

## СРАВНЕНИЕ КОДИРУЮЩИХ ИИ-АГЕНТОВ НА ПРИМЕРЕ ХАРАКТЕРНЫХ ДЛЯ ANDROID-РАЗРАБОТКИ ЗАДАЧ

**Зюзин Р. А.<sup>1</sup>**

**Научный руководитель – канд. фил. наук, доцент Анашкина Н. А.<sup>1</sup>**

<sup>1</sup>Университет ОмГТУ

RAZ3113@yandex.ru

### **Введение**

В последние годы популярность искусственного интеллекта сильно выросла. Во многом это связано с повышением функциональных возможностей генеративного ИИ (ГИИ) [1]. Бурное развитие индустрии ГИИ привело к ситуации, заключающейся в постоянном появлении новых языковых моделей, ИИ-ассистентов и прочих инструментов, основанных на ИИ. Так за последний год произошёл ряд крупных релизов кодирующих ИИ-агентов, повлиявших на популяризацию использования такого типа систем генерации кода в рамках характерных для android-разработки задач [2, 3].

Учитывая эту популяризацию, актуальными становятся ряд вопросов. Способны ли ИИ-агенты решать задачи кодирования лучше, чем «классические» ИИ-ассистенты? Если да, то какое из существующих решений лучше всего подходит для решения характерных для android-разработки задач? Какие из популярных решений бесплатны и доступны в РФ? Насколько значимо влияние конкретного ИИ-агента как абстракции над LLM при генерации кода? Ответы на эти вопросы рассматриваются в рамках исследования.

### **Основная часть**

Данное исследование является логическим продолжением «сравнения моделей генеративного искусственного интеллекта на примере использования в android-разработке», проведённого чуть менее года назад [4]. Необходимость нового исследования обусловлена появлением на рынке ГИИ принципиально нового типа систем, потенциально обеспечивающего улучшение результатов использования LLM в ходе использования таких моделей в android-разработке.

В рамках текущего исследования для оценивания корректности работы ИИ-агентов выделены критерии логики их работы, которым они должны соответствовать в процессе генерации кода. Данные критерии рассматриваются наравне с критериями оценивания сгенерированного кода и в итоге влияют на полученный ИИ-агентом балл за выполнение задания.

Так как на текущий момент уже существуют сотни ИИ-агентов для генерации кода, то принято решение рассмотреть в исследовании только решения, интегрированные в IDE «Android Studio (полная интеграция или на уровне плагина). Также рассматриваются только бесплатные версии решений. Среди агентов, проходящих по описанным выше критериям, взята выборка из ряда наиболее популярных систем ИИ-агентов для написания кода. Агенты из этой выборки решали 3 задачи, охватывающие различные аспекты современной android-разработки: создание навигации для UI-слоя, вёрстка UI-элементов и создание data-слоя приложения. Итоговые значения, полученные в результате исследования, посчитаны как среднее арифметическое значение суммы баллов по критериям для каждого из заданий. Для агентов, получивших одинаковое количество баллов, учитывалось среднее количество попыток, потребовавшихся на соответствие условиям каждого из заданий.

ИИ-агенты в сравнении с ИИ-ассистентами из предыдущего исследования показали на одном и том же задании более хорошие результаты. С заданиями среди

текущей выборки решений стабильно качественно справились агенты «Gemini» и «GitHub Copilot». Ряд ИИ-агентов из выборки доступны в РФ. Лучшим среди них признана «Koda». Также по итогам исследования сделан вывод о том, что ИИ-агент, как дополнительный слой абстракции над LLM, значительно влияет на итоговое поведение агента при решении задачи и на результат её решения (разница в итоговых баллах более 15%).

Для оценки достоверности результата исследования произведён расчёт коэффициента корреляции этих результатов с индексом кодирования, полученным в результате исследований «Artificial Analysis» для LLM, аналогичным используемым в исследовании [5]. Обнаружена сильная положительная корреляция, что верифицирует результаты исследования.

### **Выводы**

Получены достаточно достоверные данные об оценке способностей кодирующих ИИ-агентов к генерации кода в рамках характерных в android-разработке задач. Сравнение с результатами прошлого исследования [4] наглядно продемонстрировало стремительное развитие моделей ГИИ, а также возможность появления принципиально новых типов систем генерации кода, позволяющих сделать результат более приближенным к поставленным пользователем (программистом) задачам.

### **Литература**

1. Openai – Improving Language Understanding by Generative Pre-Training [Электронный ресурс]. – Режим доступа: [https://cdn.openai.com/research-covers/language-unsupervised/language\\_understanding\\_paper.pdf](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf) (Дата обращения 16.02.2026).
2. See Junie in Action: Your Coding Agent in IntelliJ IDEA, a JetBrains IDE [Электронный ресурс]. – Режим доступа: <https://youtu.be/ufPGsZtqrac?si=PeUwcixdex1vH8kj> (Дата обращения 16.02.2026).
3. Agent mode | Android studio [Электронный ресурс]. – Режим доступа: <https://developer.android.com/studio/gemini/agent-mode> (Дата обращения 16.02.2026).
4. Зюзин Р. А. Сравнение моделей генеративного искусственного интеллекта на примере использования в android-разработке // ИТ. НАУКА. КРЕАТИВ. Материалы II Международного форума. 2025. Том 2. С. 38–50.
5. Artificial Analysis – Intelligence Benchmarking [Электронный ресурс]. – Режим доступа: <https://artificialanalysis.ai/methodology/intelligence-benchmarking> (Дата обращения 16.02.2026).