



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

26.01.2022 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

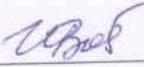
Форма обучения  
очная

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

Магнитогорск  
2022 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  
25.01.2022, протокол № 4

Зав. кафедрой  И.И. Баранкова

Рабочая программа одобрена методической комиссией ИЭиАС  
26.01.2022 г. протокол № 5

Председатель  В.Р. Храмшин

Рабочая программа составлена:  
ст. преподаватель кафедры ИиИБ,

 Ю.А. Мазнина

Рецензент:

Начальник УИТ и АСУ, канд. техн. наук

 К.А. Рубан

## Лист актуализации рабочей программы

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 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 2024 - 2025 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 2025 - 2026 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 2026 - 2027 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 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

### **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, освоении методологий структурного и объектно-ориентированного 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:

Тестирование систем защиты информации автоматизированных систем

Технология построения защищенных распределенных приложений

Разработка и эксплуатация автоматизированных систем в защищенном исполнении

Управление информационной безопасностью

### **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 акад. часов, в том числе:

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

Форма аттестации - экзамен

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. 1. Введение в технологию разработки программного обеспечения								
1.1 Языки и парадигмы программирования. Инструментальные средства для разработки программного обеспечения. Компиляторы и интерпретаторы	4	0,5		0,5/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию	Тестирование	ОПК-7.2
1.2 Типы архитектур программного обеспечения: монолитная архитектура, клиент-серверная архитектура, сервис-ориентированная архитектура, микросервисная архитектура программного обеспечения, облачная архитектура		0,5		0,5/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию	Тестирование	ОПК-7.2
1.3 Жизненный цикл разработки программного обеспечения. Модели жизненного цикла. Коллективная разработка ПО		0,5		1/1И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию.	Тестирование	ОПК-7.2

1.4 Анализ требований. Разработка технического задания. Проектирование программного обеспечения: проектирование архитектуры, устройства компонент, пользовательского интерфейса		0,5		2/1,5И	2,15	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.2
Итого по разделу		2		4/3,5И	5,15			
2. 2. Обеспечение качества программного обеспечения								
2.1 Основные принципы обеспечения качества программного обеспечения. Стандарты ISO 9000	4	0,5		1/0,5И	1	Самостоятельное изучение учебной и научно 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. Документирование программного обеспечения		0,5		1/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.2, ОПК-7.3
Итого по разделу		1		2/1И	2			
3. 3. Язык программирования Python								
3.1 Особенности языка программирования Python: интерпретируемый язык, динамическая типизация, концепция полностью объектно ориентированного языка, стиль написания кода, соглашения по оформлению кода pep8, организация кода в модели и пакеты	4	0,5		1/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию	Тестирование	ОПК-7.1, ОПК-7.2, ОПК-7.3
3.2 Язык программирования Python: числовые, булевы и строковые литералы, основные операции над простыми типами, синтаксические конструкции, условия, циклы, функция range(). Системные библиотеки Python: math, time, random		0,5		2/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к АКР	Аудиторная контрольная работа	ОПК-7.1, ОПК-7.2, ОПК-7.3

3.3 Коллекции в языке программирования Python: строки.	0,5		1/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к АКР	Аудиторная контрольная работа	ОПК-7.1, ОПК-7.2, ОПК-7.3
3.4 Коллекции в языке программирования Python: списки, кортежи, множества. Методы списков и операции со списками. Срезы списков. Присваивание в срез. Копирование списка.	0,5		1/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к АКР	Аудиторная контрольная работа	ОПК-7.1, ОПК-7.2, ОПК-7.3
3.5 Коллекции в языке программирования Python: словари	0,5		2/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к АКР	Аудиторная контрольная работа	ОПК-7.1, ОПК-7.2, ОПК-7.3
3.6 Функции в языке программирования Python. Область видимости переменных. Именованные параметры функций. Значения параметров по умолчанию. Функции с переменным числом аргументов, лямбда-функции	0,5		2/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
3.7 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	0,5		2/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3

3.8 Особенности объектно-ориентированного подхода в Python: реализация инкапсуляции, наследования и полиморфизма средствами языка Python		0,5		2/0,5И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
3.9 Работа с файловой системой и популярными форматами файлов (csv- и json-файлами, zip-архивами)		0,5		2/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
3.10 Исключения в Python. Генерирование и перехват исключений		0,5		2/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
Итого по разделу		5		17/5И	11			
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		1/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию	Тестирование	ОПК-7.2
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)		1		2/0,5И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к АКР	Аудиторная контрольная работа	ОПК-7.1, ОПК-7.2, ОПК-7.3

4.3 Основные инструменты для разработки веб-приложений: HTML, CSS, JavaScript, библиотека jQuery. Подход AJAX		3		4/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
4.4 Фреймворки для веб-разработки. Модель MVC		0,5		1/0,5И	1	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию	Тестирование	ОПК-7.2
Итого по разделу		5		8/2И	5			
5. Разработка веб-приложений с использованием микрофреймворка Flask								
5.1 Микрофреймворк Flask: особенности, принципы работы, основные этапы создания веб-приложения		0,5		2/0,5И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию	Тестирование	ОПК-7.1, ОПК-7.2, ОПК-7.3
5.2 Фреймворк Flask: HTML-шаблоны (templates), подключение и использование static-файлов (JS, CSS), работа с формами	4	1		4/1,5И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
5.3 Микрофреймворк Flask: технология объектно-реляционного отображения, работа с БД и моделями (models)		0,5		4/1И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3

5.4 Микрофреймворк Flask: механизмы сессий, cookie, авторизация и права пользователей	0,5		4/1,85И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
5.5 Микрофреймворк Flask: проектирование RESTful API	0,5		2/0,5И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
5.6 Микрофреймворк Flask: тестирование и развертывание веб-приложения	0,5		2/0,5И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
5.7 Микрофреймворк Flask: безопасность веб-приложения	0,5		2/0,5И	2	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ОПК-7.1, ОПК-7.2, ОПК-7.3
Итого по разделу	4		20/6,35И	14			
Итого за семестр	17		51/17,85И	37,15		экзамен	
Итого по дисциплине	17		51/17,85И	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 (модуля)**

### **а) Основная литература:**

1. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2021. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/469759> (дата обращения: 01.06.2021).

2. 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, 2021. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/470923> (дата обращения: 01.06.2021).

3. Маркин, А. В. Программирование на SQL в 2 ч. Часть 1 : учебник и практикум для вузов / А. В. Маркин. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 403 с. — (Высшее образование). — ISBN 978-5-534-12256-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/452357> (дата обращения: 01.06.2021).

3. Маркин, А. В. Программирование на SQL в 2 ч. Часть 2 : учебник и практикум для вузов / А. В. Маркин. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2021. — 340 с. — (Высшее образование). — ISBN 978-5-534-12258-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/469959> (дата обращения: 01.06.2021).

4. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2021. — 161 с. — (Высшее образование). — ISBN 978-5-534-10971-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/472985> (дата обращения: 01.06.2021).

5. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2021. — 147 с. — (Высшее образование). — ISBN 978-5-534-09172-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/471564> (дата обращения: 01.06.2021).

6. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / В. М. Шелудько ;

Южный федеральный университет. - Ростов-наДону ; Таганрог : Издательство Южного федерального университета, 2017. - 107 с. - ISBN 978-5-9275-2648-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1021664> (дата обращения: 01.06.2021).

**б) Дополнительная литература:**

1. Гагарина, Л. Г. Технология разработки программного обеспечения : учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0707-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1011120> (дата обращения: 01.06.2021).

2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-487-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1206074> (дата обращения: 01.06.2021).

3. Жуков, Р. А. Язык программирования Python: практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook\_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1412168> (дата обращения: 01.06.2021).

4. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2021. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/470281> (дата обращения: 01.06.2021).

5. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2021. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/470387> (дата обращения: 01.06.2021).

6. Нестеров, С. А. Базы данных : учебник и практикум для вузов / С. А. Нестеров. — Москва : Издательство Юрайт, 2021. — 230 с. — (Высшее образование). — ISBN 978-5-534-00874-6. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/469516> (дата обращения: 01.06.2021).

7. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2021. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/477353> (дата обращения: 01.06.2021)

**в) Методические указания:**

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

**Программное обеспечение**

Наименование ПО	№ договора	Срок действия лицензии
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
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

#### **Профессиональные базы данных и информационные справочные системы**

Название курса	Ссылка
Информационная система - Единое окно доступа к информационным ресурсам	URL: <a href="http://window.edu.ru/">http://window.edu.ru/</a>
Поисковая система Академия Google (Google Scholar)	URL: <a href="https://scholar.google.ru/">https://scholar.google.ru/</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>

#### **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.

Лекционные аудитории (ауд. 2124, ауд. 226, ауд. 238, ауд. 365, ауд. 388, ауд. 433 и т.д.)

Компьютерные классы (ауд. 245, 247, 249, 279, 372, 394 и т.д.)

Аудитории для самостоятельной работы (ауд. 132а): компьютерные классы, читальные залы библиотеки.

### 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 так же с применением кейс-технологий.

#### 1. Контрольные вопросы и задания для проведения текущего контроля

##### Раздел 1. Введение в технологию разработки программного обеспечения

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. Структурное программирование. Декомпозиция задачи и проектирование «сверху-вниз».
7. Модульное программирование: принципы, сферы применения. Связность и сцепление модулей.
8. Объектно-ориентированное программирование: основные принципы, сфера применения.
9. Типы архитектур программного обеспечения: монолитная архитектура, сервис-ориентированная архитектура, микросервисная архитектура. Привести примеры.
10. Типы архитектур программного обеспечения: архитектура клиент-сервер, двухзвенная архитектура, трехзвенная архитектура. Привести примеры.
11. Архитектура распределенных систем, архитектура облачных систем.
12. API как средство интеграции приложений.
13. Технологии OLE, DDE, COM, DCOM, CORBA: принцип работы, область использования, примеры.
14. Жизненный цикл разработки программного обеспечения. Стандарты ISO 9000 в области разработки программного обеспечения.
15. Модели жизненного цикла разработки программного обеспечения.
16. Коллективная разработка ПО.
17. Анализ требований к программному обеспечению.
18. Разработка технического задания.
19. Проектирование программного обеспечения: проектирование архитектуры, устройства компонент, пользовательского интерфейса.

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

1. Основные принципы обеспечения качества программного обеспечения. Стандарты ISO 9000 в области качества программного обеспечения.
2. Тестирование и отладка программного обеспечения.
3. Документирование программного обеспечения. Средства автоматизации документирования программного обеспечения.

### **Раздел 3. Язык программирования Python**

1. Особенности языка программирования Python. Соглашения по оформлению кода `pep8`, организация кода в модели и пакеты.
2. Язык программирования Python: числовые, булевы и строковые литералы, основные операции над простыми типами.
3. Язык программирования Python: синтаксические конструкции, условия, циклы, функция `range()`.
4. Язык программирования Python: системные библиотеки Python: `math`, `time`, `random`, `os`.
5. Коллекции в языке программирования Python: строки
6. Коллекции в языке программирования Python: списки, кортежи, множества. Методы списков и операции со списками. Срезы списков. Присваивание в срез. Копирование списка.
7. Коллекции в языке программирования Python: словари
8. Функции в языке программирования Python. Область видимости переменных. Именованные параметры функций. Значения параметров по умолчанию. Функции с переменным числом аргументов.
9. Функциональное программирование в Python: списковые включения, встроенные функции высших порядков, лямбда-функции.
10. Функциональное программирование в Python: функции как объект, декораторы функций, замыкания, функторы.
11. Функциональное программирование в Python: итераторы и итерируемые объекты, функции-генераторы, оператор `yield`.
12. Особенности объектно-ориентированного подхода в Python: реализация инкапсуляции, наследования и полиморфизма средствами языка Python.
13. Исключения в Python. Генерирование и перехват исключений.
14. Библиотека `tkinter`: назначение, сфера применения, обзор возможностей, менеджеры геометрии.
15. Библиотека `pymru`: назначение, сфера применения, обзор возможностей.
16. Библиотека `matplotlib`: назначение, сфера применения, обзор возможностей.
17. Работа с протоколом HTTP и API средствами языка программирования Python (библиотека `requests`)
18. Работа с файловой системой и популярными форматами файлов: `csv`-формат.
19. Работа с файловой системой и популярными форматами файлов: `json`-формат.
20. Работа с файловой системой и популярными форматами файлов: `xml`-формат.
21. Работа с файловой системой и популярными форматами файлов: `zip`-архивы.
22. Общие принципы тестирования программного обеспечения. Тестирование кода на Python: библиотеки `unittest`, `pytest`.
23. Документирование кода в Python. Библиотека `doctest`.

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

1. Основные виды веб приложений и их архитектура.
2. Смена подходов к разработке веб-приложений. Концепция Web 2.0.
3. Понятие веб-сервиса. Построение веб-сервисов с использованием протокола SOAP и архитектуры REST.
4. Фреймворки для веб-разработки. Привести примеры. Модель MVC. Привести примеры.
5. Протокол HTTP: назначение, принцип работы, виды HTTP-запросов.
6. Основные инструменты для разработки веб-приложений: JavaScript, библиотека `jQuery`.
7. Браузерное окружение, DOM, браузерные события.

8. Подход AJAX при разработке веб-приложений.

## Раздел 5. Разработка веб-приложения с использованием микрофреймворка Flask

1. Микрофреймворк Flask: особенности, принципы работы, основные этапы создания веб-приложения
2. Фреймворк Flask: HTML-шаблоны (templates), подключение и использование static-файлов (JS, CSS), работа с формами
3. Микрофреймворк Flask: технология объектно-реляционного отображения, работа с БД и моделями (models)
4. Микрофреймворк Flask: механизмы сессий, cookie, авторизация и права пользователей
5. Микрофреймворк Flask: проектирование RESTful API
6. Микрофреймворк Flask: тестирование и развертывание веб-приложения
7. Микрофреймворк Flask: безопасность веб-приложения

### 2. Примеры индивидуальных домашних заданий

#### Пример 1.

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  $\text{modify}(n) = n$ .

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

## **Пример 2.**

Используя фреймворк Flask, создайте веб-приложение для отгадывания загадок. Предусмотрите 2 вида пользователей: у администратора будет возможность загружать загадки, а у пользователей – возможность их отгадывать.

## **3. Примеры аудиторных контрольных работ:**

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 возбуждаться исключение.

2. Используя библиотеку `tkinter`, создайте программу, которая будет выводить на экран полный адрес и координаты краеведческого музея г. Магнитогорска.

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>Понимание областей и особенностей применения языков программирования высокого уровня; знание основных видов интегрированных сред разработки программного обеспечения;</p> <p>владение современными технологиями и методами программирования, предназначенными для создания прикладных программ;</p> <p>навыки реализации алгоритмов на языках программирования высокого уровня;</p> <p>владение навыками программирования с использованием эффективных реализаций структур данных и алгоритмов;</p> <p>знание основных методов эффективного кодирования;</p> <p>навыки кодирования алгоритмов с соблюдением требований к качественному стилю программирования;</p> <p>владение способами обработки исключительных ситуаций;</p> <p>навыки использования библиотек прикладных программ для решения прикладных задач профессиональной области;</p> <p>навыки использования динамически подключаемых библиотек;</p> <p>способность использовать языки, системы и инструментальные средства разработки автоматизированных систем;</p>	<p>Примеры практических заданий для экзамена:</p> <ol style="list-style-type: none"> <li>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.</li> <li>2. Используя язык Python, реализуйте генератор factorials(n), генерирующий последовательность факториалов натуральных чисел.</li> <li>3. На языке программирования Python объявите класс Автомобиль, содержащий не менее 4-х полей и 2-х методов. Создайте список из объектов класса Автомобиль.</li> <li>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.</li> </ol> <p>Теоретические вопросы к экзамену Технология разработки ПО:</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</li> </ol>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	<p>владение технологиями программирования распределенных автоматизированных систем;</p>	<p>разработки программного обеспечения: назначение, разновидности, краткая характеристика, примеры.</p> <p>5. Интегрированная среда разработки: область применения, состав, функции, примеры.</p> <p>6. Структурное программирование. Декомпозиция задачи и проектирование «сверху-вниз».</p> <p>7. Модульное программирование: принципы, сферы применения. Связность и сцепление модулей.</p> <p>8. Объектно-ориентированное программирование: основные принципы, сфера применения.</p> <p>Вопросы по синтаксису языка Python:</p> <p>24. Особенности языка программирования Python. Соглашения по оформлению кода Python, организация кода в модели и пакеты.</p> <p>25. Язык программирования Python: числовые, булевы и строковые литералы, основные операции над простыми типами.</p> <p>26. Язык программирования Python: синтаксические конструкции, условия, циклы, функция range().</p> <p>27. Язык программирования Python: системные библиотеки Python: math, time, random, os.</p> <p>28. Коллекции в языке программирования Python: строки</p> <p>29. Коллекции в языке программирования Python: списки, кортежи, множества. Методы списков и операции со списками. Срезы списков. Присваивание в срез. Копирование списка.</p> <p>30. Коллекции в языке программирования Python: словари</p> <p>31. Функции в языке программирования Python. Область видимости переменных. Именованные параметры функций. Значения параметров по умолчанию. Функции с переменным числом аргументов.</p> <p>32. Функциональное программирование в Python: списковые включения, встроенные</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>функции высших порядков, лямбда-функции.</p> <p>33. Функциональное программирование в Python: функции как объект, декораторы функций, замыкания, функторы.</p> <p>34. Функциональное программирование в Python: итераторы и итерируемые объекты, функции-генераторы, оператор yield.</p> <p>35. Особенности объектно-ориентированного подхода в Python: реализация инкапсуляции, наследования и полиморфизма средствами языка Python.</p> <p>36. Исключения в Python. Генерирование и перехват исключений.</p> <p>37. Библиотека tkinter: назначение, сфера применения, обзор возможностей, менеджеры геометрии.</p> <p>38. Библиотека numpy: назначение, сфера применения, обзор возможностей.</p> <p>39. Библиотека matplotlib: назначение, сфера применения, обзор возможностей.</p> <p>40. Работа с протоколом HTTP и API средствами языка программирования Python (библиотека requests)</p> <p>41. Работа с файловой системой и популярными форматами файлов: csv-формат.</p> <p>42. Работа с файловой системой и популярными форматами файлов: json-формат.</p> <p>43. Работа с файловой системой и популярными форматами файлов: xml-формат.</p> <p>44. Работа с файловой системой и популярными форматами файлов: zip-архивы.</p> <p>Веб программирование на Python</p> <p>1. Основные инструменты для разработки веб-приложений: HTML и CSS.</p> <p>2. Основные инструменты для разработки веб-приложений: JavaScript, библиотека jQuery.</p> <p>3. Браузерное окружение, DOM, браузерные события.</p> <p>4. Подход AJAX при разработке веб-приложений</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>5. Фреймворк Flask: особенности, принципы работы, основные этапы создания веб-приложения.</p> <p>6. Фреймворк Flask и библиотека sqlalchemy: технология объектно-реляционного отображения, работа с БД и моделями (models).</p> <p>7. Фреймворк Flask: работа с отображениями (views).</p> <p>8. Фреймворк Flask: HTML-шаблоны (templates), подключение и использование static-файлов (JS, CSS).</p> <p>9. Фреймворк Flask: работа с формами, библиотека flask-wtf.</p> <p>10. Фреймворк Flask: механизмы сессий, cookie, авторизация и права пользователей.</p>
<p>ОПК-7.2: Проектирует программное обеспечение с использованием средств автоматизации</p>	<p>Знание методологий и владение методами проектирования сложного программного обеспечения;</p> <p>владение основными навыками проектирования сложного программного обеспечения с использованием средств автоматизации;</p> <p>навыки формирования требований и разработки внешних спецификаций для разрабатываемого программного обеспечения, автоматизированных систем;</p> <p>навыки использования паттернов проектирования</p> <p>навыки разработки программной документации;</p> <p>знание принципов сопровождения программного обеспечения;</p> <p>знание показателей качества программного обеспечения;</p> <p>навыки выбора эффективных способов реализации профессиональных задач;</p> <p>навыки проектирования алгоритмов с соблюдением требований к качественному стилю программирования.</p>	<p>Теоретические вопросы к экзамену Технология разработки ПО:</p> <p>1. Типы архитектур программного обеспечения: монолитная архитектура, сервис-ориентированная архитектура, микросервисная архитектура. Привести примеры.</p> <p>2. Типы архитектур программного обеспечения: архитектура клиент-сервер, двухзвенная архитектура, трехзвенная архитектура. Привести примеры.</p> <p>3. Архитектура распределенных систем, архитектура облачных систем.</p> <p>4. API как средство интеграции приложений.</p> <p>5. Технологии OLE, DDE, COM, DCOM, CORBA: принцип работы, область использования, примеры.</p> <p>Архитектура веб-приложений:</p> <p>1. Основные виды веб приложений и их архитектура.</p> <p>2. Смена подходов к разработке веб-приложений. Концепция Web 2.0.</p> <p>3. Понятие веб-сервиса. Построение веб-сервисов с использованием протокола SOAP и архитекту-ры REST.</p> <p>4. Фреймворки для веб-разработки. Привести примеры. Модель MVC. Привести примеры.</p> <p>5. Протокол HTTP: назначение, принцип работы, виды HTTP-запросов.</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>Жизненный цикл разработки ПО:</p> <ol style="list-style-type: none"> <li>1. Жизненный цикл разработки программного обеспечения. Стандарты ISO 9000 в области разработки программного обеспечения.</li> <li>2. Модели жизненного цикла разработки программного обеспечения.</li> <li>3. Коллективная разработка ПО.</li> <li>4. Анализ требований к программному обеспечению.</li> <li>5. Разработка технического задания.</li> <li>6. Проектирование программного обеспечения: проектирование архитектуры, устройства компонент, пользовательского интерфейса.</li> <li>7. Основные принципы обеспечения качества программного обеспечения. Стандарты ISO 9000 в области качества программного обеспечения.</li> </ol>
<p>ОПК-7.3: Проводит комплексное тестирование и отладку программных систем</p>	<p>Владение методами тестирования и отладки программного обеспечения в соответствии с современными технологиями и методами программирования;</p> <p>умение проводить модульное, интеграционное, системное, приемочное и другие виды тестирования программных систем.</p>	<p>Примеры практических заданий для экзамена:</p> <ol style="list-style-type: none"> <li>1. Реализуйте на языке Python функцию <code>time2min(h, m)</code>, которая переводит часы и минуты в минуты с начала суток (00:00). Проведите тестирование функции, используя библиотеки <code>unittest</code> или <code>pytest</code>.</li> </ol> <p>Теоретические вопросы к экзамену:</p> <ol style="list-style-type: none"> <li>1. Тестирование и отладка программного обеспечения.</li> <li>2. Документирование программного обеспечения. Документирование кода в Python.</li> <li>3. Общие принципы тестирования программного обеспечения. Тестирование кода на Python: библиотеки <code>unittest</code>, <code>pytest</code>.</li> </ol>

**б) Примерный перечень индивидуальных заданий:**

1. Создание инструмента обнаружения хостов UDP.
2. Создание sniffера 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:**

**Показатели и критерии оценивания экзамена:**

– на оценку «отлично» – обучающийся должен показать высокий уровень знаний не только на уровне воспроизведения и объяснения информации, но и интеллектуальные навыки решения проблем и задач, нахождения уникальных ответов к проблемам, оценки и вынесения критических суждений;

– на оценку **«хорошо»** – обучающийся должен показать средний уровень знаний не только на уровне воспроизведения и объяснения информации, но и интеллектуальные навыки решения проблем и задач;

– на оценку **«удовлетворительно»** – обучающийся должен показать пороговый уровень знаний на уровне воспроизведения и объяснения информации, навыки решения типовых задач;

– на оценку **«неудовлетворительно»** – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать навыки решения типовых задач.