

ПРИМЕНЕНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ РАЗРАБОТКЕ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ ПОД ОПЕРАЦИОННУЮ СИСТЕМУ ANDROID

Горох П.И.¹, ассистент, p.goroh@bsuir.by
Кузмин И.А.², ассистент, i.kuzmin@bsuir.by

2026

1. Белорусский государственный университет информатики и радиоэлектроники
2. Белорусский государственный университет информатики и радиоэлектроники

Ключевые слова: искусственный интеллект; Android; мобильная разработка; большие языковые модели; Android Bench; машинное обучение; генерация кода; Jetpack Compose; Room; IDE; Android Studio; Gemini; GPT; Claude; автоматизация разработки; оценка ИИ-моделей.

Аннотация: В статье рассматривается применение искусственного интеллекта (ИИ) в разработке мобильных приложений под операционную систему Android с учётом современных тенденций и появлением специализированных бенчмарков для оценки возможностей больших языковых моделей (LLM). Показано, что существующие универсальные тесты не отражают специфику Android-разработки, поэтому для анализа используется официальный бенчмарк Android Bench (весна 2026 года), оценивающий точность решения практических задач, включая генерацию кода на Kotlin, работу с Jetpack Compose, Room и архитектурными компонентами. Приведена сравнительная характеристика популярных ИИ-моделей (GPT-5.5, Gemini 3.1 Pro Preview, Claude Opus 4.7, GPT-5.3 Codex, Claude 4.6 Sonnet), а также результаты их тестирования по показателю точности выполнения Android-задач. Дополнительно рассмотрены особенности интеграции ИИ в различные IDE, включая Android Studio, VSCode и IntelliJ IDEA. На основе анализа показано, что для исправления ошибок и разработки UI наиболее эффективна связка Android Studio и Gemini 3.1 Pro, тогда как для проектирования архитектуры предпочтительны Cursor с GPT-5.5 или Claude Opus. Сделан вывод о значимости ИИ-инструментов для повышения качества и скорости разработки Android-приложений.

В области разработки программного обеспечения с использованием искусственного интеллекта (ИИ) появился ряд бенчмарков для оценки возможностей больших языковых моделей (LLM). Разработчики под Android сталкиваются со специфическими задачами, которые не охвачены существующими бенчмарками, поэтому предлагается рассмотреть инструмент, ориентированный на ключевые показатели разработки под операционную систему Android.

Для сравнения ИИ-моделей и рабочих сред (IDE) для Android-разработки возьмём официальный бенчмарк Android Bench от Google (данные за весну 2026

года). Этот тест оценивает точность решения реальных задач, написание кода с Jetpack Compose, работу с Room DB и репозиториями [1].

Дадим краткую характеристику популярным ИИ-моделям, которые будут рассмотрены для сравнительного анализа:

1 GPT-5.5. Хорошо справляется с архитектурой и сложными интеграциями, однако предоставляет крайне дорогой API.

2 GPT-5.4. Неплохо справляет с архитектурой и тестированием репозитория, однако имеет высокую стоимость доступа к API.

3 Gemini 3.1 Pro Preview. Отлично работает с Android Studio, хорошо знает Kotlin, но иногда даёт лишний шаблонный код.

4 Claude Opus 4.7. Чистый UI-код (Compose), хорошо удерживает контекст, но имеет жёсткие ограничения на запросы, и характеризуется относительно медленной скоростью генерации ответа.

5 GPT-5.3 Codex. Быстро пишет и дописывает функции, однако тяжело понимает сложные UI-компоненты.

6 Gemini 3.1 Pro Preview. Отлично работает с Android Studio, хорошо знает Kotlin, но часто генерирует шаблонный код.

7 Claude 4.6 Sonnet. Быстро генерирует и исправляет ошибки, но часто теряет контекста в больших проектах.

Сравнение ИИ-моделей будет произведено по Android Bench, с использованием показателя точности, который показывает средний процент успешно решённых Android-задач и ошибок на 100 тестов. Результаты сравнения продемонстрированы в таблице 1.

ИИ-модель	Показатель точности, %
GPT-5.5	74,0
GPT 5.4	72,4
Gemini 3.1 Pro Preview	72,4
Claude Opus 4.7	68,7
GPT-5.3 Codex	67,7
Gemini 3.0 Pro Preview	60,4
Claude 4.6 Sonnet	58,4

Таблица 1 – Сравнение ИИ-моделей в Android разработке

Для примера, базовая модель Gemini 2.5 Flash набирает всего 16.7% в этом тесте, что доказывает, что простые модели мало подходят для сложной Android-разработки.

Интеграцию ИИ оценим по удобству, размеру контекстного окна, поддержке экосистемы Gradle/Kotlin внутри рабочей среды IDE. Результаты сравнительного анализа интеграции приведем в таблице 2.

IDE	ИИ-модель	Понимание контекста	Удобство для Android
Android Studio	Gemini Code Assist	Относительно большое окно до 128 тысяч токенов	Официальный инструмент, ИИ напрямую работает с эмуляторами, Logcat, профилировщиком памяти и SDK
VSCode	Cursor	Большое окно до 1 миллиона токенов	Хорош для создания архитектуры, но нет встроенных инструментов для сборки и отладки.
IntelliJ IDEA	JetBrains AI Assistant	Среднее, ориентировано на структуру проектов JetBrains	Хорошо работает с Kotlin и Java, но уступает Android Studio при работе с ресурсами и JetpackCompose

Таблица 2 – Сравнение IDE в связке с ИИ-моделями

Таким образом для исправления ошибок и написания UI в готовом проекте удобнее использовать Android Studio с Gemini 3.1 Pro, благодаря тесной интеграции с Android SDK. Что же касается создания сложной архитектуры с нуля лучше подходят Cursor с GPT-5.5 или Claude Opus – они эффективнее связывают много файлов между собой.

Список использованных источников

1. Android Developers [Электронный ресурс]. – Режим доступа: <https://developer.android.com>. – Дата доступа: 10.02.2025.