

ОСОБЛИВОСТІ ВИКОРИСТАННЯ WINDSURF ЯК СПЕЦІАЛІЗОВАНОГО ІНСТРУМЕНТУ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ РОЗРОБКИ КОДУ

Сучасний етап розвитку інформаційних технологій характеризується активним впровадженням штучного інтелекту (ШІ) в різноманітні сфери людської діяльності. Одним із найбільш перспективних напрямків застосування ШІ є розробка програмного забезпечення. Дослідження D. Guo, Q. Zhu, D. Yang та інших співавторів [1] показали, що великі мовні моделі демонструють значний потенціал у генерації програмного коду, що підтверджується розробленим ними набором тестів HumanEval. Робота команди Meta AI під керівництвом B. Rozière та співавторів [2] представила сімейство моделей Code Llama, здатних працювати з контекстами до 100K токенів, відкриваючи нові можливості для аналізу та генерації складних програмних рішень. Ці досягнення стимулювали розробку спеціалізованих інструментів, що дозволяють ефективно використовувати потенціал ШІ у розробці коду. Важливий внесок у цьому напрямку зробили J. Liu, C. S. Xia, Y. Wang та L. Zhang [3], розробивши фреймворк EvalPlus для ретельного тестування згенерованого коду. Windsurf, розроблений компанією Codeium, є інноваційним інтелектуальним асистентом програмування, що інтегрує ці напрацювання та пропонує розробникам унікальні можливості для взаємодії з ШІ на всіх етапах розробки. Метою даної роботи є ґрунтовне вивчення функціональних можливостей Windsurf, аналіз його переваг та обмежень, а також визначення перспектив його використання в контексті сучасних тенденцій розвитку генеративного ШІ. Дослідження базується на аналізі технічної документації Windsurf, практичному тестуванні його можливостей, а також порівнянні з аналогічними інструментами.

Windsurf, розроблений компанією Codeium, являє собою повнофункціональне IDE нового покоління, яке фокусується на інтеграції з генеративним ШІ. Ключовим компонентом Windsurf є Cascade – інтелектуальний чатбот, що дозволяє розробнику взаємодіяти з ШІ за допомогою текстових запитів [4]. Cascade здатний генерувати код на основі опису, надавати допомогу

у вирішенні програмних завдань, а також здійснювати пошук необхідної інформації. Відмінною рисою Cascade є його здатність до «real-time» колаборації з розробником. Система відстежує дії користувача в редакторі коду та автоматично адаптує свої пропозиції, враховуючи контекст поточної роботи. Наприклад, при перейменуванні змінної Cascade автоматично запропонує змінити її назву в усіх релевантних місцях коду. Така тісна інтеграція з процесом розробки потенційно може значно підвищити продуктивність праці, але водночас, виникає питання про можливу надмірну залежність розробника від ШІ та втрату контролю над процесом кодування.

Windsurf підтримує роботу з різними моделями генеративного ШІ, включаючи GPT-4o, Claude 3.5 Sonnet, а також власну модель Cascade Base, оптимізовану для швидкої генерації коду. Важливим аспектом є можливість тонкого налаштування параметрів ШІ, таких як швидкість автодоповнення, дозвіл на автоматичне виконання команд терміналу та інші.

Cascade є центральним елементом Windsurf, що забезпечує взаємодію розробника з генеративним ШІ. Цей інструмент реалізує концепцію «AI Flows», пропонуючи новий підхід до програмування за допомогою штучного інтелекту, що відрізняється від традиційних методів. Cascade доступний у двох режимах: Write та Chat. Режим Write призначений для безпосередньої генерації та модифікації коду, дозволяючи створювати нові файли та редагувати існуючі. Режим Chat оптимізований для відповідей на питання, пов'язані з кодом, або для обговорення загальних принципів програмування.

Для зручності роботи Cascade пропонує ряд додаткових функцій, таких як можливість завантаження зображень, які можуть бути використані як контекст для генерації коду, та інтеграція з веб-пошуком для самостійного знаходження необхідної інформації. Крім того, Cascade вміє розпізнавати помилки в коді та пропонувати способи їх виправлення, а також пояснювати фрагменти коду, виділені користувачем, що робить його цінним інструментом для навчання та розуміння коду.

Окрім Cascade, ще однією ключовою особливістю Windsurf є система Memories, яка забезпечує глибоке розуміння кодової бази проекту. Memories являє собою механізм контексту, що дозволяє ШІ запам'ятовувати та використовувати інформацію про структуру коду, залежності між модулями та інші важливі аспекти проекту. Існує два типи Memories: визначені користувачем та згенеровані Cascade автоматично.

Користувачки Memories дозволяють розробнику задавати власні правила та інструкції для Cascade. Зазначені правила можуть бути оформлені у вигляді текстових файлів з розширенням `windsurf.rules` для локальних правил, що діють в межах одного проєкту, або `global.rules.md` для глобальних правил, що застосовуються до всіх проєктів. Правила можуть містити вказівки щодо стилю кодування, використання певних бібліотек або API, а також інші рекомендації. Cascade враховує ці правила при генерації коду, що дозволяє отримувати результати, які максимально відповідають вимогам проєкту.

Автоматично згенеровані Memories створюються Cascade в процесі роботи над проєктом. Система аналізує код, виявляє важливі патерни та зберігає їх для подальшого використання. Наприклад, Cascade може запам'ятати, що в певному проєкті перевага надається використанню конкретної бібліотеки для роботи з даними, і в подальшому буде автоматично пропонувати її при генерації нового коду. Користувач може переглядати та редагувати автоматично згенеровані Memories через налаштування Windsurf.

Система Memories забезпечує більш глибокий рівень інтеграції ШІ в процес розробки. Завдяки Memories, Cascade стає не просто інструментом для генерації коду, а справжнім інтелектуальним помічником, який адаптується до стилю та потреб конкретного розробника.

Windsurf пропонує ряд практичних інструментів, що роблять його зручним та ефективним середовищем розробки. Інтеграція з терміналом дозволяє виконувати команди безпосередньо з IDE. Cascade може автоматично генерувати команди на основі запитів користувача. Наприклад, розробник може попросити Cascade «встановити пакет `numpy`» і система самостійно сформує та виконає відповідну команду в терміналі. Windsurf підтримує автоматичне виконання команд з можливістю налаштування списків дозволених та заборонених команд, що дозволяє автоматизувати рутинні операції, зберігаючи при цьому контроль над безпекою.

Для досвідчених користувачів передбачено розширені можливості, такі як підтримка SSH та робота з контейнерами Docker, які дозволяють використовувати Windsurf для розробки складних проєктів з розподіленою архітектурою.

Підсумовуючи, Windsurf є потужним інструментом для розробки програмного забезпечення з використанням генеративного ШІ. Його унікальні можливості, такі як Cascade з підтримкою «*real-time*» колаборації, система Memories для глибокого розуміння контексту проєкту, а також тісна інтеграція з терміналом та підтримка

розширених функцій, роблять його одним з найбільш перспективних IDE на ринку. Використання Windsurf потенційно може значно підвищити продуктивність розробки, скоротити час на виконання рутинних завдань та покращити якість коду, особливо в проєктах, де активно застосовуються технології ШІ, таких як розробка ігор, вебдодатків зі складним інтерфейсом користувача, або систем для аналізу даних.

Подальший розвиток Windsurf, ймовірно, буде пов'язаний з розширенням підтримки різних мов програмування, вдосконаленням моделей генеративного ШІ, а також з появою нових функцій, що ще більше спростять процес розробки. Однак, важливо зазначити, що використання подібних інструментів ставить перед науковою спільнотою ряд важливих питань. Необхідно дослідити, як забезпечити безпеку та надійність коду, згенерованого ШІ, а також як вплине використання таких інструментів на навички розробників у довгостроковій перспективі. Якою буде роль людини-програміста в епоху, коли значна частина коду може бути згенерована автоматично? Як зміняться наші уявлення про творчість, авторство та інтелектуальну власність у контексті програмування? Чи зможемо ми довіряти машинам у прийнятті рішень, що стосуються складних програмних систем? Виходячи з комплексу цих запитань, потрібні подальші дослідження у цій галузі, щоб повністю зрозуміти потенціал та обмеження взаємодії людини та ШІ в контексті розробки програмного забезпечення. Безсумнівно, Windsurf є важливим кроком на шляху до створення інтелектуальних середовищ розробки майбутнього, і його подальше вивчення та вдосконалення є ключовим завданням для дослідників та розробників.

Література:

1. Guo, D., Zhu, Q., Yang, D., Xie, Z., Dong, K., Zhang, W., Chen, G., Bi, X., Wu, Y., Li, Y. K., Luo, F., Xiong, Y., & Liang, W. (2024). DeepSeek-Coder: When the Large Language Model Meets Programming – The Rise of Code Intelligence. DOI: <https://doi.org/10.48550/arXiv.2401.14196>
2. Rozière, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X., Adi, Y., Liu, J., Remez, T., Rapin, J., Kozhevnikov, A., Evtimov, I., Bitton, J., Bhatt, M. P., Ferrer, C. C., Grattafiori, A., Xiong, W., D'efossez, A., Copet, J., Azhar, F., Touvron, H., Martin, L., Usunier, N., Scialom, T., & Synnaeve, G. (2023). Code Llama: Open Foundation Models for Code. DOI: <https://doi.org/10.48550/arXiv.2308.12950>
3. Liu, J., Xia, C., Wang, Y., & Zhang, L. (2023). Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. DOI: <https://doi.org/10.48550/arXiv.2305.01210>
4. Codeium Windsurf. Getting Started. Documentation. URL: <https://docs.codeium.com/windsurf/getting-started>