

РАССМОТРЕНО

на заседании методической  
комиссии гуманитарных и программно-  
вычислительных дисциплин

Председатель МК  Стрודה Т.Н.

Протокол №10 « 14 » 05 2025 г.

УТВЕРЖДАЮ

Зам. директора по учебной работе



И.А. Овчинникова

« 14 » 05 2025 г

СОГЛАСОВАНО

Генеральный директор ООО «Айти Грэйд»



М.А. Тапцов

« 14 » 05 2025 г.

### КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ

для промежуточной аттестации (экзамен квалификационный)  
по ПМ.01. Разработка кода для обучения искусственного интеллекта  
для специальности

**09.02. 13 Интеграция решений с применением технологий искусственного интеллекта**

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

Составитель: Котяткина А. Н. – преподаватель СКТ(ф)СПбГУТ

Экзамен квалификационный является итоговой формой контроля по профессиональному модулю и проверяет готовность студента к выполнению указанного вида профессиональной деятельности, сформированности у него компетенций, определенных в разделе «Требования к результатам освоения ППССЗ» ФГОС СПО.

Всем студентам предлагается единый вариант задания. Экзаменационное задание для всех студентов является одинаковым, чтобы обеспечить всем равные условия выполнения. При выполнении заданий студенты могут пользоваться персональными компьютерами и наглядными пособиями, материалами справочного характера, нормативными документами и различными образцами, которые разрешены к использованию на экзамене квалификационном и указаны в билете в разделе инструкция.

Результаты экзамена квалификационного определяются на основании оценочной ведомости и/или результатов решения профессиональных задач оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно», вносятся в итоговую ведомость экзамена квалификационного аттестационной комиссии и объявляются в тот же день.

Решение аттестационной комиссии об окончательной оценке студента по экзамену квалификационному принимается на закрытом заседании простым большинством голосов членов аттестационной комиссии, участвующих в заседании. При равном числе голосов голос председателя является решающим.

Время проведения экзамена – 6 часов.

## Критерии оценки экзамена квалификационного

### Общая характеристика критериев оценивания

Предметом оценивания в процессе экзамена служит профессиональная компетенция, овладение которой предполагает освоенные знания, умения и приобретенный практический опыт в ходе изучения 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 ПМ.01 Разработка кода для обучения искусственного интеллекта. Сопутствующими компетенциями при формировании профессиональных компетенций служат общие компетенции.

### Критерии оценки

Оценка	Критерии
5 «отлично»	получается студент, который набрал в сумме 21-25 баллов (учитывая пороговое значение по весу критерия в соответствии с ПК)
4 «хорошо»	получает студент, который набрал в сумме 16-20 баллов (учитывая пороговое значение по весу критерия в соответствии с ПК),
3 «удовлетворительно»	получает студент, который набрал в сумме 11-15 баллов (учитывая пороговое значение по весу критерия в соответствии с ПК)
2 «неудовлетворительно»	получает студент, который набрал в сумме менее 11 баллов (учитывая пороговое значение по весу критерия в соответствии с ПК).

Экзамен по профессиональному модулю проводится в устной форме по билетам. Билет содержит практические задания для проверки освоенных профессиональных компетенций (ПК):

Код	Наименование видов деятельности и профессиональных компетенций
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 программных средств.
ПК 1.6	Выполнять тестирование программного кода.
ПК 1.7	Составлять тестовые сценарии.

**Билет 1.**

Оборудование: ПК, TensorFlow, Python.

**Прочитайте текст. Выполните задание.**

Составьте алгоритм вычисления суммы всех чисел в массиве и опишите последовательность его выполнения.

**Билет 2.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Разработайте алгоритм поиска кратчайшего пути в графе (например, Дейкстры):

- Опишите работу алгоритма на псевдокоде или в виде блок-схемы;
- Реализуйте этот алгоритм на Python или другом выбранном языке;
- Протестируйте на простом примере (граф с 5-7 вершинами);
- Опишите, как можно ускорить работу алгоритма для больших графов.

**Билет 3.**

Оборудование: ПК, TensorFlow, Python.

**Прочитайте текст. Выполните задание.**

Выполните обработку массива случайных чисел с помощью библиотеки NumPy:

- Создайте массив из 1000 случайных чисел;
- Отсортируйте массив;
- Найдите и выведите среднее значение;
- Сравните скорость обработки с обычными списками Python (сделайте вывод).

**Билет 4.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Постройте модель «дерево решений» для задачи классификации:

- Используйте scikit-learn и набор данных Iris (или аналогичный);
- Обучите модель, разделив данные на обучающую и тестовую выборки;
- Оцените точность полученной модели;
- Выведите важность признаков.

**Билет 5.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Создайте модуль поиска дубликатов в большом CSV-файле:

- Импортируйте и обработайте CSV-файл с помощью Pandas;
- Найдите дубликаты по заданному столбцу и удалите их;
- Оцените производительность (например, время выполнения при разном размере файла);
- Кратко опишите, как защищаете данные при обработке.

**Билет 6.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Интегрируйте свой ИИ-модуль в уже существующую программу:

- Опишите, как ваш модуль подключается к основной системе;
- Реализуйте простую интеграцию с помощью функции/класса (пример кода);
- Опишите возможные сложности при интеграции и их решения.

**Билет 7.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Настройте взаимодействие двух модулей через API:

- Опишите, как будет происходить обмен данными (REST, gRPC и т.п.);
- Покажите пример запроса от одного модуля к другому;
- Предложите шаги по уменьшению задержек между вызовами (поясните выбранный способ оптимизации).

**Билет 8.**

Оборудование: ПК, TensorFlow, Python.

**Прочитайте текст. Выполните задание.**

Оформите программный модуль по стандарту:

- Добавьте Docstring с описанием назначения модуля и его параметров;
- Структурируйте код на функции и классы (если необходимо);
- Используйте комментарии для пояснения сложных участков кода.

**Билет 9.**

Оборудование: ПК, TensorFlow, Python.

**Прочитайте текст. Выполните задание.**

Выполните статический анализ программного кода:

- Установите и запустите pylint или flake8 для вашего Python-кода;
- Исправьте не менее 5 замечаний, обнаруженных анализатором;
- Приложите исходный и исправленный варианты кода.

**Билет 10.**

Оборудование: ПК, Git, PDF.

**Прочитайте текст. Выполните задание.**

Сгенерируйте документацию для проекта и организуйте командную работу:

- Оформите документацию с помощью Sphinx или Doxygen (по вашему выбору);
- Разместите проект и документацию на GitHub/GitLab;
- Сделайте скриншот или описание, что каждый участник видит актуальную документацию.

**Билет 11.**

Оборудование: ПК, Git, PDF.

**Прочитайте текст. Выполните задание.**

Смоделируйте и разрешите конфликт при слиянии веток:

- Создайте две ветки в Git, внесите конфликтные изменения;
- Попробуйте их слить (merge);
- Опишите и покажите шаги по разрешению конфликта;
- Оформите pull request на ревью.

**Билет 12.**

Оборудование: ПК, Git, CI/CD, PDF.

**Прочитайте текст. Выполните задание.**

Настройте простой процесс CI/CD:

- Включите автоматический запуск тестов в GitHub Actions или GitLab CI (желательно на Python);
- Опишите структуру yaml-файла для пайплайна;
- Поясните, что происходит на каждом этапе CI/CD (build/test/deploy).

**Билет 13.**

Оборудование: ПК, PyCharm.

**Прочитайте текст. Выполните задание.**

Отладьте программный модуль:

- Откройте ваш код в IDE с поддержкой отладки (например, PyCharm);
- Поставьте точки останова и пройдите по коду пошагово;
- Зафиксируйте, как изменяются значения переменных на каждом шаге.

**Билет 14.**

Оборудование: ПК, PyCharm, README.

**Прочитайте текст. Выполните задание.**

Добавьте логирование и профилирование:

- Внедрите журналирование событий с помощью logging;
- Измерьте время выполнения выбранной функции через cProfile или аналог;
- Приложите лог-файл и отчёт профайлера.

**Билет 15.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Найдите и исправьте ошибку в многопоточной программе:

- Напишите небольшой пример программы с потоками (threading/multiprocessing);
- Используйте отладчик для поиска гонки данных или deadlock'a;
- Опишите найденную ошибку и способ её устранения.

**Билет 16.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Создайте юнит-тесты:

- Напишите минимум 5 тестов на отдельные функции модуля;
- Используйте PyTest или unittest;
- Приложите отчёт о прохождении тестов (скриншот/вывод консоли).

**Билет 17.**

Оборудование: ПК, CI/CD, PDF.

**Прочитайте текст. Выполните задание.**

Реализуйте и автоматизируйте интеграционные тесты:

- Опишите логическую цепочку модулей для проверки (например, обработка, анализ, вывод);
- Автоматизируйте запуск тестирования всей цепочки;
- Приложите вывод с результатами.

**Билет 18.**

Оборудование: ПК, CI/CD, PDF.

**Прочитайте текст. Выполните задание.**

Включите проверку кода в CI/CD пайплайн:

- Настройте автоматический запуск всех тестов после каждого коммита;
- Покажите логи успешного и неуспешного запуска;
- Объясните, в чём польза такой проверки.

**Билет 19.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Разработайте тестовые сценарии (включая крайние и ошибочные случаи):

- Оформите не менее 5 сценариев для вашей функции/модуля;
- Для каждого: опишите вход, ожидаемое поведение, результат;
- Объясните, зачем нужны тесты на ошибочные входные данные.

**Билет 20.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Заполните шаблон тест-кейса:

- Используйте готовую форму (например, название, цель, шаги, входные данные, ожидаемый результат, критерий успешности);
- Примените к проверке поиска элемента в списке;
- Заполните не менее 2 кейсов.

**Билет 21.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Автоматизируйте создание и выполнение тестовых сценариев:

- Используйте инструмент (например, pytest-bdd или аналог);
- Опишите процесс запуска тестов по изменению кода;
- Приложите журнал с результатами тестирования.

**Билет 22.**

Оборудование: ПК, Jupyter Notebook, Pandas, Python.

**Прочитайте текст. Выполните задание.**

Обработка и очистка набора данных:

- Считайте CSV-файл в Pandas;
- Проведите очистку данных (удалите пустые строки, приведите форматы, удалите дубликаты);
- Сохраните и приложите итоговый файл;
- Поясните, почему этапы очистки важны перед обучением ИИ-моделей.

**Билет 23.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Реализуйте связанный список и используйте его в задаче:

- Опишите структуру классами Python;
- Реализуйте функции добавления, удаления элементов;
- Примените для хранения элементов и продемонстрируйте работу с несколькими элементами;
- Кратко сравните с использованием стандартных списков.

**Билет 24.**

Оборудование: ПК, GitHub, README.

**Прочитайте текст. Выполните задание.**

Создайте README для репозитория:

- Подготовьте файл с кратким описанием проекта, структуры, инструкции по запуску;
- Загрузите на GitHub или GitLab;
- Проверьте отображение README в репозитории.

**Билет 25.**

Оборудование: ПК, Jupyter Notebook, Python.

**Прочитайте текст. Выполните задание.**

Рефакторинг кода:

- Возьмите пример кода с нарушением стиля (даётся преподавателем или берёте свой);
- Приведите к единому стилю (PEP8), улучшите читаемость;
- Прокументируйте внесённые изменения (прямо в коде или отдельным файлом).

### Критерии оценивания экзаменационного задания.

Профессиональная компетенция	Основные показатели оценки результатов	Балл	Пороговый балл*
ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.	ОПОР 1 Анализ и декомпозиция технического задания	1	1
	ОПОР 2 Проектирование алгоритмической логики модуля	1	
	ОПОР 3 Реализация алгоритма в виде программного кода	1	
	ОПОР 4 Верификация и тестирование соответствия результатов ТЗ	1	
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием	ОПОР 1. Обоснование выбора языка программирования	1	2
	ОПОР 2. Грамотное применение основных принципов технологии структурного и объектно-ориентированного программирования	1	
	ОПОР 3. Разработка программы по разработанному алгоритму как отдельного модуля	1	
	ОПОР 4. Оформление документации на модуль в соответствии со стандартами	1	
ПК 1.3. Оформлять программный код в соответствии с техническим заданием.	ОПОР 5. Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения	1	2
	ОПОР 6. Разработка тестовых наборов данных	1	
	ОПОР 7. Проверка работоспособности программного обеспечения	1	
	ОПОР 8. Рефакторинг и оптимизация программного кода	1	
	ОПОР 9. Исправление зафиксированных дефектов	1	
ПК 1.4. Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.	ОПОР 10. Формализация и алгоритмизация поставленных задач	1	2
	ОПОР 11. Написание программного кода с использованием языков программирования, определения и манипулирования данными	1	
	ОПОР 12. Оформление программного кода в соответствии с установленными требованиями	1	
	ОПОР 13. Работа с системой контроля версий	1	
	ОПОР 14. Проверка и отладка программного кода	1	
ПК 1.5. Выполнять отладку программных модулей с использованием специализированных программных средств.	ОПОР 15. Локализация дефекта (поиск источника ошибки)	1	1
	ОПОР 16. Диагностика причины дефекта	1	
	ОПОР 17. Разработка и применение исправления (устранение дефекта)	1	
	ОПОР 18. Верификация исправления и регрессионное тестирование	1	
ПК 1.6. Выполнять тестирование программного кода.	ОПОР 19. Планирование и проектирование тестов	1	2
	ОПОР 20. Реализация (написание) тестового кода	1	
	ОПОР 21. Выполнение тестов и анализ результатов	1	
	ОПОР 22. Поддержание и оптимизация тестовой батареи	1	
ПК 1.7. Составлять тестовые сценарии.	ОПОР 23. Анализ требований и проектирование структуры сценариев	1	1
	ОПОР 24. Формализация сценариев по стандартным шаблонам	1	
	ОПОР 25. Валидация и оптимизация набора сценариев	1	

\* Компетенция считается освоенной