ПРИЛОЖЕНИЕ

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Колледж Рязанского государственного радиотехнического университета
имени В.Ф. Уткина

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

МДК.01.02 РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ С ПОДДЕРЖКОЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Специальность	09.02.13 Интеграция решений с применением технологий искусственного интеллекта
Квалификация выпускника	Специалист по работе с искусственным интеллектом
Форма обучения	очная
Год набора	2027 г.

Рязань 2026

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные средства – это совокупность контрольно-оценочных материалов (контрольных заданий, описаний форм и процедур проверки), предназначенных для проведения промежуточной аттестации для объективной оценки качества освоения обучающимися содержания МДК.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения МДК, целям и требованиям образовательной программы в ходе проведения промежуточной аттестации.

Промежуточная аттестация по МДК предусматривает экзамен. Форма проведения экзамена – решение задач, теоретические вопросы.

2 ПАСПОРТ ОЦЕНОЧНЫХ СРЕДСТВ ПО МДК

Наименование разделов и тем	Формируемые компетенции	Методы оценки
Тема 2.1. Платформы и инструменты мобильной разработки	ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4	Экзамен
Тема 2.2. Интеграция ИИ в мобильные приложения	ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4	Экзамен
Тема 2.3. Разработка интерактивных мобильных ИИ-приложений	ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4	Экзамен
Тема 2.4. Развертывание мобильных приложений с ИИ	ПК .1, ПК 1.2, ПК 1.3, ПК 1.4	Экзамен

3 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Код ПК, ОК	Критерии оценки результата (показатели освоенности компетенций)	Формы контроля и методы оценки
ПК.1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	**Оценка "отлично":** алгоритмы разработаны в полном соответствии с ТЗ, оптимизированы и понятны. **Оценка "хорошо":** алгоритмы разработаны в соответствии с ТЗ, но допускают незначительные отклонения. **Оценка "удовлетворительно":** алгоритмы разработаны с частичным соответствием ТЗ.	Экзамен
ПК.1.2 Разрабатывать программные модули в соответствии с техническим заданием	**Оценка "отлично":** программные модули разработаны в полном соответствии с ТЗ, тесты проходят успешно. **Оценка "хорошо":** программные модули разработаны с минимальными несоответствиями, тесты в целом успешны. **Оценка "удовлетворительно":** программные модули разработаны с существенными доработками.	Экзамен
ПК.1.3 Оформлять программный код в соответствии с	**Оценка "отлично":** код полностью оформлен в соответствии с требованиями, включая комментарии и стиль кода.	Экзамен

Код ПК, ОК	Критерии оценки результата (показатели освоенности компетенций)	Формы контроля и методы оценки
техническим заданием	**Оценка "хорошо":** код оформлен в соответствии с требованиями, допускаются мелкие недочёты. **Оценка "удовлетворительно":** код оформлен частично в соответствии с требованиями.	
ПК.1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.	**Оценка "отлично":** система контроля версий используется эффективно, изменения фиксируются корректно. **Оценка "хорошо":** система контроля версий используется, но имеются мелкие нарушения порядка фиксации изменений. **Оценка "удовлетворительно":** система контроля версий используется частично или с ошибками.	Экзамен

4 ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.1 Экзамен (ПК.1.1 – ПК.1.4)

Перечень вопросов

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

1. Основные этапы жизненного цикла мобильного приложения
2. Отличия нативной, гибридной и кроссплатформенной разработки
3. Архитектурные паттерны мобильных приложений (MVC, MVP, MVVM)
4. Понятие Activity и Fragment в Android, их жизненный цикл
5. Понятие ViewController в iOS и управление его состоянием
6. Управление памятью в мобильных ОС (сборка мусора, ARC в iOS)
7. Понятие контекста (Context) в Android и его использование
8. Основы работы с локальными базами данных (SQLite, Room, CoreData)
9. Сетевые запросы в мобильных приложениях (Retrofit, Alamofire)
10. Асинхронное программирование в мобильной разработке (Coroutines, Async/Await)
11. Понятие интентов (Intents) в Android и их роль в навигации
12. Понятие фрагментации Android-устройств и её влияние на разработку
13. Работа с фоновыми задачами (WorkManager, Background Tasks)
14. Понятие Edge AI и его преимущества для мобильных приложений
15. Отличия облачного инференса от on-device (на устройстве)
16. Основные фреймворки для запуска ИИ на мобильных устройствах
17. Архитектура TensorFlow Lite и его место в экосистеме TensorFlow
18. Понятие интерпретатора (Interpreter) в TensorFlow Lite
19. Формат моделей .tflite и его особенности
20. Понятие делегатов (Delegates) в TensorFlow Lite
21. Использование GPU-делегата для ускорения инференса
22. Использование NPU и аппаратных ускорителей в мобильных ИИ

23. Понятие квантования моделей и его влияние на размер и скорость
24. Post-training quantization: виды и применение
25. Основы работы с TensorFlow Lite Task Library
26. Понятие on-device обучения (Federated Learning) и его применение
27. Ограничения мобильных устройств при работе с ИИ-моделями
28. Кэширование результатов инференса для повышения производительности
29. Обновление ИИ-моделей в мобильном приложении без обновления самого приложения
30. Архитектура MobileNet и её применение в мобильных приложениях
31. Архитектура EfficientNet и её оптимизация для мобильных устройств
32. Архитектура SSD MobileNet для мобильной детекции
33. Архитектура YOLO и её мобильные версии (YOLOv8n, YOLOv11n)
34. Архитектура MobileBERT и её применение в мобильных приложениях
35. Архитектура ALBERT и её оптимизация для мобильных устройств
36. Токенизация текста с помощью Wordpiece и Sentencepiece
37. Голосовые ассистенты на мобильных устройствах: архитектура
38. Распознавание речи (ASR) на устройстве: основные подходы
39. Архитектура YAMNet и её применение в мобильных приложениях
40. Уменьшение размера модели: pruning (обрезка)
41. Уменьшение размера модели: дистилляция знаний
42. Уменьшение размера модели: квантование до 8-бит
43. Сравнение float32, float16 и int8 квантования для мобильных устройств
44. Использование NNAPI для ускорения на Android
45. Использование Core ML для ускорения на iOS
46. Понятие thread management при инференсе на мобильном устройстве
47. Влияние размера входного изображения на скорость инференса
48. Пакетная обработка (batching) на мобильных устройствах: применимость
49. Оптимизация препроцессинга данных для мобильных устройств

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием

50. Работа с разрешениями (Permissions) в Android и iOS
51. Работа с сенсорами и камерой мобильного устройства
52. Основы Material Design и Human Interface Guidelines
53. Управление зависимостями в мобильных проектах (Gradle, CocoaPods)
54. Основы отладки мобильных приложений (Logcat, Xcode Debugger)
55. Основные этапы интеграции TFLite модели в Android-приложение
56. Основные этапы интеграции TFLite модели в iOS-приложение
57. Преимущества и ограничения Task Library для разработчиков без глубоких знаний ML
58. Работа с изображениями в мобильных ИИ-приложениях: препроцессинг
59. Работа с текстовыми данными: токенизация на устройстве
60. Работа с аудиоданными в реальном времени на мобильных устройствах
61. Задача классификации изображений на мобильном устройстве
62. Задача детекции объектов (Object Detection) в реальном времени
63. Задача сегментации изображений на мобильных устройствах
64. Распознавание лиц на мобильном устройстве: основные алгоритмы
65. Распознавание жестов рук на основе камеры смартфона
66. Распознавание позы человека (Pose Estimation) в мобильных приложениях

67. Распознавание рукописных цифр на мобильном устройстве
68. Оптическое распознавание символов (OCR) на мобильных устройствах
69. Интеграция OCR с Text-to-Speech для приложений помощи слабовидящим
70. Распознавание объектов в дополненной реальности (AR)
71. Стилизация изображений (Style Transfer) на мобильных устройствах
72. Повышение разрешения изображений (Super Resolution) на устройстве
73. Классификация продуктов по фотографии в мобильном приложении
74. Определение дефектов на производстве с помощью мобильной камеры
75. Мобильные приложения для распознавания растений и животных
76. Распознавание дорожных знаков для автомобильных приложений
77. Мобильная детекция текста на изображениях (Text Detection)
78. Работа с видеопотоком в реальном времени для задач CV
79. Оптимизация FPS при обработке видео на мобильных устройствах
80. Использование камеры и фонарика в связке с ИИ-модулем
81. Калибровка камеры для задач компьютерного зрения
82. Обработка изображений с низкой освещённостью
83. Понятие data augmentation при подготовке мобильных CV-моделей
84. Задача классификации текста на мобильном устройстве
85. Анализ тональности текста (Sentiment Analysis) на устройстве
86. Задача Question Answering с использованием BERT на мобильном устройстве
87. Интеграция BertQuestionAnswerer в Android-приложение
88. Интеграция BertQuestionAnswerer в iOS-приложение
89. Понятие контекста в Question Answering и его ограничения на мобильных
90. Задача Named Entity Recognition (NER) на мобильном устройстве
91. Мобильные приложения для перевода текста в офлайн-режиме
92. Чат-боты на основе локальных моделей в мобильных приложениях
93. Синтез речи (TTS) на мобильных устройствах
94. Интеграция ИИ-модуля для обработки пользовательских отзывов
95. Автоматическое заполнение форм на основе NLP
96. Умный поиск по документам на мобильном устройстве
97. Классификация спама в SMS на устройстве
98. Определение языка текста на мобильном устройстве
99. Извлечение ключевых слов из текста на устройстве
100. Построение диалоговых систем с ограниченным словарём
101. Проблема галлюцинаций в мобильных LLM и способы её смягчения
102. Классификация аудиособытий на мобильном устройстве
103. Распознавание музыки и определение жанра на устройстве
104. Детекция шума и его фильтрация в реальном времени
105. Распознавание голосовых команд на мобильном устройстве
106. Интеграция ИИ для анализа сенсорных данных (акселерометр, гироскоп)
107. Распознавание активности пользователя (ходьба, бег, езда)
108. Использование ИИ для анализа данных о здоровье с носимых устройств
109. Мобильные приложения для анализа почерка
110. Определение эмоций по голосу на мобильном устройстве
111. Анализ текстовых сообщений в мессенджерах с помощью on-device ИИ
112. Интеграция ИИ в мобильные игры (процедурная генерация)
113. Распознавание жестов на основе данных тачскрина
114. Использование ИИ для адаптации интерфейса под пользователя
115. Предиктивный ввод текста на мобильном устройстве

116. Работа с памятью при загрузке больших моделей
117. Ленивая загрузка моделей и освобождение ресурсов
118. Обработка ошибок инференса и graceful degradation

ПК 1.3. Оформлять программный код в соответствии с техническим заданием

119. Публикация приложений в Google Play и App Store: основные требования
120. Безопасность ИИ-моделей на мобильных устройствах
121. Понятие Model Interpretability в контексте мобильных приложений
122. Мониторинг производительности ИИ-модуля в production
123. Оценка качества CV-модели на мобильных устройствах
124. Проблема переобучения в контексте мобильных CV-приложений
125. Основные ошибки при интеграции CV-моделей в мобильные приложения
126. Оптимизация NLP-моделей для мобильных устройств
127. Оценка качества NLP-модели в мобильном приложении
128. Метрики производительности ИИ-модели на мобильном устройстве (Latency, FPS, Memory)
129. Понятие latency и его влияние на пользовательский опыт
130. Профилирование ИИ-модуля на Android (Android Profiler)
131. Профилирование ИИ-модуля на iOS (Instruments)
132. Тестирование производительности ИИ-модуля на разных устройствах
133. Бенчмаркинг моделей с помощью TensorFlow Lite Benchmark Tool
134. Энергопотребление ИИ-модуля и его оптимизация
135. Виды тестирования мобильных ИИ-приложений
136. Юнит-тестирование ИИ-модуля на мобильном устройстве
137. Интеграционное тестирование связки UI и ИИ-модуля
138. Тестирование на различных версиях ОС и моделях устройств
139. Отладка ИИ-модуля в эмуляторе: ограничения и возможности
140. Логирование работы ИИ-модуля в production
141. Сбор телеметрии и метрик использования ИИ-функций
142. Защита ИИ-модели от извлечения (model extraction)
143. Обфускация кода и защита интеллектуальной собственности
144. Конфиденциальность данных пользователя при on-device инференсе
145. Соответствие требованиям GDPR при обработке данных на устройстве
146. Обработка персональных данных (изображения, голос) в ИИ-приложениях
147. Безопасное хранение моделей и ключей API
148. Аудит безопасности мобильного ИИ-приложения
149. Этические аспекты использования ИИ в мобильных приложениях

ПК 1.4. Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки

150. Управление зависимостями в мобильных проектах (Gradle, CocoaPods)
151. Публикация приложений в Google Play и App Store: основные требования
152. Обновление ИИ-моделей в мобильном приложении без обновления самого приложения
153. Обфускация кода и защита интеллектуальной собственности

Перечень заданий

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

1. Задания, связанные с проектированием алгоритмов, выбором подходов, построением пайплайнов обработки данных и логики работы модулей:
2. Разработка макета экрана приложения (Activity/Fragment) с использованием компонентов Material Design
3. Настройка навигации между экранами с помощью Intent (Android) или Segue (iOS)
4. Разработка адаптивного интерфейса (под разные размеры экранов) для ИИ-приложения
5. Написание функции изменения размера (Resize) и обрезки (Crop) изображения до нужных параметров модели (например, 224x224)
6. Реализация препроцессинга изображения: нормализация пикселей (приведение к диапазону 0–1 или -1–1)
7. Разработка модуля поворота изображения (Rotation) согласно EXIF-данным
8. Разработка модуля выделения области интереса (ROI) на изображении
9. Написание функции конвертации Bitmap в ByteBuffer (для передачи в TFLite)
10. Написание функции конвертации ByteBuffer с результатами обратно в Bitmap для отображения
11. Вывод информации о входных и выходных тензорах модели (размерность, тип данных)
12. Разработка функции интерпретации выходных данных (получение индекса класса с максимальной вероятностью)
13. Сопоставление индексов классов с текстовыми метками (labels.txt) для вывода названия объекта
14. Настройка параметров Task Library (порог уверенности, максимальное количество объектов)
15. Оптимизация работы с памятью: отказ от создания новых Bitmap при каждом кадре (использование переиспользуемых буферов)
16. Разработка приложения для распознавания текста (OCR) с камеры с использованием ML Kit / TFLite
17. Разработка простого чат-бота с офлайн-моделью для ответов на типовые вопросы
18. Создание приложения-переводчика с использованием локальной модели (или API)
19. Реализация приложения для распознавания жестов рук (на основе готовой модели)
20. Комплексный проект: разработка мобильного приложения, которое делает фото, классифицирует объект, сохраняет результат в локальную БД и отображает историю запросов

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием

21. Задания, связанные с непосредственной реализацией модулей, интеграцией библиотек, работой с камерой, UI, БД и inference:
22. Реализация обработки нажатий кнопок и вывода текстовых данных в UI-элементы (TextView/Label)
23. Настройка запроса разрешений (Permissions) на доступ к камере и хранилищу

24. Разработка экрана настроек приложения с сохранением данных в SharedPreferences/UserDefaults

25. Реализация простого таймера обратного отсчета с обновлением UI в реальном времени

26. Реализация уведомлений (Notifications) о завершении фоновой задачи

27. Создание и настройка меню приложения (Toolbar, Options Menu)

28. Реализация анимации загрузки (ProgressBar) при ожидании ответа от ИИ-модуля

29. Разработка модуля захвата фото с камеры устройства и отображения его в ImageView

30. Реализация выбора изображения из галереи устройства для дальнейшей обработки

31. Интеграция библиотеки OpenCV для мобильной платформы и выполнение базовой фильтрации (оттенки серого)

32. Реализация захвата видеопотока с камеры и его отображения в реальном времени (SurfaceView/TextureView)

33. Добавление зависимостей TensorFlow Lite в проект (Gradle/CocoaPods)

34. Загрузка модели .tflite из папки assets и инициализация Interpreter

35. Реализация запуска инференса для простой модели классификации изображений (например, MobileNet)

36. Реализация вывода результата классификации на экран

37. Интеграция квантованной модели (int8) и проверка корректности её работы

38. Использование GPU-делегата (GpuDelegate) для ускорения работы модели

39. Использование NNAPI-делегата на Android (или Core ML на iOS) для аппаратного ускорения

40. Интеграция библиотеки org.tensorflow:tensorflow-lite-task-vision

41. Реализация задачи классификации изображений с помощью ImageClassifier

42. Реализация задачи детекции объектов с помощью ObjectDetector (вывод bounding box на экран)

43. Интеграция библиотеки tensorflow-lite-task-text и реализация анализа тональности текста (Sentiment Analysis)

44. Реализация задачи ответов на вопросы (Question Answering) с использованием BertQuestionAnswerer

45. Интеграция библиотеки для аудио и реализация классификации аудиособытий (например, распознавание звука сирены)

46. Разработка экрана для ввода текста и вывода результата работы NLP-модели

47. Реализация асинхронного запуска инференса во избежание блокировки UI-потока

48. Реализация закрытия интерпретатора и освобождения ресурсов (close) при уничтожении Activity

49. Тестирование приложения на устройстве с ограниченными ресурсами (эмуляция низкой памяти)

ПК 1.3. Оформлять программный код в соответствии с техническим заданием

50. Задания, связанные с профилированием, оптимизацией, замером метрик, тестированием и обеспечением качества кода:

51. Профилирование времени выполнения инференса (замер Latency с помощью System.currentTimeMillis())

52. Сравнение производительности модели float32 и квантованной модели int8 на одном устройстве

53. Настройка количества потоков (NumThreads) для интерпретатора и оценка влияния на скорость

ПК 1.4. Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки

54. Инициализировать локальный репозиторий Git в папке с проектом и настроить пользователя (user.name, user.email).

55. Добавить файлы в индекс (git add) и сделать коммит с осмысленным сообщением (git commit -m).

56. Создать новую ветку (git branch), переключиться на нее и внести изменения.

57. Слить ветку с основной (git merge) и разрешить возникший конфликт.

58. Просмотреть историю коммитов (git log) и откатить изменения к предыдущему коммиту (git reset или git revert).

59. Связать локальный репозиторий с удаленным (git remote add) и отправить изменения (git push).