

**Автономная некоммерческая организация профессионального образования  
«ПЕРМСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»  
(АНО ПО «ПГТК»)**

УТВЕРЖДЕНА  
Педагогическим советом АНО ПО «ПГТК»  
(протокол от 05.02.2026 № 01)  
Председатель Педагогического совета, директор  
И.Ф. Никитина



**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
УЧЕБНОЙ ДИСЦИПЛИНЫ**

**МДК.01.03 ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ**

для специальности

**09.02.13 «Интеграция решений с применением технологий  
искусственного интеллекта»**  
(код и наименование специальности)

Квалификация выпускника  
**Специалист по работе с искусственным интеллектом**

Форма обучения  
**Очная**

Пермь 2026

Фонд оценочных средств учебной дисциплины МДК.01.03 ТЕСТИРОВАНИЕ ПРОГРАММНЫХ МОДУЛЕЙ составлен в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта» (утвержден приказом Министерства Просвещения Российской Федерации от 24 декабря 2024 г. N 1025).

Программа предназначена для студентов и преподавателей АНО ПО «ПГТК».

Автор – составитель: Могильникова Н.С., старший преподаватель.

Фонд оценочных средств учебной дисциплины рассмотрена и одобрена на заседании кафедры математических и естественно-научных дисциплин, протокол, № 01 от 04.02.2026.

## Содержание ФОС УД

1. Паспорт фонда оценочных средств
  - 1.1. Область применения фонда оценочных средств
  - 1.2. Организация текущего контроля успеваемости и промежуточной аттестации по итогам освоения учебной дисциплины
2. Контроль и оценка достижения запланированных результатов обучения
  - 2.1. Перечень вопросов и заданий для текущего контроля знаний
  - 2.2. Перечень вопросов и заданий для промежуточной аттестации
  - 2.3. Критерии оценивания ПА
3. Рекомендуемая литература и иные источники

## 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

### 1.1. Область применения фонда оценочных средств

Фонд оценочных средств предназначен для оценивания достижений запланированных результатов по учебной дисциплины МДК.01.03 Тестирование программных модулей программы подготовки специалистов среднего звена (далее ППССЗ) по специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта».

Фонд оценочных средств (ФОС) представляет собой комплект материалов для проведения промежуточной аттестации и текущего контроля.

Результаты обучения - это усвоенные знания и освоенные умения по дисциплине в целях овладения предусмотренных стандартом общих и профессиональных компетенций.

Фонд оценочных средств позволяет оценивать формирование элементов профессиональных компетенций (ПК) и элементов общих компетенций (ОК) через освоение умений, знаний и навыков.

Код ОК, ПК	Уметь	Знать
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием. ПК 1.3 Оформлять программный код в соответствии с техническим заданием. ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки. ПК 1.5 Выполнять отладку программных модулей с использованием специализированных	Анализировать технические задания и выявлять требования к алгоритмам. Применять методы алгоритмизации для решения задач программирования. Разрабатывать оптимальные алгоритмы для решения задач в области ИИ. Реализовывать программные модули на основе требований технического задания. Соблюдать при разработке принципы «чистого кода». Использовать стандартные библиотеки и фреймворки для ускорения разработки.  Оформлять код в соответствии с принятыми стандартами и требованиями. Документировать разработанный программный код. Соблюдать соглашения о наименованиях переменных, функций и классов (например, PEP8 для Python). Работать с системами контроля версий для управления проектами. Организовывать совместную работу над проектом через ветки разработки и слияние изменений. Разрешать конфликты при слиянии кода.  Использовать инструменты для отладки программного кода.	Основные методы и подходы к построению алгоритмов ( типовые поисковые алгоритмы, жадные алгоритмы, динамическое программирование, рекурсивные подходы). Принципы эффективной обработки данных. Языки программирования, применяемые для разработки алгоритмов. Принципы модульного программирования. Языки программирования для разработки модулей. Стандартные фреймворки и библиотеки для работы с ИИ.  Основные принципы чистого кода (Clean Code). Стандарты и практики документирования программного обеспечения. Инструменты для автоматической проверки качества кода (например, PyLint, ESLint).  Принципы работы распределенных систем контроля версий. Основные команды и операции в системах контроля версий (например: commit, pull, push, merge).

<p>программных средств. ПК 1.6 Выполнять тестирование программного кода. ПК 1.7 Составлять тестовые сценарии. ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста</p>	<p>Идентифицировать и исправлять ошибки в программе. Применять методы логирования для анализа выполнения программ. 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 критичности дефектов. Разрабатывать автоматизированные тесты для тестирования модулей и/или отдельных функций Восстанавливать окружение и тесты после сбоя 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>распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы владеть актуальными методами работы в профессиональной и смежных сферах оценивать результат и последствия своих действий (самостоятельно или с помощью наставника) определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска оценивать практическую значимость результатов поиска</p>	<p>Методы разрешения конфликтов в ходе групповой разработки.</p> <p>Принципы работы отладчиков и логирования. Способы выявления ошибок в программе (отладка по шагам, точки останова). Инструменты для отладки кода (например, PyCharm, Visual Studio Debugger). Техники выполнения тестовых прогонов. Инструменты и среды выполнения тестирования Языки разработки автоматизированных тестов Инструменты для тестирования программного кода. Правила выполнения отчетов о тестировании</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). актуальный профессиональный и социальный контекст, в котором приходится работать и жить структура плана для решения задач, алгоритмы выполнения работ в профессиональной и смежных областях основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте методы работы в профессиональной и смежных сферах порядок оценки результатов решения задач профессиональной деятельности номенклатура информационных источников, применяемых в профессиональной деятельности приемы структурирования информации</p>
---	--	--

	применять средства информационных технологий для решения профессиональных задач использовать современное программное обеспечение в профессиональной деятельности использовать различные цифровые средства для решения профессиональных задач грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке проявлять толерантность в рабочем коллективе	формат оформления результатов поиска информации современные средства и устройства информатизации, порядок их применения программное обеспечение в профессиональной деятельности, в том числе цифровые средства психологические основы деятельности коллектива правила оформления документов правила построения устных сообщений особенности социального и культурного контекста
--	--	---

## **1.2. Организация текущего контроля успеваемости и промежуточной аттестации по итогам освоения программы учебной дисциплины**

В период обучения по образовательной программе СПО осуществляется текущий контроль успеваемости студентов, промежуточная аттестация по учебным дисциплинам и профессиональным модулям.

Текущий контроль осуществляется в пределах учебного времени, отведенного на учебную дисциплину, оценивается по пятибалльной шкале. Текущий контроль проводится с целью объективной оценки качества освоения программы дисциплины, а также стимулирования учебной деятельности студентов, подготовки к промежуточной аттестации и обеспечения максимальной эффективности учебного процесса. Для оценки качества подготовки используются различные формы и методы контроля. Текущий контроль учебной дисциплины осуществляется в форме устного опроса; защиты практических заданий, реферата, творческих работ; выполнения контрольных и тестовых заданий; решения ситуационных задач и других форм контроля, предусмотренных программой учебной дисциплины.

Промежуточная аттестация проводится в форме, предусмотренной планом учебного процесса: дифференцированного зачета, экзамена.

В период сложной санитарно-эпидемиологической обстановки или других ситуациях невозможности очного обучения и проведения аттестации студентов колледж реализует образовательные программы или их части с применением электронного обучения, дистанционных образовательных технологий в предусмотренных законодательством формах обучения или при их сочетании, при проведении учебных занятий, практик, текущего контроля успеваемости, промежуточной аттестации обучающихся.

Форма промежуточной аттестации по учебной дисциплине МДК.01.03 Тестирование программных модулей - дифференцированный зачет.



## 2. КОНТРОЛЬ И ОЦЕНКА ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1. Перечень вопросов и заданий для текущего контроля

В результате текущей аттестации по учебной дисциплине МДК.01.03 Тестирование программных модулей осуществляется проверка сформированности умений и знаний, направленных на формирование соответствующих ФГОС СПО общих и профессиональных компетенций.

#### Основы тестирования программных приложений

1. Что такое тестирование программного обеспечения?*Ответ:* Тестирование программного обеспечения — это систематический процесс выявления дефектов, ошибок и несоответствий в программе, направленный на обеспечение высокого качества продукта и соответствия установленным требованиям.
2. Назовите и кратко опишите основные виды тестирования программного обеспечения.*Ответ:* Основные виды тестирования:
  - Функциональное тестирование — проверка поведения программы с точки зрения функциональных требований.
  - Регрессионное тестирование — повторное тестирование после внесения изменений, чтобы удостовериться, что исправленные участки не нарушили существующие функциональные возможности.
  - Нагрузочное тестирование — проверка реакции программы на большие объемы данных и нагрузку.
  - Юзабилити-тестирование — проверка удобства и интуитивности интерфейса программы пользователями.
  - Безопасность — выявление потенциальных угроз и рисков нарушения безопасности данных.
3. В чем состоит цель юнит-тестирования?*Ответ:* Юнит-тестирование направлено на проверку отдельных элементов программы (модулей, классов, функций), чтобы убедиться, что каждый компонент функционирует должным образом отдельно от других.
4. Что такое smoke testing (дымовое тестирование)?*Ответ:* Smoke testing — это быстрый начальный вид тестирования, предназначенный для проверки основной функциональности приложения, чтобы убедиться, что оно готово к проведению более глубокого тестирования.
5. Какие этапы входят в стандартный процесс тестирования программного обеспечения?*Ответ:* Этапы процесса тестирования:
  - Планирование и дизайн тестирования.
  - Исполнение тестов.
  - Документирование обнаруженных дефектов.
  - Повторное тестирование и фиксация исправлений.
  - Генерация отчетов и документирование результатов.
6. Что такое bug tracking system (система отслеживания ошибок)?*Ответ:* Bug tracking system — это специализированное программное средство для регистрации, учета и контроля выявленных дефектов в течение всего жизненного цикла разработки программного обеспечения.
7. Перечислите и объясните основные уровни тестирования программного обеспечения.*Ответ:* Уровни тестирования:
  - Модульное тестирование — проверка отдельных единиц программы.
  - Интеграционное тестирование — проверка взаимодействия двух или более компонентов программы.
  - Системное тестирование — комплексное тестирование всего приложения целиком.
  - Приемочное тестирование — подтверждение готовности продукта заказчиком или



пользователями.

8. Что такое регрессионное тестирование и зачем оно нужно?*Ответ:* Регрессионное тестирование — это повторное прохождение тестов после внесения изменений в программу, чтобы убедиться, что новое поведение не нарушает прежнюю функциональность.
9. Назовите основные характеристики качественного теста.*Ответ:* Характеристики качественного теста:
  - Адекватность покрытия функционала.
  - Независимость от конкретной реализации.
  - Прозрачность и воспроизводимость.
  - Масштабируемость и простота обновления.
10. Каковы основные цели нагрузочного тестирования?*Ответ:* Нагрузочное тестирование проводится для оценки поведения программы при высоких уровнях нагрузки (количество запросов, пользователей, транзакций), выявления узких мест и определения максимальных рабочих характеристик системы.
11. Какие задачи решает автоматизированное тестирование и в чем его преимущества по сравнению с ручным тестированием?*Ответ:* Автотестирование ускоряет процесс тестирования, снижает вероятность человеческого фактора, повышает покрытие тестами, облегчает повторное тестирование и снижает расходы на эксплуатацию продукта.
12. Что такое monkey testing (обезьянье тестирование)?*Ответ:* Monkey testing — это форма хаотического тестирования, при которой случайно генерируемые входные данные вводятся в систему для выявления неожиданных сбоев и ошибок.
13. В чем разница между black-box и white-box видами тестирования?*Ответ:* Black-box тестирование основано на проверке внешнего поведения программы без знания внутренней структуры, в то время как white-box тестирование основывается на знании внутреннего устройства программы и позволяет проверять внутренние элементы и пути выполнения кода.
14. Что такое exploratory testing (исследовательское тестирование)?*Ответ:* Exploratory testing — это интерактивный подход к тестированию, при котором тестирующий одновременно исследует продукт и создает тесты, основываясь на интуиции и опыте, без строгих планов и сценариев.
15. Какие инструменты используются для автоматизированного тестирования веб-приложений?*Ответ:* Например, Selenium, Cypress, Playwright, TestComplete, JMeter и др., каждый из которых специализируется на определенных видах тестов и технологиях.
16. Какую роль играет QA инженер в проекте разработки программного обеспечения?*Ответ:* QA инженеры участвуют в обеспечении качества продукта, определяя стратегию тестирования, создавая планы и сценарии, проводя тестирование, документируя дефекты и координируя усилия по устранению неполадок.
17. Что такое acceptance criteria (критерии приемки)?*Ответ:* Acceptance criteria — это четко сформулированные условия, которым должна соответствовать программа для удовлетворения потребностей заказчика и пользователей.
18. Какие методы тестирования используются для оценки юзабилити (usability testing)?*Ответ:* Usability testing включает наблюдения за поведением пользователей, опросы, фокус-группы, eye-tracking и другое, чтобы понять удобство использования продукта и выявить неудобства или барьеры для пользователей.
19. Что такое coverage metrics (показатели покрытия)?*Ответ:* Coverage metrics показывают процент охвата кода тестами, позволяя количественно оценить полноту тестирования.
20. Зачем проводятся стресс-тесты и чем они отличаются от нагрузочных тестов?*Ответ:* Stress testing проводится для проверки поведения системы при экстремальных условиях нагрузки, превышающей обычные рабочие пределы, чтобы выяснить устойчивость и реакцию на непредвиденную нагрузку.

## **Основы тест-дизайна**

### **Практическая работа. Подготовка тестового пакета и задания на тестирование**

Цель работы:

Научиться составлять эффективный тестовый пакет, который позволит выявить максимальное количество дефектов и обеспечить стабильную работу приложения.

Порядок выполнения работы:

1. Определите объект тестирования: выберите готовое программное приложение (web-сайт, мобильное приложение или десктопное приложение).
2. Создайте тестовый пакет: составьте исчерпывающий набор тестов, покрывающих все основные и второстепенные функции объекта тестирования. Включайте тесты на граничные условия, ошибочные ситуации и уникальные кейсы.
3. Классификация тестов: сгруппируйте тесты по категориям (функциональные, нефункциональные, интеграционные, регрессионные и т.п.).
4. Задание на тестирование: напишите инструкцию для исполнителя тестов, включающую порядок действий, ожидаемые результаты и среду выполнения.
5. Документация: создайте таблицу для записи результатов тестирования, куда будете заносить информацию обо всех пройденных тестах, зафиксированных дефектах и мерах по их устранению.
6. Выполните тестирование: исполните созданный вами тест-пакет, записывая все результаты в таблицу.
7. Анализ результатов: проанализируйте результаты тестирования, выделив наиболее частые ошибки и слабые стороны приложения.
8. Рекомендации по улучшению: на основании проведенного анализа сформулируйте рекомендации по повышению качества и надежности приложения.

### **Практическая работа. Подготовка тестового сценария**

Цель работы:

Освоить процесс составления тестового сценария, позволяющего последовательно протестировать каждую важную функцию приложения.

Порядок выполнения работы:

1. Определение целей тестирования: четко обозначьте цели, которые хотите достичь с помощью сценария (проверка определенной функции, выполнение определенного сценария использования и т.д.).
2. Описание тестового сценария: создайте сценарий, который описывает шаг за шагом каждое действие, которое необходимо совершить для проверки интересующей вас функции.

Обязательно укажите ожидаемое поведение системы на каждом этапе.

3. Тип сценария: укажите тип сценария (регрессионный, исследовательский, load test и т.д.) и объясните, почему именно этот тип подходит для выбранного случая.
4. Контрольные точки: определите ключевые моменты, в которых система должна выдавать определенный результат, и добавьте их в сценарий.
5. Условия завершения: установите правила, по которым считается, что сценарий успешно выполнен (например, отсутствие ошибок, достижение желаемого результата).
6. Тестирование сценария: исполнив сценарий вручную или автоматически, проверьте соответствие фактических результатов запланированным.
7. Итоговый отчет: внесите все замечания и отклонения в итоговый отчет, сделайте выводы о качестве проверяемых функций и дайте рекомендации по совершенствованию продукта.

### **Система оценивания:**

Содержание и объем материала, подлежащего проверке в контрольной работе, определяется программой. При проверке усвоения материала выявляется полнота, прочность усвоения учащимися теории и умение применять ее на практике в знакомых и незнакомых ситуациях.

Отметка зависит также от наличия и характера погрешностей, допущенных учащимися.

- *грубая ошибка* - полностью искажено смысловое значение понятия, определения;
- *погрешность* отражает неточные формулировки, свидетельствующие о нечетком представлении рассматриваемого объекта;
- *неДочет* - неправильное представление об объекте, не влияющего кардинально на знания определенные программой обучения;
- *мелкие погрешности* - неточности в устной и письменной речи, не искажающие смысла ответа или решения, случайные описки и т.п.

Эталоном, относительно которого оцениваются знания учащихся, является обязательный минимум содержания информатики и информационно-коммуникационных технологий.

Исходя из норм (четырёхбалльной системы), заложенных во всех предметных областях выставляете отметка:

«5» ставится при выполнении всех заданий полностью или при наличии 1 -2 мелких погрешностей;

«4» ставится при наличии 1-2 недочетов или одной ошибки;

«3» ставится при выполнении 2/3 от объема предложенных заданий;

«2» ставится, если допущены существенные ошибки, показавшие, что учащийся не владеет обязательными умениями поданной теме в полной мере (незнание основного программного материала)

### **Тест по теме «Тестирование ИИ-приложений»**

Часть 1. Теоретические вопросы (выбрать один правильный ответ)

1. Что отличает тестирование ИИ-приложений от обычного тестирования программного обеспечения?

- A) Больше внимание уделяется дизайну пользовательского интерфейса.
  - B) Главное внимание уделяется качеству и количеству используемой обучающей выборки.
  - C) Больше разнообразие и непредсказуемость возможных состояний системы.
  - D) Основное внимание уделено скорости выполнения программы.
2. Как называется дефект, связанный с чрезмерной привязкой модели к обучающим данным?
- A) Regression.
  - B) Overfitting.
  - C) Debugging.
  - D) Validation error.
3. Как называют метод тестирования, при котором специально создаются неправильные входные данные для проверки реакции системы?
- A) Load testing.
  - B) Fuzz testing.
  - C) Integration testing.
  - D) Security testing.
4. Что такое regressive testing в контексте ИИ-приложений?
- A) Тестирование на старых данных для подтверждения обратной совместимости.
  - B) Периодическое повторное тестирование после внесения изменений в код.
  - C) Анализ новых данных для оценки прогностической силы модели.
  - D) Повторное тестирование для восстановления работоспособности старой версии.
5. Какой тип тестирования предназначен для проверки, насколько быстро и точно модель ИИ способна обрабатывать запросы в режиме реального времени?
- A) Performance testing.
  - B) Functional testing.
  - C) Security testing.
  - D) Compatibility testing.
6. Что такое Adversarial examples в контексте тестирования ИИ?
- A) Специально созданные образцы данных, приводящие к сбоям в работе модели.
  - B) Атаки хакеров на инфраструктуру ИИ-приложений.
  - C) Случаи неправильного функционирования из-за плохой обучающей выборки.
  - D) Выход за рамки допустимых условий работы модели.
7. Что такое cross-validation и зачем оно используется в тестировании ИИ-приложений?
- A) Это метод, позволяющий многократно проводить тесты на разных подмножествах данных.

- B) Это проверка совместимости приложения с различными браузерами и ОС.
  - C) Это тестирование, выполняемое сторонними организациями.
  - D) Это методика тестирования на ограниченном количестве экземпляров.
8. Что значит term “bias” в контексте машинного обучения?
- A) Количество дефектов в модели.
  - B) Отклонение результатов от истинных значений вследствие неравномерной обучающей выборки.
  - C) Временная задержка в работе приложения.
  - D) Неправильная настройка параметров модели.
9. Какая разновидность тестирования важна для выявления ситуаций, когда ИИ принимает неправильное решение при наличии редких аномалий в данных?
- A) Stress testing.
  - B) Outlier detection testing.
  - C) Functional testing.
  - D) Load testing.
10. Что означает термин «data drift» в контексте ИИ-приложений?
- A) Постепенное ухудшение точности модели из-за изменения характера поступающих данных.
  - B) Потеря данных при передаче по сети.
  - C) Нарушение целостности базы данных.
  - D) Появление новых моделей в приложении.

## Часть 2. Практические вопросы (открытая форма)

11. Опишите шаги по тестированию ИИ-приложения, предназначенного для распознавания лиц.
12. Почему классический подход к тестированию (black box, white box) может оказаться неэффективным для ИИ-приложений?
13. Приведите пример возможного сценария тестирования для чат-бота, основанного на искусственном интеллекте.
14. Какие инструменты и платформы вы используете для тестирования ИИ-приложений? Аргументируйте свой выбор.
15. Объясните, почему возникает overfitting и как с ним бороться.
16. Каким образом можно проверить надежность модели ИИ при изменении контекста использования (например, смена региона, языка)?
17. В чем принципиальное отличие тестирования модели, обученной на большом объеме данных, от традиционной функциональной проверки?

18. Какие существуют специальные подходы к тестированию нейронных сетей?
19. Что такое «Adversarial attacks» и как они связаны с тестированием ИИ-приложений?
20. Опишите ситуацию, когда нагрузочное тестирование (load testing) окажется важным для ИИ-приложения.

Критерии оценки:

- За каждый правильный ответ в первой части начисляется 1 балл.
- Во второй части оценивается полнота и правильность ответа по пятибалльной шкале (5 баллов за идеальный ответ, 3 балла за частично верный ответ, 1 балл за минимальный прогресс).

Общая максимальная оценка за тест составляет 30 баллов.

## **2.2. Перечень вопросов и заданий для промежуточной аттестации**

**Форма: дифференцированный зачет**

1. Что такое тестирование программного обеспечения? Его цели и задачи.
2. Какие существуют основные этапы жизненного цикла тестирования?
3. Что такое тест-кейс? Какие обязательные атрибуты должны присутствовать в описании тест-кейса?
4. Какие методы тестирования программного обеспечения принято выделять?
5. Какие основные разновидности тестирования по видам (функциональное, нагрузочное, интеграционное, регрессионное, юзер-тестирование и т.д.)?
6. Что такое White Box и Black Box тестирование? В чем их отличия и достоинства?
7. Каковы основные принципы хорошего тест-кейса?
8. Что такое test-driven development (TDD)? Когда его целесообразно применять?
9. Какие показатели качества программного обеспечения выделяются? (функциональность, надежность, производительность, безопасность и т.д.)
10. Что такое мок-апы (mock objects)? Где и когда они используются?
11. Какие инструменты применяются для автотестирования GUI-приложений?
12. Что такое непрерывное тестирование (Continuous Testing)? Какие инструменты поддерживают эту концепцию?
13. Как устроен процесс ручного тестирования? Чем он отличается от автоматизированного?

14. Какие подходы к автоматизации тестирования существуют? (BDD, ATDD, Page Object Pattern и т.д.)
15. Что такое нагрузка на систему и как её измеряют? Какие показатели используются при нагрузочном тестировании?
16. Какие задачи решает интеграционное тестирование?
17. Что такое регрессионное тестирование? Как оно проводится?
18. Какие цели преследует нагрузочное тестирование?
19. Что такое отказоустойчивость (fault tolerance)? Как её проверяют?
20. Какие бывают стратегии покрытия кода тестами (statement coverage, branch coverage, path coverage)?
21. Как выбирается подходящий тип тестирования для конкретной задачи?
22. Какие типы ошибок чаще всего встречаются в программных продуктах и как их предотвратить?
23. Что такое статический анализ кода? Как он связан с качеством ПО?
24. Что такое рефакторинг и как он влияет на качество тестирования?
25. Какие роли в команде играют специалисты по тестированию? Как организована их деятельность?
26. Что такое тестирование программного обеспечения? Его цели и задачи.
27. Какие существуют основные этапы жизненного цикла тестирования?
28. Что такое тест-кейс? Какие обязательные атрибуты должны присутствовать в описании тест-кейса?
29. Какие методы тестирования программного обеспечения принято выделять?
30. Какие основные разновидности тестирования по видам (функциональное, нагрузочное, интеграционное, регрессионное, юзер-тестирование и т.д.)?
31. Что такое White Box и Black Box тестирование? В чем их отличия и достоинства?
32. Каковы основные принципы хорошего тест-кейса?
33. Что такое test-driven development (TDD)? Когда его целесообразно применять?
34. Какие показатели качества программного обеспечения выделяются? (функциональность, надежность, производительность, безопасность и т.д.)
35. Что такое мок-апы (mock objects)? Где и когда они используются?
36. Какие инструменты применяются для автотестирования GUI-приложений?
37. Что такое непрерывное тестирование (Continuous Testing)? Какие инструменты поддерживают эту концепцию?

38. Как устроен процесс ручного тестирования? Чем он отличается от автоматизированного?
39. Какие подходы к автоматизации тестирования существуют? (BDD, ATDD, Page Object Pattern и т.д.)
40. Что такое нагрузка на систему и как её измеряют? Какие показатели используются при нагрузочном тестировании?
41. Какие задачи решает интеграционное тестирование?
42. Что такое регрессионное тестирование? Как оно проводится?
43. Какие цели преследует нагрузочное тестирование?
44. Что такое отказоустойчивость (fault tolerance)? Как её проверяют?
45. Какие бывают стратегии покрытия кода тестами (statement coverage, branch coverage, path coverage)?
46. Как выбирается подходящий тип тестирования для конкретной задачи?
47. Какие типы ошибок чаще всего встречаются в программных продуктах и как их предотвратить?
48. Что такое статический анализ кода? Как он связан с качеством ПО?
49. Что такое рефакторинг и как он влияет на качество тестирования?
50. Какие роли в команде играют специалисты по тестированию? Как организована их деятельность?
51. Как выстроить процесс повторного тестирования после исправления ошибок?
52. Что такое технический долг (technical debt)? Как его контролировать?
53. Какие этапы проходят артефакты тестирования (test artifacts)?
54. Как оценивается готовность продукта к выпуску (release readiness)?
55. Как организован процесс тестирования мобильных приложений?
56. Что такое риск-ориентированное тестирование? Какие риски учитываются при проведении такого тестирования?
57. Какие метрики качества тестирования можно выделить? (defect density, defect escape rate, code coverage и т.д.)
58. Какие существуют международные стандарты тестирования программного обеспечения (ISO, IEEE)?
59. Как формируется план тестирования (test plan)? Какие разделы он обязательно должен содержать?
60. Как организовано тестирование безопасности (security testing)? Какие виды атак проверяются?



61. Что такое usability-тестирование? Какие методики используются для его проведения?
62. Какие проблемы могут возникать при автоматизированном тестировании и как их решать?
63. Как определяется критерий останова тестирования (stopping criterion)?
64. Какие ошибки и сложности возникают при интеграции различных модулей и подсистем?
65. Что такое End-to-end тестирование и как оно применяется на практике?

### ***Критерии оценивания устного ответа на экзамене***

**Оценка «5» («отлично»)** соответствует следующей качественной характеристике: «изложено правильное понимание вопроса и дан исчерпывающий на него ответ, содержание раскрыто полно, профессионально, грамотно».

Выставляется студенту,

- усвоившему взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившему творческие способности в понимании, изложении и использовании учебно-программного материала;
- обнаружившему всестороннее систематическое знание учебно-программного материала, четко и самостоятельно (без наводящих вопросов) отвечающему на вопрос билета.

**Оценка «4» («хорошо»)** соответствует следующей качественной характеристике: «изложено правильное понимание вопроса, дано достаточно подробное описание предмета ответа, приведены и раскрыты в тезисной форме основные понятия, относящиеся к предмету ответа, ошибочных положений нет».

Выставляется студенту,

- обнаружившему полное знание учебно-программного материала, грамотно и по существу отвечающему на вопрос билета и не допускающему при этом существенных неточностей;
- показавшему систематический характер знаний по дисциплине и способному к их самостоятельному пополнению и обновлению в ходе дальнейшей учебы и профессиональной деятельности.

### **Оценка «3» («удовлетворительно»)**

Выставляется студенту,

- обнаружившему знание основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, предусмотренных программой;
- допустившему неточности в ответе и при выполнении экзаменационных заданий, но обладающими необходимыми знаниями для их устранения под руководством преподавателя.

### **Оценка «2» («неудовлетворительно»)**

Выставляется студенту,

- обнаружившему существенные пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий;
- давшему ответ, который не соответствует вопросу экзаменационного билета.

### 3. Рекомендуемая литература и иные источники

#### Основные источники:

1. Николукин, М. С. Тестирование программного обеспечения : учебное пособие / М. С. Николукин, В. В. Конкина, К. И. Патутин. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2025. — 80 с. — ISBN 978-5-8265-2883-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/154963.html>

2. Алетдинова, А. А. Системы искусственного интеллекта : учебное пособие / А. А. Алетдинова, М. Г. Гриф. — Новосибирск : Новосибирский государственный технический университет, 2023. — 76 с. — ISBN 978-5-7782-5124-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/156060.html>

#### Дополнительная литература:

1. Безопасность систем искусственного интеллекта. Ч.2. Доверенный искусственный интеллект : учебное пособие / П. С. Ложников, А. Е. Смотуга, С. С. Жумажанова, А. Е. Сулаво. — Омск : Омский государственный технический университет, 2023. — 74 с. — ISBN 978-5-8149-3614-1, 978-5-8149-3731-5 (ч.2). — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/140828.html>

#### ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п.п.	Содержание изменения	Дата, номер протокола заседания кафедры, подпись зав.кафедрой
1	2	3
1		
2		

3		
4		