

**Частное образовательное учреждение
профессионального образования
Брянский техникум управления и бизнеса**

**ДОКУМЕНТ ПОДПИСАН
КВАЛИФИЦИРОВАННОЙ
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 09D41FB70039B39F944142467F307B5036
Владелец: Прокопенко Любовь Леонидовна
Действителен: с 14.08.2025 до 14.08.2026

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ

ПМ.01 Разработка кода для искусственного интеллекта

**по специальности 09.02.13 «Интеграция решений с применением технологий
искусственного интеллекта»**

Брянск 2026

СОДЕРЖАНИЕ

| | |
|--------------------------------------------------------|-----------|
| 1. ОБЩАЯ ХАРАКТЕРИСТИКА ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ | 3 |
| 2. ПЕРЕЧНИ ОЦЕНОЧНЫХ СРЕДСТВ | 7 |
| 3. КРИТЕРИИ ОЦЕНКИ ФОС | 15 |
| 4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ | 16 |

1. ОБЩАЯ ХАРАКТЕРИСТИКА ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ ПМ.01 «Разработка кода для искусственного интеллекта»

Фонд оценочных средств предназначен для проверки результатов освоения профессионального модуля программы специалистов среднего звена в соответствии с ФГОС по специальности 09.02.13 **Интеграция решений с применением технологий искусственного интеллекта**

В состав ПМ. 01 входит:

- МДК.01.01 «Разработка программных модулей в системах искусственного интеллекта»
- МДК.01.02 «Разработка мобильных приложений с поддержкой искусственного интеллекта»
- МДК.01.03 «Тестирование программных модулей»
- УП.01.01 «Учебная практика»
- ПП.01.01 «Производственная практика»
- ПМ.01.01 (К) Экзамен по модулю.

1.1. Формы промежуточной аттестации по ПМ.01

| Элементы | Формы промежуточной аттестации |
|------------------------------------------------------------------------------------|--------------------------------|
| МДК.01.01 «Разработка программных модулей в системах искусственного интеллекта» | экзамен |
| МДК.01.02 «Разработка мобильных приложений с поддержкой искусственного интеллекта» | экзамен |
| МДК.01.03 «Тестирование программных модулей» | дифференцированный зачет |
| УП.02.01 «Учебная практика» | Дифференцированный зачет |
| ПП.02.01 «Производственная практика» | Дифференцированный зачет |
| ПМ.01.01 (К) Экзамен по модулю. | Экзамен по модулю |

1.2. Результаты освоения профессионального модуля ПМ.01, подлежащие проверке

В результате контроля и оценки по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Общие компетенции:

| Код | Наименование общих компетенций |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ОК 01 | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. |
| ОК 02 | Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности |
| ОК 03 | Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях |
| ОК 04 | Эффективно взаимодействовать и работать в коллективе и команде. |
| ОК 05 | Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста. |
| ОК 06 | Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения. |
| ОК 07 | Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях. |
| ОК 09 | Пользоваться профессиональной документацией на государственном и иностранном языках. |

Профессиональные компетенции:

| Виды деятельности | Профессиональные компетенции, соответствующие видам деятельности |
|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| разработка кода для искусственного интеллекта | <p>ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.</p> <p>ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.</p> <p>ПК 1.3. Оформлять программный код в соответствии с техническим заданием.</p> <p>ПК 1.4. Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.</p> <p>ПК 1.5. Выполнять отладку программных модулей с использованием специализированных программных средств.</p> <p>ПК 1.6. Выполнять тестирование программного кода.</p> <p>ПК 1.7. Составлять тестовые сценарии.</p> |

В результате изучения профессионального модуля ПМ.01 обучающийся должен:

| Код ОК, ПК | Уметь | Знать | Владеть навыками |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ПК 1.1 | <p>Анализировать технические задания и выявлять требования к алгоритмам.</p> <p>Применять методы алгоритмизации для решения задач программирования.</p> <p>Разрабатывать оптимальные алгоритмы для решения задач в области ИИ.</p> | <p>Основные методы и подходы к построению алгоритмов (типичные поисковые алгоритмы, жадные алгоритмы, динамическое программирование, рекурсивные подходы).</p> <p>Принципы эффективной обработки данных.</p> <p>Языки программирования, применяемые для разработки алгоритмов.</p> | <p>Разработки, оптимизации и оценки сложности алгоритмов для ИИ-программ.</p> <p>Использования библиотек и инструментов для работы с алгоритмами и данными (например: Pandas, NumPy, Scikit-learn).</p> <p>Применения структур данных (деревья, графы, списки) для реализации алгоритмов.</p> |
| ПК 1.2 | <p>Реализовывать программные модули на основе требований технического задания.</p> <p>Соблюдать при разработке принципы «чистого кода».</p> <p>Использовать стандартные библиотеки и фреймворки для ускорения разработки.</p> | <p>Принципы модульного программирования.</p> <p>Языки программирования для разработки модулей.</p> <p>Стандартные фреймворки и библиотеки для работы с ИИ.</p> | <p>Разработки модульных ИИ-систем, соответствующих требованиям производительности и безопасности.</p> <p>Внедрения разработанных ИИ-модулей в комплексные программные системы.</p> <p>Оптимизации кода и работы с интерфейсами для взаимодействия между модулями.</p> |
| ПК 1.3 | Оформлять код в | Основные принципы | Оформления, |

| | | | |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | соответствии с принятыми стандартами и требованиями. Документировать разработанный программный код. Соблюдать соглашения о наименованиях переменных, функций и классов (например, PEP8 для Python). | чистого кода (Clean Code). Стандарты и практики документирования программного обеспечения. Инструменты для автоматической проверки качества кода (например, PyLint, ESLint). | документирования и структурирования кода для последующей поддержки. Использования инструментов статического анализа кода для выявления ошибок и улучшения качества. Работы с системами документирования кода (например, Doxygen, Sphinx). |
| ПК 1.4 | Работать с системами контроля версий для управления проектами. Организовывать совместную работу над проектом через ветки разработки и слияние изменений. Разрешать конфликты при слиянии кода. | Принципы работы распределенных систем контроля версий. Основные команды и операции в системах контроля версий (например: commit, pull, push, merge). Методы разрешения конфликтов в ходе групповой разработки. | Управления проектами с использованием систем контроля версий для организации командной работы. Разрешения конфликтов при слиянии веток и использования pull request для рецензирования кода. Настройки процессов CI/CD для автоматического тестирования и развертывания кода. |
| ПК 1.5 | Использовать инструменты для отладки программного кода. Идентифицировать и исправлять ошибки в программе. Применять методы логирования для анализа выполнения программ. | Принципы работы отладчиков и логирования. Способы выявления ошибок в программе (отладка по шагам, точки останова). Инструменты для отладки кода (например, PyCharm, Visual Studio Debugger). | Отладки программных модулей с использованием пошаговой проверки. Применения методов логирования и профилирования производительности. Использования специальных средств для отладки многопоточных программ. |
| ПК 1.6 | Проводить различные виды тестирования (юнит-тестирование, интеграционное тестирование). Выполнять настройки окружения и подготовку тестовых данных Фиксировать результаты выполнения тестов и подготавливать отчеты о результатах тестов. Определять уровень | Техники выполнения тестовых прогонов. Инструменты и среды выполнения тестирования Языки разработки автоматизированных тестов Инструменты для тестирования программного кода. Правила выполнения отчетов о тестировании | Выполнения статического тестирования программного кода на предмет выявления ошибок/дефектов алгоритмов, в том числе – на наличие обработки исключений Выполнения тестирования программных модулей в соответствии в тест- |

| | | | |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <p>критичности дефектов. Разрабатывать автоматизированные тесты для тестирования модулей и/или отдельных функций Восстанавливать окружение и тесты после сбоя</p> | | <p>планом Генерирования тестовых данных Выполнения интеграционного тестирования в соответствии с заданием Выполнения регрессионного тестирования в соответствии с заданием. Работы с CI/CD пайплайнами для автоматизации тестирования.</p> |
| ПК 1.7 | <p>Проектировать тестовые сценарии на основе тестовых планов. Разрабатывать тестовые пакеты и задания на выполнение тестирования. Использовать шаблоны для написания тест-кейсов. Оценивать риски при отборе тестов для регрессионного тестирования. Оценивать тесты на соответствие целям тестирования.</p> | <p>Цели, задачи и виды тестирования. Понятие стратегии тестирования. Жизненный цикл дефекта. Основы тест-дизайна: тестовый сценарий, тестовый пакет, чек-лист, основные шаблоны. Основные инструменты проектирования тестов. Методы и подходы к написанию тестов (Test-Driven Development, Behavior-Driven Development).</p> | <p>Разработки тестовых сценариев в соответствии с тестовым планом (тестирование производительности, надежности, UI-тестирование), в том числе с применением средств автоматизации проектирования. Разработки тестовых пакетов и заданий на выполнение тестирования. Оценки тестовых данных на предмет покрытия строк и покрытия ветвей, выполнения валидации данных. Автоматизации создания и выполнения тестовых сценариев.</p> |

2. ПЕРЕЧНИ ОЦЕНОЧНЫХ СРЕДСТВ УРОВНЯ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ ПМ.01 «РАЗРАБОТКА КОДА ДЛЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»

Оценка качества подготовки обучающихся по профессиональному модулю ПМ.01 «Разработка программных модулей в системах искусственного интеллекта» осуществляется в ходе экзамена по модулю ПМ.01.01.(К) «Экзамен по модулю» является формой аттестации по профессиональному модулю.

Перечень вопросов из МДК.01.01 «Разработка программных модулей в системе искусственного интеллекта»

1. Искусственный интеллект как научное направление.
2. Предмет, объект, метод, цель и задачи дисциплины "Системы искусственного интеллекта".
3. Зарождение исследований в области искусственного интеллекта (ИИ).
4. Два направления: логическое и нейрокибернетическое.
5. Ранние исследования в 50-60-е годы (Н.Винер, Мак-Каллок, Розенблатг, Саймон, Маккартни, Слэйджл, Сэмюэль, Гелернер, Н.Амосов).
6. Язык программирования LISP для построения систем ИИ.
7. Появление в конце 60-х годов интегральных (интеллектуальных) роботов и первых экспертных систем. Успехи экспертных систем застой в нейрокибернетике в 70-е годы.
8. Бум нейрокибернетики в начале 80-х годов (Хопфилд). Появление логического программирования и языка PROLOG.
9. Программа создания ЭВМ 5-го поколения. Стратегическая компьютерная инициатива США. Исследования по ИИ в СССР и России.
10. Свойства знаний и отличие знаний от данных.
11. Типы знаний: декларативные и процедурные, экстенциональные и интенциональные.
12. Нечеткие знания.
13. Виды и природа нечеткости.
14. Проблема понимания смысла как извлечения знаний из данных и сигналов.
15. Прикладные системы ИИ — системы, основанные на знаниях.
16. Понятие инженерии знаний.
17. Экспертные системы. Их области применения и решаемые ими задач.
18. Обобщенная структура экспертных систем.
19. Интеллектуальные роботы. Их обобщенная структура.
20. Системы общения на естественном языке и речевой ввод-вывод.
21. Применение ИИ в системах управления производством. Применение ИИ в делопроизводстве и в сети Internet.
22. Алгоритмические и логические модели представления знаний. Эвристические методы представления знаний.
23. Понятие предиката, формулы, кванторов всеобщности и существования. Интерпретация формул в логике предикатов 1-го порядка. Метод резолюции для доказательства теорем в логике 1-го порядка.
24. Логика Хорна как основа языка логического программирования Prolog. Недостатки логики 1-го порядка как метода представления знаний. Пути повышения выразительных возможностей логики 1-го порядка: введение модальностей и повышение значности.
25. Логика возможного-необходимого. Трехзначная семантика Лукасевича. Семантика возможных миров.
26. Правила-продукции. Структура правил продукций.
27. Типы ядер правил-продукций и варианты их интерпретаций. Граф И/ИЛИ и поиск данных. Управление выводом в продукционной системе.
28. Методы логического вывода: прямой и обратный.
29. Стратегии выбора правил при логическом выводе.
30. Важность качества данных для ИИ-моделей.
31. Методы сбора данных: веб-скрапинг, API, базы данных.

32. Методы предобработки данных: очистка данных, нормализация, кодирование категориальных данных, работа с пропусками и выбросами.
33. Подготовка данных для обучения моделей ИИ. Виды обучения: обучение с учителем, обучение без учителя, обучение с подкреплением.
34. Основные алгоритмы машинного обучения: линейная регрессия, логистическая регрессия, метод ближайших соседей (kNN), деревья решений, метод опорных векторов (SVM).
35. Кластеризация: k-means, агломеративная кластеризация. Системы рекомендаций.
36. Методы оценки качества моделей: точность, полнота, F-мера, ROC-кривые.
37. Валидация моделей: кросс-валидация, разделение данных на тренировочные и тестовые.
38. Регуляризация моделей: L1 и L2-регуляризация.
39. Оптимизация гиперпараметров моделей.
40. Глубокое обучение и нейронные сети.
41. Архитектуры нейронных сетей: многослойные перцептроны (MLP), сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN).
42. Процессы обучения нейронных сетей: обратное распространение ошибки, стохастический градиентный спуск, функции активации (ReLU, сигмоидальная).
43. Применение нейронных сетей в задачах классификации, распознавания образов и анализа временных рядов.

Перечень вопросов из МДК.01.02 «Разработка мобильных приложений с поддержкой искусственного интеллекта»

1. Введение в мобильную разработку: Android и iOS.
2. Установка и настройка Android Studio,
3. Создание первого Android-приложения.
4. Основы работы с Kotlin и Java для разработки мобильных приложений.
5. Использование TensorFlow Lite для встраивания моделей ИИ в мобильные приложения.
6. Применение предобученных моделей ИИ для распознавания изображений, текста и речи на мобильных устройствах.
7. Оптимизация моделей для работы на мобильных платформах
8. Взаимодействие с пользователем: разработка интуитивного интерфейса.
9. Применение ИИ в реальном времени: распознавание речи, работа с изображениями.
10. Взаимодействие с сенсорами устройства для получения данных.
11. Системы контроля версий для управления проектом.
12. Развертывание приложений в магазинах мобильных приложений.

Перечень вопросов из МДК.01.03 «Тестирование программных модулей»

1. Понятие качества программного обеспечения (ГОСТ Р ИСО/МЭК 25051).
2. Метрики качества. Определение целей тестирования. Уровни тестирования.
3. Виды тестирования: модульное, интеграционное, системное, приемочное.
4. Типы тестирования, основанные на спецификациях
5. Тестирование на основе сценариев использования.
6. Тестирование на основе диаграммы причинно-следственных связей
7. Виды тестирования производительности.
8. Регрессионное тестирование.
9. Жизненный цикл дефекта.
10. Уровни серьезности дефектов.
11. Понятие стратегии тестирования.
12. Тестовый сценарий. Тестовый план.
13. Чек-лист.

14. Тестовый пакет, задание на тестирование
 15. Шаблоны тестов.
 16. Основные инструменты проектирования тестов
«Черный ящик» или типы, основанные на спецификациях: эквивалентное разбиение, анализ граничных значений, использование таблиц решений, диаграммы причинно-следственных связей, тестирование переходов состояний, тестирование на основе сценариев использования
 17. Основные метрики оценки качества моделей.
 18. Матрица ошибок (confusion matrix) и её компоненты.
 19. Точность (Accuracy, Precision), недостатки метрик. Отклик (Recall или TPR, False Positive Rate, F1-score)
 20. Оценка качества модели при различных пороговых значениях: AUC-площадь (Area Under Curve) под кривой рабочих характеристик модели (ROC-кривой Receiver Operating Characteristics curve).
 21. Ограничения применения. Методы интерполяции
 22. Метрики регрессии, обучение линейной регрессии
 23. Средняя абсолютная погрешность (MAE - Mean Absolute Error). Средняя абсолютная процентная погрешность (MAPE - Mean Absolute Percentage Error). Другие оценки средних.
 24. Ограничения методов. Оптимальная сложность модели
- Метрики кластеризации
25. Инструменты для автоматизации выполнения тестовых примеров. Автоматизация тестов в CI/CD. Генераторы данных
 26. Тестирование мобильных ИИ-приложений. Основные проблемы AI-инструментов
 27. Понятие настройки окружения. Заглушки. Тестовые стенды. E2E тесты.
 28. Тестирование отказоустойчивости, стресс-тестирование, тестирование безопасности.
 29. Инструменты автоматизации тестирования веб-приложений с ИИ (например, SOAPUI, Postman)
 30. Мониторинг и профилирование производительности ИИ-приложений.

Перечень примерных практических заданий:

Задание 1.

Создание базовой модели ИИ для классификации данных.

Задание 2.

Предобработка данных для машинного обучения: очистка, нормализация, кодирование

Задание 3.

Применение кластеризации для сегментации данных.

Задание 4.

Оценка качества модели с использованием ROC-кривой и F-меры.

Задание 5.

Настройка гиперпараметров модели с использованием GridSearchCV.

Задание 6.

Создание сверточной нейронной сети для распознавания изображений.

Задание 7.

Проектирование архитектуры ИИ-системы с учетом модульности и масштабируемости.

Задание 8.

Контейнеризация ИИ-модели с использованием Docker.

Задание 9.

Развертывание ИИ-системы в Kubernetes.

Задание 10.

Подготовка тестового пакета и задания на тестирование

Задание 11.

Построение и визуализация матрицы ошибок

Задание 12.

Оценка качества нейронной сети с использованием ROC-кривой

Задание 13.

Разработка юнит-тестов для модели машинного обучения.

Задание 14.

Интеграция модели ИИ в веб-приложение.

Задание 15.

Тестирование безопасности ИИ-приложений.

Задание 16.

Тестирование совместимости с браузерами

Задание 17.

Мониторинг производительности ИИ-модели с использованием систем мониторинга и оповещения и мониторинга и визуализации данных

Перечень тестовых заданий по ПМ.01

1) Выбрать правильный ответ

Какой компонент обязательно должен быть установлен на компьютере для корректной работы Android Studio?

- А) Java Development Kit (JDK)
- Б) Microsoft .NET Framework
- В) Python
- С) Node.js

2) Какое важное условие часто рекомендуется соблюдать при выборе пути установки или создании имени пользователя ОС (особенно в Windows) для предотвращения сбоев в работе среды?

- А) Использовать только заглавные буквы
- Б) Использовать не более 5 символов в названии
- В) Избегать кириллицы (русских букв) в путях к папкам
- Г) Устанавливать программу только на диск С.

3) Для чего используется встроенный инструмент SDK Manager?

- А) Для сборки готового APK-файла
- Б) Для загрузки и управления версиями платформ Android, библиотеками и инструментами разработчика

- В) Для запуска виртуального устройства
- Г) Для публикации приложения в Google Play

4) Что означает аббревиатура API в контексте Android SDK

- А) Application Programming Interface — набор библиотек для взаимодействия ОС и приложения
- Б) Android Package Installer — формат установочного файла
- В) Accelerated Performance Indicator — уровень быстродействия эмулятора
- Г) Automated Project Initializer — стартовый мастер проектов

5) Какая технология отвечает за аппаратную виртуализацию для быстрого запуска Android Virtual Device (AVD) на процессорах Intel/AMD?

- А) OpenGL
- Б) Gradle
- В) HAXM / Hyper-V
- Г) Instant Run

6) Что нужно включить на физическом Android-устройстве, чтобы иметь возможность устанавливать и отлаживать на нем приложения прямо из Android Studio?

- А) Режим "Не беспокоить"
- Б) Экономия заряда батареи
- В) Режим "Для разработчиков" и отладку по USB
- Г) Автоматическое обновление

7) Какой инструмент отвечает за процесс сборки (компиляции и упаковки) проекта в Android Studio?

- А) AVD Manager
- Б) Gradle
- В) SDK Manager
- Г) ADB

8) Установите соответствие между уровнями тестирования и их описаниями в жизненном цикле программного обеспечения.

- А) Модульное (Unit) тестирование
- Б) Интеграционное тестирование
- В) Системное тестирование
- Г) Приемочное тестирование

1. Проверка всей системы в сборе на соответствие исходным бизнес-требованиям. Проводится перед передачей заказчику.
2. Тестирование взаимодействия между несколькими независимыми программными модулями или компонентами.
3. Проверка работы самого мелкого, изолированного участка кода (например, отдельной функции или метода).
4. Проверка продукта конечными пользователями для подтверждения того, что система решает их задачи

9) Установите соответствие между техниками тест-дизайна для программных модулей и принципами их работы.

- А) Эквивалентное разбиение
- Б) Анализ граничных значений
- В) Предугадывание ошибки
- Г) Таблица принятия решений

1. Тестирование данных, которые находятся на самых «краях» допустимых диапазонов (например, при лимите от 1 до 100)
2. Использование интуиции и опыта тестировщика для определения потенциально уязвимых мест в логике модуля.
3. Определение набора условий (входных параметров) и возможных вариантов развития событий для создания комбинаций тестов.
4. Деление всех возможных входных данных на группы (классы), где ожидается одинаковый результат, чтобы не проверять каждое значение отдельно

10) Соотнесите методы тестирования модулей с их фокусом проверки

- А) Метод «Белого ящика» (White box)
- Б) Метод «Черного ящика» (Black box)
- В) Метод «Серого ящика»

1. Тестировщик знает внутреннюю архитектуру и структуру кода; упор делается на полное покрытие кода тестами (покрытие веток, условий).
2. Комбинированный подход: фокус на логике приложения, но с базовым пониманием внутреннего устройства компонентов.
3. Тестировщик не видит исходный код и проверяет модуль исключительно через его входные и выходные данные (интерфейс)

11) Установите соответствие между понятиями модульного тестирования и их определениями

- А) Мок-объект (Mock)
- Б) Тестовое покрытие (Code coverage)
- В) Регрессионное тестирование
- Г) Assertion (Утверждение)

1. Метрика, показывающая, какая доля исходного кода (в процентах) была проверена тестами.
2. Проверка, что добавление нового функционала или исправление ошибок не сломало старую, уже проверенную логику.
3. Специальная логическая конструкция в коде теста, которая сравнивает фактический результат работы функции с ожидаемым (например, `\(\text{assert}\{x == 5\}\)`)

12) Расположите шаги разработки и выполнения модульного теста в правильном порядке

- А) Написание самого теста для проверки поведения метода.
- Б) Запуск теста и получение результата (красный/зеленый).
- В) Анализ спецификации (требований) к программному модулю.
- Г) Рефакторинг кода модуля или теста при необходимости.
- Д) Определение входных данных и ожидаемого результата.

13) Расположите шаги разработки через тестирование (TDD — Test-Driven Development) в правильном порядке.

- А) Написать минимальный код, который заставит тест пройти.
- Б) Написать тест, который проверяет будущий функционал (тест падает).
- В) Выполнить рефакторинг кода (улучшить структуру без изменения поведения).
- Г) Запустить все тесты, чтобы убедиться в их работоспособности

14) Расположите этапы подготовки и выполнения модульного теста согласно шаблону AAA (Arrange, Act, Assert).

- А) Получение фактического результата вызова метода (Act).
- Б) Подготовка исходных данных, моков и вызов тестируемого метода.

- В) Инициализация объектов и установка начальных условий (Arrange).
- Г) Проверка соответствия полученного результата ожидаемому (Assert).

15) Установите правильную последовательность этапов создания мобильного приложения с поддержкой искусственного интеллекта от идеи до публикации

- А) Постановка бизнес-задачи и определение того, какую проблему пользователя должен решать ИИ.
- Б) Сбор и разметка данных для обучения или дообучения нейросети.
- В) Выбор модели ИИ (локальная модель, облачные API или дообучение готовой).
- Г) Проектирование UX/UI и разработка клиентской части (Frontend) для смартфона.
- Д) Интеграция и обучение алгоритмов, настройка взаимодействия приложения с сервером.
- Е) QA-тестирование и проверка точности нейросети.
- Ж) Публикация в магазинах приложений.

16) Если в приложении используется внешняя языковая модель (например, OpenAI API, Google Gemini), установите логическую цепочку обработки запроса пользователя

- А) Пользователь вводит текст или загружает фото в интерфейсе приложения.
- Б) Мобильное приложение формирует JSON-запрос с токеном авторизации.
- В) Передача данных через интернет к облачному серверу провайдера ИИ.
- Г) Нейросеть обрабатывает запрос и генерирует ответ.
- Д) Сервер возвращает зашифрованный ответ (токены/символы) обратно в приложение.
- Е) Парсинг ответа в приложении и вывод информации на экран смартфона.

17) Расположите этапы работы с данными при создании встроенной (On-Device) модели машинного обучения внутри мобильного приложения

- А) Сбор сырых данных (датасета).
- Б) Очистка и предварительная обработка (нормализация, удаление шумов).
- В) Обучение модели на обучающей выборке.
- Г) Валидация и тестирование на тестовой выборке.
- Д) Конвертация модели в мобильный формат (например, TFLite или CoreML).
- Е) Внедрение конвертированной модели в бинарный код мобильного приложения.

18) В процессах контроля качества AI-приложений важна комплексная проверка. Установите последовательность шагов при выявлении аномалий в ответах ИИ

- А) Повторение действий пользователя для воспроизведения некорректного ответа ИИ.
- Б) Проверка входных данных (валидация перед отправкой в нейросеть).
- В) Анализ логов (logs) и времени отклика сетевого запроса.
- Г) Локализация проблемы: ошибка в коде приложения или галлюцинация самой нейросети.
- Д) Формирование отчета об ошибке (баг-репорта) для команды.
- Е) Корректировка промпта (системного сообщения) или дообучение модели.

19) Какой подход к интеграции ИИ является наиболее оптимальным для приложений, требующих работы в режиме офлайн

- А) Использование облачных API (например, OpenAI API)
- Б) Локальный запуск предварительно обученной квантованной модели (например, с помощью TFLite) Разработка мобильных приложений с поддержкой ИИ (Rgkript)
- В) Поточковая передача данных на сервер стороннего разработчика
- Г) Использование встроенного браузера (WebView)

20) Что такое «квантование» (Quantization) модели при разработке мобильного ИИ?

- А) Процесс сжатия исходного кода приложения для уменьшения размера APK/IPA
- Б) Преобразование весов модели из чисел с плавающей запятой (32-бит) в форматы с меньшей разрядностью (например, 16-бит или 8-бит) для экономии памяти и ускорения инференса Интеграция решений с ИИ (Shat29)

- В) Деление датасета на части для обучения нейросети
- Г) Шифрование пользовательских данных, отправляемых в нейросеть

**Правильный ответ на каждое задание оценивается одним баллом.
Максимальное количество баллов-20**

3. КРИТЕРИИ ОЦЕНКИ ФОС

ПМ.01 «Разработка кода для искусственного интеллекта»

Оценка экзамена, дифференцированного зачета, устного опроса выражается в баллах (при устном ответе).

«отлично» - студент показывает глубокие осознанные знания по освещаемому вопросу, владение основными понятиями, терминологией; владеет конкретными знаниями, умениями по данной дисциплине в соответствии с ФГОС СПО: ответ полный, доказательный, четкий, грамотный, иллюстрирован практическим опытом профессиональной деятельности;

«хорошо» – студент показывает глубокое и полное усвоение содержания материала, умение правильно и доказательно излагать программный материал. Допускает отдельные незначительные неточности в форме и стиле ответа;

«удовлетворительно» – студент понимает основное содержание учебной программы, умеет показывать практическое применение полученных знаний. Вместе с тем допускает отдельные ошибки, неточности в содержании и оформлении ответа: ответ недостаточно последователен, доказателен и грамотен;

«неудовлетворительно» – студент имеет существенные пробелы в знаниях, допускает ошибки, не выделяет главного, существенного в ответе. Ответ поверхностный, бездоказательный, допускаются речевые ошибки.

При оценивании письменных работ (ответов на контрольные вопросы, выполнении курсовых, контрольных работ, рефератов, выполнении практических заданий различного вида), учитывается правильность оформления работы и требования, предъявляемые к оценкам:

«отлично» - письменная работа отвечает всем требованиям, изложенным в методических указаниях по выполнению письменных работ; студент показывает глубокие осознанные знания по освещаемому вопросу, владение основными понятиями, терминологией; владеет конкретными знаниями, умениями по данной дисциплине в соответствии с ФГОС СПО: ответ полный, доказательный, четкий, грамотный, иллюстрирован практическим опытом профессиональной деятельности;

«хорошо» - письменная работа в основном отвечает всем требованиям, изложенным в методических указаниях по выполнению письменных работ, но допускаются отдельные незначительные ошибки в оформлении, незначительные неточности в форме изложения материала; студент показывает глубокое и полное усвоение содержания материала, умение правильно и доказательно излагать программный материал. Допускает отдельные незначительные неточности в форме и стиле ответа;

«удовлетворительно» - письменная работа в основном отвечает всем требованиям, изложенным в методических указаниях по выполнению письменных работ, но допускаются незначительные ошибки в оформлении, незначительные неточности в форме изложения материала, встречаются опечатки; студент понимает основное содержание учебной программы, умеет показывать практическое применение полученных знаний. Вместе с тем допускает отдельные ошибки, неточности в содержании и оформлении ответа: ответ недостаточно последователен, доказателен и грамотен;

«неудовлетворительно» - письменная работа не отвечает требованиям, изложенным в методических указаниях по выполнению письменных работ (содержание работы не раскрывает заявленную тему, нарушена логика изложения материала, не достигнута цель, не выполнены задачи исследования); студент имеет существенные пробелы в знаниях, допускает ошибки, не выделяет главного, существенного в ответе. Ответ поверхностный, бездоказательный, допускаются речевые ошибки.

4. Информационное обеспечение

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы

4.2.1. Основные источники:

1. Воронов, М. В. Системы искусственного интеллекта : учебник и практикум для среднего профессионального образования / М. В. Воронов, В. И. Пименов, И. А. Небаев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 268 с. — (Профессиональное образование). — ISBN 978-5-534-17699-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590238>

2. Станкевич, Л. А. Интеллектуальные системы и технологии : учебник и практикум для среднего профессионального образования / Л. А. Станкевич. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 478 с. — (Профессиональное образование). — ISBN 978-5-534-20364-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/587749>

3. Гниденко, И. Г. Технология разработки программного обеспечения : учебник для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/585518>

4. Щербак, А. В. Поддержка и тестирование программных модулей : учебник для среднего профессионального образования / А. В. Щербак. — Москва : Издательство Юрайт, 2026. — 145 с. — (Профессиональное образование). — ISBN 978-5-534-19290-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590253>

4.2.2. Дополнительные источники:

1. Аббясов, В. М. Промышленные роботы и робототехнические системы : учебник для среднего профессионального образования / В. М. Аббясов, С. Л. Петухов. — Москва : Издательство Юрайт, 2026. — 168 с. — (Профессиональное образование). — ISBN 978-5-534-16770-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/600285>

2. Чернышев, С. А. Основы программирования на Python : учебник для среднего профессионального образования / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 349 с. — (Профессиональное образование). — ISBN 978-5-534-17056-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588671>

3. Маркин, А. В. Программирование на SQL : учебник для среднего профессионального образования / А. В. Маркин. — Москва : Издательство Юрайт, 2026. — 435 с. — (Профессиональное образование). — ISBN 978-5-534-11093-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/587541>

4.2.3. Интернет – ресурсы:

1. Электронно-библиотечная система «ЮРАЙТ» - <https://www.biblio-online.ru>

2. Электронно-библиотечная система «IPRbooks» - <http://www.iprbookshop.ru>

3. Информационно-правовой портал «ГАРАНТ» - <http://www.garant.ru/>