

ПРИМЕНЕНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ПРОГРАММИРОВАНИИ

APPLICATION OF ARTIFICIAL INTELLIGENCE IN PROGRAMMING

V. Bunkin

Summary. Artificial intelligence technologies are radically transforming the technological landscape, including programming. The article considers key aspects concerning the application of artificial intelligence technologies for code generation. The problems and difficulties that need to be solved to reveal the full potential of this technology for software developers are highlighted separately. A special emphasis is made on the application of artificial intelligence in the process of testing and code verification. Examples of the most popular platforms are given.

Keywords: artificial intelligence, programme code, testing, generation, development.

Бунькин Виктор Иванович

Кандидат технических наук, доцент,
Университет «Синергия»
bunkinvi@gmail.com

Аннотация. Технологии искусственного интеллекта кардинальным образом трансформируют технологический ландшафт, в том числе в области программирования. В статье рассмотрены ключевые аспекты, касающиеся применения технологий искусственного интеллекта для генерации кода. Отдельно выделены проблемы и сложности, которые требуют решения с целью раскрытия всего потенциала данной технологии для разработчиков программ. Особый акцент сделан на применении искусственного интеллекта в процессе тестирования и проверки кода. Приведены примеры наиболее популярных платформ.

Ключевые слова: искусственный интеллект, программный код, тестирование, генерация, разработка.

Искусственный интеллект (ИИ) добился значительных успехов в различных отраслях, и в данном случае разработка и тестирование программного обеспечения (ПО) не являются исключением. ИИ стремительно меняет процесс создания новых программных систем и продуктов.

Разработка ПО — сложный и трудоемкий процесс, включающий различные этапы анализа и кодирования. Кроме того, он является дорогостоящим, особенно в среде, которая придерживается процедур, стандартов и должна соблюдать определенный порядок команд [1]. В данном контексте автоматизированная генерация кода становится все более важной, особенно благодаря использованию машинного обучения (ML) и инструментов ИИ. Эти инструменты значительно повышают эффективность и скорость процесса написания кода, позволяя программистам генерировать, дополнять или рецензировать код с помощью простых текстовых запросов на естественном языке. Пользователь вводит текстовую подсказку, описывающую, что должен делать код, а генеративный инструмент разработки на основе ИИ автоматически создает код. Он также может модернизировать устаревший код и переводить код с одного языка программирования на другой.

Отдельного внимания заслуживает использование ИИ в процессе тестирования ПО. В некоторых современных программных приложениях миллионы строк кода, поэтому тестировщикам сложно найти и устранить все ошибки. ИИ может помочь в тестировании и отладке,

анализируя и обрабатывая большие объемы данных для выявления тенденций и отклонений, которые обычные люди могут не заметить [2]. О том, насколько перспективным является данное направление развития ИИ наглядно свидетельствует объем мирового рынка соответствующих продуктов. Согласно отчету Statista рынок ИИ для тестирования программного обеспечения к 2033 году составит около 10,6 млрд долларов США по сравнению с 1,9 млрд долларов США в 2023 году, а среднегодовой темп роста достигнет 18,70 % в прогнозируемый период с 2024 по 2033 год (см. рис. 1) [3].

Таким образом, с учетом вышеизложенного, изучение перспектив и возможностей использования ИИ в программировании представляет собой важную научно-практическую задачу, которая и предопределила выбор темы данной статьи.

Описанию ключевых технологий, лежащих в основе генерации кода с помощью ИИ, посвятили свои труды Базарбаев Х., Байрамов Т., Дадаева Л., Кононова З.А., Алтухова С.О., Mary Kalantzis, Bill Cope, Juliana Vélez, William McShea, Hila Shamon.

Над разработкой методов улучшения генеративных моделей ИИ для создания программного кода, которые включают расширение входных запросов контекстной информацией, такой как типы токенов кода, предшествующие типы задач и выходные данные, трудятся Подкидышева Е.А., Пяткова К.А., Ручкин В.Н., Сергиевский М.В., Винокур А.И., Sima Das, Ajay Kumar Balmiki, Nimay Chandra Giri.

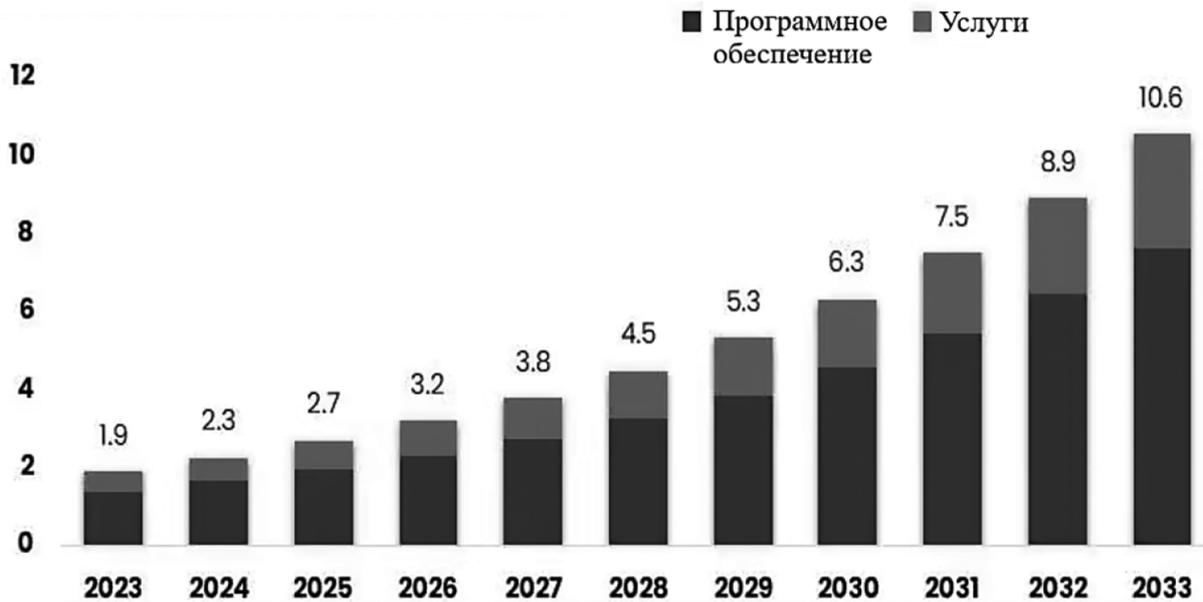


Рис. 1. Размер глобального рынка ИИ для тестирования программного обеспечения по компонентам [3]

Преимущества тестирования программного кода с использованием ИИ в условиях гибкой разработки, когда требования часто меняются, а поддерживать обновленные тестовые примеры вручную может быть проблематично, описывают Вьюнов Д.А., Быков Д.А., Березин И.С., Александрова Н.А., Prateek Sikka, Abhijit R Asati, Chandra Shekhar.

Высоко оценивая имеющиеся на сегодняшний день труды и наработки, следует отметить, что ряд вопросов в данной предметной области требует отдельного внимания и более глубокого анализа. В частности, нерешенной является проблема организации модульного тестирования с использованием ИИ. Также в развитии нуждаются подходы, GitHub Copilot и TabNine, которые, несмотря на всю свою прогрессивность, смещают задачи написания кода в сторону его обзора.

Таким образом, цель статьи заключается в изучении возможностей и перспектив применения ИИ в программировании, в частности в процессе написания кода и его тестирования.

При генерации кода с помощью ИИ используются алгоритмы, которые обучаются на существующем исходном коде — как правило, создаваемом в рамках проектов с открытым кодом для публичного использования — и генерируют новый код на основе этих примеров [4]. Большие языковые модели способны обрабатывать и понимать язык (в том числе, и язык программирования), конструировать текст, отвечать на вопросы, изучать закономерности и связи в языке.

В настоящее время ИИ генерирует код тремя способами:

1. Разработчик начинает набирать код, а ИИ пытается его автоматически заполнить.
2. Разработчик пишет комментарий на естественном языке, и ИИ генерирует предложение, основываясь на том, чего хочет добиться разработчик.
3. Разработчик напрямую общается с ИИ, например, просит его написать что-то конкретное или исправить ошибку.

На рисунке 2 показан простой сценарий интерактивного программирования с использованием ИИ.

Представленная на рис. 1 концепция подразумевает итеративное взаимодействие пользователей с моделями ИИ с помощью подходов с малым количеством кода, пока они не достигнут желаемого результата. Однако несмотря на то, что обратная связь с пользователями имеет большое значение, ее сохранение ограничивается одним сеансом общения. Это ограничение обусловлено свойствами, присущими генеративным моделям ИИ, которые требуют явного переобучения для интеграции новых данных, что на сегодняшний день является открытой проблемой использования ИИ в процессе программирования.

Очевидно, что выше обозначенная проблема использования генеративного ИИ для создания программного кода является отнюдь не единственной. Применение данного инструмента сталкивается с рядом проблем, связанных с качеством данных, сложностью, неоднозначностью, пониманием контекста, этическими соображениями и производительностью. Решение этих сложностей требует междисциплинарных усилий, включая достижения в области исследований ИИ, управления данными, практики разработки программного обеспе-

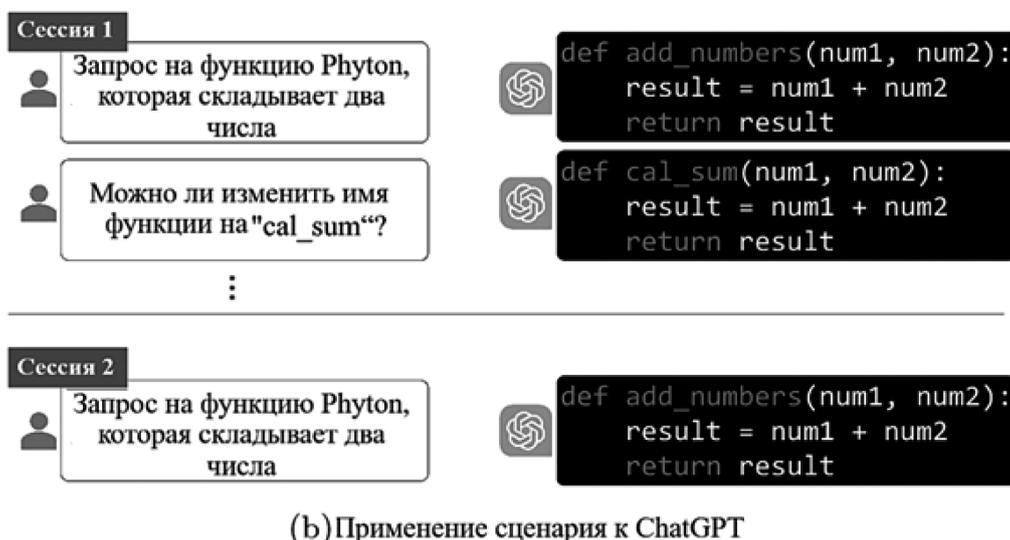
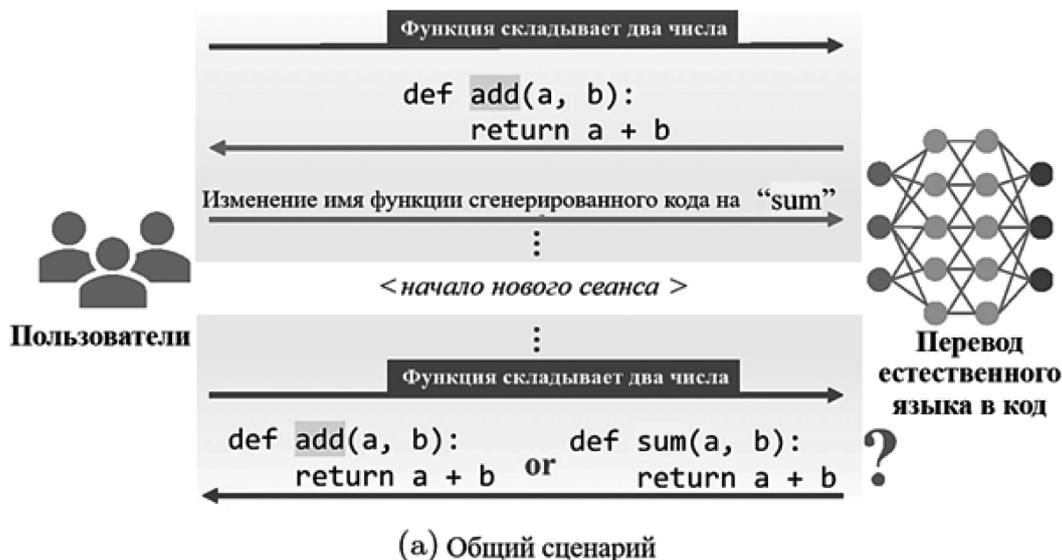


Рис. 2. Сценарии создания кода с помощью ИИ [5]

чения и нормативно-правовой базы. Когда эти проблемы будут преодолены, разработчики смогут использовать весь потенциал генерации кода на основе ИИ для повышения производительности и качества программного продукта.

В таблице 1 представлен краткий обзор проблем, с которыми сталкивается программирование на основе ИИ.

Что касается тестирования программного кода, то в данной сфере возможности ИИ включают в себя использование автоматизированных решений, которые основаны на алгоритмах машинного обучения вместо predefined правил. Эти интеллектуальные системы могут:

1. Обучаться на основе существующих наборов данных.
2. Адаптироваться к новым сценариям тестирования.

3. Самостоятельно делать прогнозы и принимать решения.
4. Совершенствоваться с течением времени при увеличении объема данных и времени использования.

По сути, ИИ привносит в тестирование программного обеспечения возможности самообучения. Это позволяет выполнять динамические и автономные тесты без явного программирования [6].

На сегодняшний день одним из самых перспективных инструментов в тестировании ПО с использованием ИИ является платформа Diffblue Cover, которая автоматически пишет модульные тесты для кода Java, одного из самых популярных языков программирования. Diffblue Cover применяет обучение с подкреплением для анализа кодовой базы и генерирует человекочитаемые, исполняемые тесты, которые охватывают широкий спектр

Проблемы, сопровождающие использование ИИ при написании программного кода

Проблемы	Идентификация	Анализ
Качество и количество данных	Модели ИИ требуют большого объема высококачественных обучающих данных	Ограниченная доступность разнообразных, хорошо проаннотированных наборов данных кода может препятствовать производительности модели и приводить к предвзятому или неточному генерированию кода
Сложность и ремонтпригодность	Генерируемый код может быть чрезмерно сложным или трудным для сопровождения	Сложность генерируемого кода может препятствовать сотрудничеству, повышать риск ошибок при обслуживании и мешать масштабируемости кодовой базы
Двусмысленность и понимание естественного языка	Извлечение точных требований из естественного языка может быть сложной задачей	Данные естественного языка могут содержать двусмысленность, зависимость от контекста и лингвистические нюансы, что приводит к неточностям или неправильной интерпретации при генерации кода
Отсутствие понимания контекста	Моделям ИИ может не хватать контекстного понимания семантики кода и ограничений	Контекстуально неадекватная генерация кода может привести к неэффективности, несоответствиям и проблемам совместимости в созданной кодовой базе
Этические и юридические аспекты	При генерации кода с помощью ИИ возникают проблемы, связанные с интеллектуальной собственностью и этикой	Разработчики должны учитывать этические и юридические аспекты, связанные с правом собственности, лицензированием и неправомерным использованием генерируемого ИИ кода
Производительность и масштабируемость	Модели ИИ могут демонстрировать узкие места в производительности или ограничения масштабируемости	Оптимизация эффективности и масштабируемости систем генерации кода на основе ИИ имеет решающее значение для своевременной и экономически эффективной разработки

сценариев. Еще один мощный инструмент — EvoSuite, фреймворк с открытым исходным кодом, использующий генетические алгоритмы для генерации тестовых наборов для Java-программ, однако генерируемые EvoSuite тесты не так читабельны, как те, что создает Diffblue Cover.

Таким образом, подводя итоги, отметим, ИИ является краеугольным камнем современных методологий создания ПО, обещая ускорение циклов разработки, повышение производительности и качества кода. Одним из новаторских достижений в этой области является генерация кода на основе ИИ, а также тестирование полученного продукта.

ЛИТЕРАТУРА

1. Бевзенко С.А. Исследование методов автоматического программирования с применением искусственного интеллекта // Молодой ученый. 2024. № 11 (510). С. 13–15.
2. Kaigeng Li, Wechen Jia, Zhi Li Regulation of Appropriate Prompts for Users in Text-Based Generative Artificial Intelligence Programs // Software: Practice and Experience. 2024. № 56. P. 98–108.
3. Janosch Menke, Samuel Homberg Introduction to artificial intelligence and deep learning using interactive electronic programming notebooks // Archiv der Pharmazie. 2023. Volume 356, Issue 7. P. 13–19.
4. Мирабова Л., Худайназарова М. Искусственный интеллект и машинное обучение в программировании: тенденции и перспективы // Ceteris Paribus. 2023. № 10. С. 65–67.
5. Edgar Serna M. Integration of properties of virtual reality, artificial neural networks, and artificial intelligence in the automation of software tests: A review // Journal of Software: Evolution and Process. 2019. Volume 31, Issue 7. P. 76–83.
6. Грызлов Д.В., Куваева Е.Н. Использование искусственного интеллекта в программировании // Научный аспект. 2024. Т. 17. № 8. С. 2121–2127.

© Бунькин Виктор Иванович (bunkinvi@gmail.com)
Журнал «Современная наука: актуальные проблемы теории и практики»