Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гриб Владислав Валерьевич
Должность: Ректор
Дата подписания: 06.11.2026 17:39:38
Уникальный программный ключ:
637517d24e103c3db032acf37e83...

Образовательное частное учреждение высшего образования
«МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С. ГРИБОЕДОВА»
(ИМПЭ им. А.С. Грибоедова)
ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
(МЕЖДУНАРОДНЫЙ ДИПЛОМАТИЧЕСКИЙ КОЛЛЕДЖ)

УТВЕРЖДАЮ
Директор Международного
дипломатического колледжа
_____ Н.И. Платонова
«16» февраля 2026г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

Информационные технологии

Специальность
09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Квалификация выпускника:
специалист по работе с искусственным интеллектом

на базе основного общего образования

Форма обучения: очная

Москва

Фонд оценочных средств для дисциплины «Информационные технологии». Специальность 09.02.13 Интеграция решений с применением технологий искусственного интеллекта квалификация: специалист по работе с искусственным интеллектом /– М.: ИМПЭ им. А. С. Грибоедова.

Фонд оценочных средств является неотъемлемой частью рабочей программы дисциплины.

Организация-разработчик:	Образовательное частное учреждение высшего образования «Московский университет имени А. С. Грибоедова»

Директор Международного
дипломатического колледжа_____/Н.И. Платонова

1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

Настоящий Фонд оценочных средств (ФОС) по дисциплине «Информационные технологии» является неотъемлемым приложением к рабочей программе дисциплины (РПД) «Информационные технологии». На данный ФОС распространяются все реквизиты утверждения, представленные в РПД по данной дисциплине.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

Для определения качества освоения обучающимися учебного материала по дисциплине используются следующие оценочные средства:

№ п/п	Оценочное средство	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Тестирование	Вид контроля, позволяющий оценить изученный теоретический материал.	Вопросы для проведения тестирования
2	Практические задания	Вид контроля, позволяющий оценить умение обучающегося применять осваиваемую компетенцию в практических ситуациях и при решении производственных задач	Задания к практическому занятию
3	Контрольная работа	Вид контроля, позволяющий определить результат освоения компетенций по дисциплине в рамках рассматриваемой темы, оцениваемый с помощью соответствующих индикаторов достижения компетенций	Задания контрольной работы
4	Самостоятельная работа	Вид контроля, позволяющий оценить проработку теоретического материала, изучение рекомендуемой литературы, выполнение практико-ориентированных заданий (заполнение таблиц, проведение сравнительного анализа, составление схем и др.), решение практических задач, создание презентаций, написание рефератов, подборку нормативного и иного материала и выполнение других заданий	Задания самостоятельной работы
5.	Зачет/Зачет с оценкой/Экзамен	Вид контроля, позволяющий выявить степень овладения знаниями, умениями и навыками, необходимыми для дальнейшего освоения образовательной программы подготовки	Вопросы для подготовки к зачету/зачету с оценкой/экзамену

3.ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Темы для подготовки к промежуточной аттестации

1. Роль современных информационных технологий в жизненном цикле проекта по интеграции ИИ-решений.
2. Архитектура вычислительных систем: сравнительный анализ конфигураций для задач обучения и инференса моделей ИИ.
3. Классификация программного обеспечения. Роль системного и прикладного ПО в работе инженера по ИИ.
4. Принципы работы операционных систем семейств Windows и Linux. Критерии выбора ОС для серверной части ИИ-проекта.
5. Концепции виртуализации и контейнеризации (на примере Docker). Их преимущества для воспроизводимости экспериментов и развертывания ИИмоделей.
6. Современные средства поиска профессиональной информации. Стратегии эффективного поиска на платформах arXiv, GitHub, Stack Overflow.
7. Критическое мышление при работе с информацией: методы оценки достоверности источника в цифровой среде.
8. Основные угрозы информационной безопасности в ИТ-проектах. Понятия атаки на конфиденциальность, целостность и доступность (CIA-триада).
9. Российское законодательство в сфере информационных технологий: обзор ключевых законов (152-ФЗ, 149-ФЗ) и их значение для разработчика.
10. Правовые и этические аспекты использования открытого ПО и открытых моделей искусственного интеллекта. Типы лицензий (MIT, GPL, Apache 2.0).
11. Системы контроля версий: назначение, принципы работы. Роль Git в современной разработке.
12. Основные понятия Git: репозиторий, коммит, ветка (branch), слияние (merge). Жизненный цикл коммита.
13. Модели ветвления в Git: сравнительный анализ Git Flow и GitHub Flow. Преимущества и недостатки.
14. Платформы коллективной разработки GitHub и GitLab: социальные функции (Fork, Pull/Merge Request, Code Review).
15. Принципы эффективной командной работы в IT-проекте. Роль коммуникации и инструментов планирования.
16. Язык разметки Markdown: синтаксис и применение для создания технической документации (README.md, wiki).
17. Инструменты управления проектами и задачами: Trello, Jira, доски в GitHub Projects. Принципы декомпозиции задач.
18. Методологии Agile, Scrum, Kanban: базовые принципы и их отражение в инструментах управления.
19. Назначение и классификация тестирования программного обеспечения (модульное, интеграционное, системное).
20. Понятие тест-кейса, чек-листа, тестового сценария. Принципы составления тестовой документации.

21. Интегрированные среды разработки (IDE) для Python: сравнительный обзор возможностей VS Code, PyCharm, Jupyter Notebook.
22. Назначение и принципы работы linterов и formatterов (на примере Pylint и Black) для обеспечения качества кода.
23. Стандарты оформления кода (PEP 8 для Python, Google Style Guide). Значение читаемости и поддерживаемости кода.
24. Принципы документирования программного кода. Виды комментариев, документирование функций и модулей (docstrings).
25. Непрерывная интеграция и непрерывная поставка (CI/CD): базовые понятия, преимущества для процесса разработки.
26. Принципы работы с удаленным репозиторием Git: clone, pull, push, fetch. Работа с удаленными ветками.
27. Разрешение конфликтов слияния (merge conflicts) в Git: причины возникновения и стратегии решения.
28. Протоколы передачи данных в сетях: сравнительная характеристика TCP и UDP. Области применения.
29. Модель взаимодействия клиент-сервер. Протокол HTTP/HTTPS: методы запросов, коды состояния.
30. Принципы работы REST API. Использование API для взаимодействия с внешними сервисами и моделями ИИ.
31. Базовые сетевые концепции: модель OSI/ TCP-IP, IP-адресация, маска подсети, шлюз.
32. Облачные вычисления: модели обслуживания (IaaS, PaaS, SaaS, MLaaS) и их различия.
33. Обзор публичных облачных платформ (Yandex Cloud, SberCloud, AWS, Azure) для задач машинного обучения.
34. Критерии выбора облачного провайдера для проекта по интеграции ИИ: стоимость, доступность сервисов, региональное присутствие.
35. Контейнеризация приложений: образ (image) и контейнер (container). Преимущества Docker для упаковки приложений.
36. Структура и назначение Dockerfile. Основные инструкции (FROM, RUN, COPY, CMD, EXPOSE).
37. Репозитории готовых моделей искусственного интеллекта: обзор платформ Hugging Face, Kaggle Models, TensorFlow Hub.
38. Критерии выбора готовой модели ИИ для интеграции: точность (accuracy), быстродействие (latency), размер модели, лицензия.
39. Форматы файлов моделей машинного обучения (ONNX, PyTorch .pt, TensorFlow SavedModel). Их совместимость и преобразование.
40. Понятие MLOps (Machine Learning Operations) и его связь с изученными инструментами (Git, Docker, CI/CD, облака).

Темы для самостоятельной работы обучающихся

1. Анализ жизненного цикла данных в ИИ-проекте.
2. Сравнительный анализ правовых режимов для ПО и данных.
3. Исследование рынка отечественного офисного ПО.

4. Обзор угроз ИБ для систем с использованием ИИ.
5. Карта сетевых протоколов, используемых в микросервисной архитектуре. 6. Настройка виртуального окружения и менеджер пакетов. С
7. Создание комплексного .gitignore файла.
8. Сравнение способов установки и настройки Docker.
9. Автоматизация форматирования кода с помощью pre-commit hooks.
10. Работа с API на примере облачного провайдера.
11. Разработка Code Review Checklist. Создать детальный чек-лист для проведения code review в учебной команде.
12. Моделирование рабочего процесса по методологии Kanban.
13. Создание шаблона для Issue и Pull Request (PR) на GitHub.
14. Проведение ретроспективы учебного спринта.
15. Разработка политики именования веток в Git.
16. Написание модульных тестов с использованием unittest/pytest.
17. Анализ технического долга (technical debt).
18. Создание конфигурационного файла для линтера (.pylintrc).
19. Документирование API с помощью OpenAPI/Swagger.
20. Интеграция статического анализа кода в CI/CD.
21. Выбор модели ИИ по заданным ограничениям.
22. Создание Docker-образа для инференса модели.
23. Расчет стоимости облачного развертывания.
24. Изучение формата ONNX Runtime.
25. Работа с облачными сервисами для данных.
26. Разработка технического задания (ТЗ) для учебного ИИ-проекта.
27. Проведение A/B тестирования инструментов.
28. Создание «Справочника молодого инженера ИИ».
29. Аудит безопасности личного ИТ-стэка.

Примерные варианты тестовых оценочных заданий (ТОЗ) для контрольного рубежа в рамках текущего контроля

Тестовое задание	Оцениваемые компетенции
ВОПРОСЫ ОТКРЫТОГО ТИПА	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программа или набор программ, управляющих аппаратным обеспечением компьютера и предоставляющих сервисы для прикладного ПО. Это _____. +: операционная система	ОК 01.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Технология, позволяющая запускать несколько изолированных экземпляров операционной системы на одном физическом сервере. Это _____.	ОК 01.

+: виртуализация	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Программная технология упаковки приложения со всеми его зависимостями в стандартизированную единицу для развертывания. Это _____. +: контейнеризация	ОК 01.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс предложения изменений в проекте на GitHub, когда разработчик уведомляет других о готовности к слиянию своей ветки. Это создание _____. +: pull request (или merge request)	ПК 1.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Набор правил, определяющий формат и способ обмена данными между устройствами в сети. Это сетевой _____.. +: протокол	ОК 02.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Среда, предназначенная для разработки, отладки и тестирования программного кода. Аббревиатура: _____ +: IDE (Integrated Development Environment)	ПК 1.3.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Практика автоматической сборки и тестирования каждого изменения в коде. Аббревиатура: _____. +: CI (Continuous Integration)	ПК 1.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Крупнейшая открытая платформа для размещения и совместной работы над моделями, датасетами и приложениями машинного обучения. Это _____. +: Hugging Face.	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Основной стандарт оформления кода на языке Python. Это _____. +: PEP-8	ПК 1.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Показатель, который часто используется для оценки точности модели классификации... Это _____. +: точность	ПК 3.1.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Язык разметки для создания форматированного текста с помощью простого синтаксиса, часто используемый в файлах README. Это _____. +: Markdown	ОК 05.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Система управления контейнерами с открытым исходным кодом, ставшая отраслевым стандартом. Это _____. +: Docker	ОК 01.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Подход к разработке ПО, сочетающий практики разработки (Dev) и эксплуатации (Ops) для ускорения жизненного цикла. Это _____. +: DevOps	ПК 1.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс применения обученной модели ИИ к новым данным для получения предсказаний. Это _____. +: инференс	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Предварительно обученная на больших данных модель, которую можно дообучить или использовать напрямую для решения конкретной задачи. Это _____ модель. +: готовая(предобученная)	ПК 3.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Процесс поиска и исправления ошибок в программном коде. Это _____. +: откладка	ПК 1.1.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Инструмент для автоматического статического анализа кода на предмет соответствия стилю и выявления потенциальных ошибок. Это _____. +: линтер	ПК 1.3.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ)	ОК 01.

Модель облачных услуг, при которой потребителю предоставляется возможность использования инфраструктуры (виртуальные серверы, сети). Это _____ (IaaS). +: Инфраструктура как услуга	
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Конфликт, который возникает, когда Git не может автоматически объединить изменения из разных веток. Это _____ конфликт. +: merge (конфликт слияния)	ПК 1.4.

(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Распределенная система контроля версий, созданная Линусом Торвальдсом. Это _____. +: Git	ПК 1.4.
(ВПИСАТЬ ПРАВИЛЬНЫЙ ОТВЕТ) Методология гибкой разработки, в которой работа ведется короткими итерациями, называемыми спринтами. Это _____. +: Scrum	ОК 04.
ВОПРОСЫ ЗАКРЫТОГО ТИПА	
Какова основная функция операционной системы? -: Обеспечивать доступ к сети Интернет. +: Управлять аппаратными ресурсами компьютера и предоставлять среду для выполнения прикладных программ. -: Создавать презентации и текстовые документы. -: Защищать компьютер от вирусов.	ОК 01.
Какая из перечисленных лицензий на ПО является лицензией с «сильным копилефтом», требующей, чтобы все производные работы распространялись на тех же условиях? -: Лицензия MIT -: Лицензия Apache 2.0 +: Лицензия GNU GPL (General Public License) -: Проприетарная лицензия (EULA)	ОК 06.
Какая команда в Git используется для создания новой ветки (branch) и одновременного перехода в нее? -: git branch new-feature +: git checkout -b new-feature -: git merge new-feature -: git commit new-feature	ПК 1.4.
Какой протокол из перечисленных является протоколом транспортного уровня модели TCP/IP и гарантирует доставку данных? -: HTTP -: IP +: TCP -: FTP	ОК 02.
Модель облачных услуг «Инфраструктура как услуга» (IaaS) предполагает, что провайдер предоставляет потребителю: -: Готовое прикладное программное обеспечение (например, почтовый клиент). -: Платформу для разработки, развертывания и управления приложениями. +: Фундаментальные вычислительные ресурсы (виртуальные серверы, сетевое оборудование, хранилища). -: Консультационные услуги по миграции в облако.	ОК 01.

Какой инструмент используется для автоматического приведения кода Python к единому стилю согласно PEP 8? -: Интерпретатор Python +: Форматтер (например, black или autopep8) -: Компилятор C++ -: Система управления пакетами pip	ПК 1.3.
Выберите МЕРУ, которая НЕ является базовым принципом (элементом) триады CIA в информационной безопасности: -: Конфиденциальность (Confidentiality) +: Иммобильность (Immobility) -: Целостность (Integrity) -: Доступность (Availability)	ОК 06.
На какой из перечисленных платформ можно найти, оценить и скачать тысячи предобученных моделей искусственного интеллекта с открытым исходным кодом? -: Docker Hub +: Hugging Face -: GitHub Gist -: Google Диск	ПК 3.1.
Какой элемент методологии Scrum НЕ является правильным? -: Работа делится на короткие итерации, называемые спринтами. -: Основные роли: Владелец продукта, Scrum-мастер, Команда разработки. -: Ежедневные митинги (stand-up) для синхронизации команды. +: Подробное техническое задание (ТЗ) должно быть полностью готово и неизменно перед началом первого спринта.	ОК 04.

Шкала оценивания тестовых заданий

Данный вид контроля, рассчитан на выявление уровня усвоения теоретического и практического материала в рамках изучения дисциплин. Критерии оценивания теста: 1 вопрос = 1 балл.

Оценка «5»	30-27 Баллов
Оценка «4»	26-23 Баллов
Оценка «3»	22-18 Баллов
Оценка «2»	0-17 Баллов