работы. Обсудить возникшие вопросы со студентами группы, в рамках командно-проектной работы, и с преподавателем, в рамках консультационной помощи, реализованной либо в контактной форме, либо средствами информационно-образовательной среды ВУЗа.

### Критерии оценки внеаудиторных самостоятельных работ

Качество выполнения внеаудиторной самостоятельной работы обучающихся оценивается посредством текущего контроля самостоятельной работы обучающихся с использованием балльно-рейтинговой системы.

В качестве форм текущего контроля по дисциплине используются: индивидуальные задания, аудиторские контрольные работы, компьютерное тестирование.

Максимальное количество баллов обучающийся получает, если:

- выполняет индивидуальные задания в соответствии со всеми заявленными требованиями;

- дает правильные формулировки, точные определения, понятия терминов;

- может обосновать рациональность решения текущей задачи.;

- обстоятельно с достаточной полнотой излагает соответствующую теоретический раздел;

- правильно отвечает на дополнительные вопросы преподавателя, имеющие целью выяснить степень понимания им данного материала.

50~85% от максимального количества баллов обучающийся получает, если:

- неполно (не менее 70% от полного), но правильно выполнено задание;

- при изложении были допущены 1-2 несущественные ошибки, которые он исправляет после замечания преподавателя;

- дает правильные формулировки, точные определения, понятия терминов;

- может обосновать свой ответ, привести необходимые примеры;

- правильно отвечает на дополнительные вопросы преподавателя, имеющие целью выяснить степень понимания им данного материала.

36~50% от максимального количества баллов обучающийся получает, если:

- неполно (не менее 50% от полного), но правильно изложено задание;

- при изложении была допущена 1 существенная ошибка;

- знает и понимает основные положения данной темы, но допускает неточности в формулировке понятий;

- излагает выполнение задания недостаточно логично и последовательно;

- затрудняется при ответах на вопросы преподавателя.

35% и менее от максимального количества баллов обучающийся получает, если:

- неполно (менее 50% от полного) изложено задание;

- при изложении были допущены существенные ошибки. В "0" баллов преподаватель вправе оценить выполненное обучающимся задание, если оно не удовлетворяет требованиям, установленным преподавателем к данному виду работы или не было представлено для проверки.

Сумма полученных баллов по всем видам заданий внеаудиторной самостоятельной работы составляет рейтинговый показатель обучающегося. Рейтинговый показатель обучающегося влияет на выставление итоговой оценки по результатам изучения дисциплины.

Показатели и критерии оценивания полученных знаний представлены в приложении 2 данной РПД.