

Ғылыми мақала

FTAMP 50.14.17

<https://doi.org/10.55956/VTRK6250>

Б.М.Жақсыбаев

Магистрант

М.Х. Дулати атындағы Тараз университеті

Тараз қ., Қазақстан

Bekzat.zhm@gmail.com

Ж.К.Тасжурекова 

Техника ғылымдарының кандидаты

М.Х. Дулати атындағы Тараз университеті

Тараз қ., Қазақстан

zhk.taszhurekova@dulaty.kz

М.А.Ахметжанов 

Физика-математика ғылымдарының кандидаты

М.Х. Дулати атындағы Тараз университеті

Тараз қ., Қазақстан

Bekzat.zhm@gmail.com

ЖАСАНДЫ ИНТЕЛЛЕКТ ДӘУІРІНДЕГІ БАҒДАРЛАМАЛАУ: «ВАЙБКОДИНГ» ФЕНОМЕНІ ЖӘНЕ ТЕХНОЛОГИЯЛЫҚ САУАТТЫЛЫҚТЫҢ АРТЫҚШЫЛЫҒЫ

Аңдатпа. Бұл мақалада Cursor IDE және Claude Code сияқты заманауи жасанды интеллект (ЖИ) құралдарының бағдарламалық қамтамасыз етуді әзірлеу саласына тигізетін ықпалы талданды. Автор «вайбкодинг» ұғымын енгізді – бұл бағдарламашының төмен деңгейлі код жазудан гөрі, жоғары деңгейлі нұсқаулар беру, архитектураны жобалау және технологиялық стек таңдау арқылы ЖИ-ге бағыт беру процесі. Жеке стартапты әзірлеу кезінде 10 күн ішінде 140 000 жол кодты тек ЖИ көмегімен генерациялау тәжірибесіне негізделген кейс-стади ұсынылды. Бұл дәстүрлі тәсілмен салыстырғанда үлкен команданың жұмысын алмастыра алатынын көрсетті. Зерттеудің негізгі қорытындысы – бүгінгі таңда бағдарламашының құндылығы оның код жазу жылдамдығында емес, технологияларды терең түсіну, оларды дұрыс таңдай білу және ЖИ агенттерін тиімді басқару қабілетінде жатыр.

Тірек сөздер: Жасанды интеллект, код генерациясы, бағдарламалық инженерия, Cursor, Claude Code, вайбкодинг.

Кіріспе. Соңғы жылдары жасанды интеллект (ЖИ) технологияларының қарқынды дамуы көптеген саланы, соның ішінде бағдарламалық қамтамасыз етуді әзірлеуді түбегейлі өзгертуде. Егер бұрын GitHub Copilot сияқты құралдар кодты толықтыру және қарапайым функцияларды жазу арқылы бағдарламашыларға көмекші ретінде қызмет етсе, қазіргі таңда Cursor IDE және Claude Code сияқты кешенді ЖИ-агенттер пайда болды. Бұл жүйелер тек код фрагменттерін ұсынып қана қоймай, күрделі

техникалық тапсырмаларды толық-тай орындауға, кодты рефакторинг жасауға, тіпті жобаның архитектурасын құруға қабілетті [1]. Осының нәтижесінде бағдарламашының рөлі мен оған қойылатын талаптар айтарлықтай трансформациялануда.

Бұл зерттеудің өзектілігі дәстүрлі бағдарламалау дағдыларының құнсыздануы және жаңа құзыреттіліктерге деген сұраныстың артуымен байланысты. Қазіргі нарықта junior және middle деңгейдегі мамандардың негізгі міндеті саналатын стандартты

кодты жазу процесін ЖИ тиімдірек әрі жылдам орындай алады. Бұл жағдай «бағдарламашы енді қажет емес пе?» деген заңды сұрақ тудырады. Алайда, біздің гипотезамыз бойынша, бағдарлама-шының рөлі жойылмайды, керісінше, жаңа деңгейге көтеріледі: ол код жазушыдан (coder) технологиялық стратегке және ЖИ-оркестраторға айналады. Біз бұл жаңа парадигманы «вайбкодинг» (ағылш. *vibe* – атмосфера, бағыт және *coding* – кодтау) деп атаймыз. Вайбкодинг – бұл бағдарламашының нақты синтаксистік конструкцияларды теруге уақыт жұмсамай, жобаның «вайбын» – оның мақсатын, архитектуралық принциптерін, технологиялық шешімдерін анықтап, ЖИ-ге жоғары деңгейлі нұсқаулар беру процесі.

Мақаланың мақсаты – «вайбкодинг» феноменін теориялық және практикалық тұрғыдан зерттеу, Cursor IDE және Claude Code сияқты ЖИ-агенттердің мүмкіндіктерін салыстырмалы талдау және осы технологиялардың негізінде бағдарламашының құндылығын анықтай-тын негізгі факторлардың өзгеруін көрсету. Зерттеу автордың жеке тәжірибесіне – 10 күн ішінде ЖИ көмегімен 140 000 жолдан тұратын күрделі жобаны іске асыру кейсіне сүйенеді. Бұл тәжірибе технологиялық сауаттылықтың код жазу дағдысынан маңыздырақ екенін дәлелдейтін эмпирикалық негіз болады.

Зерттеу шарттары мен әдістері. Бұл зерттеу бағдарламалық қамтамасыз етуді әзірлеу процесіндегі ЖИ-агенттердің тиімділігін бағалауға бағытталған. Зерттеудің негізгі әдістері ретінде кейс-стади (case study), салыстырмалы талдау және техникалық құжаттаманы талдау қолданылды.

Зерттеу нысаны – ЖИ-ге негізделген заманауи код генерациялау құралдарын пайдалана отырып, коммерциялық жобаны әзірлеу процесі.

Зерттеуде екі негізгі ЖИ-агент талданды: Cursor IDE және Claude Code. Бұл құралдар өздерінің архитектуралық ерекшеліктеріне байланысты тандалды, өйткені олар бағдарламалауға екі түрлі көзқарасты ұсынады.

1. Cursor IDE (1.2 нұсқасы, 2025 ж. шілде) – бұл VS Code редакторының негізінде жасалған, ЖИ функциялары терең интеграцияланған дамыту ортасы. Оның негізгі ерекшеліктері: мультифайлдық өзгерістерге арналған Composer, жоба контекстін есте сақтауға арналған Memories жүйесі және фондық режимде күрделі тапсырмаларды орындайтын Background Agent [2]. Cursor пайдаланушыға интерактивті тәжірибе ұсынады, мұнда бағдарламашы мен ЖИ арасындағы диалог IDE ішінде жүзеге асады. Ол жүйелік промпттар мен LSP (Language Server Protocol) арқылы кодтың стилі мен архитектурасын талдап, оңтайлы шешімдер ұсынады [3].

2. Claude Code (Claude 4 интеграциясымен, 2025 ж.) – бұл CLI (командалық жол интерфейсі) арқылы жұмыс істейтін, автономдылыққа бағытталған агент. Оның негізгі артықшылығы – күрделі, көпқадамды тапсырмаларды адамның минималды араласуымен орындау қабілеті. Claude Code Git операцияларын басқара алады, тесттерді іске қосып, қателерін түзетеді және веб-іздеу арқылы өзекті ақпаратты табады. Ол инкрементті растау (yes/no/auto) механизмі арқылы әрбір қадамын пайдаланушымен келісіп отырады, бұл қауіпсіздікті арттырады [4].

Зерттеу процедураны:

1. Кейс-стади: Автор шағын және орта бизнеске арналған финтех-аналитикалық платформа стартапын әзірледі. Тапсырма – 10 күн ішінде толыққанды MVP (minimal viable product) нұсқасын жасау. Процесс барысында код жазу толығымен ЖИ-агенттерге жүктелді. Автордың міндеті – технологиялық стекті (React, Node.js, PostgreSQL) таңдау, жоба архитектурасын жобалау, ЖИ-ге жоғары деңгейлі тапсырмалар (мысалы, «пайдаланушы аутентификациясы үшін JWT негізінде модуль құру») қою және генерацияланған кодтың сапасын тексеру болды.

2. Салыстырмалы талдау: Жобаны әзірлеу барысында Cursor IDE және Claude Code құралдарының тиімділігі әртүрлі тапсырмалар үшін бағаланды. Салыстыру келесі критерийлер бойынша жүргізілді: интерфейсының ыңғайлылығы, автономдылық деңгейі, код сапасы, күрделі тапсырмаларды шешу қабілеті және құны. Мысалы, UI компоненттерін жылдам прототиптеу үшін Cursor қолданылса, күрделі backend логикасын және дерекқор миграциясын құру үшін Claude Code таңдалды.

3. Деректерді жинау: Генерацияланған код жолдарының саны (SLOC - Source Lines of Code) есептелді. Әрбір құралдың орындаған тапсырмалары, жұмсалған уақыт және туындаған қиындықтар (мысалы, ЖИ-дің ескірген кітапханаларды қолдануы немесе қате шешімдер ұсынуы) тіркеліп отырды.

Бұл әдіснама ЖИ-агенттердің нақты жұмыс жағдайындағы мүмкіндіктерін жан-жақты бағалауға және «вайбкодинг» тұжырымдамасының практикалық негіздемесін қалыптастыруға мүмкіндік берді.

Зерттеу нәтижелері және талқылау. Зерттеудің нәтижелері екі

негізгі бөлімге бөлінді: кейс-стадидің сандық және сапалық көрсеткіштері, сондай-ақ Cursor IDE мен Claude Code құралдарын салыстырмалы талдау нәтижелері.

Кейс-стади нәтижелері. Финтех-платформаның MVP нұсқасын 10 күн ішінде әзірлеу бойынша жүргізілген эксперимент келесі нәтижелерді көрсетті:

– Сандық көрсеткіштер: Жалпы көлемі 140 000 жолдан асатын код (SLOC) генерацияланды. Бұл көрсеткішке frontend (React, CSS), backend (Node.js, Express), дерекқормен жұмыс істеу логикасы (SQL сұраныстары, миграциялар) және unit-тесттер кіреді. Дәстүрлі әдіспен мұндай көлемдегі жұмысты орындау үшін, біздің бағалауымызша, 3-4 ай уақыт және 5-7 адамнан тұратын команда қажет болар еді. Ал бұл жобаны бір ғана бағдарламашы-архитектор ЖИ-агенттерді басқару арқылы орындады.

Сапалық көрсеткіштер: Генерацияланған кодтың сапасы жоғары болды, бірақ біркелкі емес. ЖИ стандартты, шаблонды тапсырмаларды (CRUD операциялары, API эндпоинттер, UI компоненттері) мінсіз орындады. Алайда, күрделі бизнес-логиканы немесе оңтайландыруды қажет ететін спецификалық есептеулерде ЖИ ұсынған шешімдер қосымша тексеруді және түзетуді қажет етті. Кодтың шамамен 15-20%-ы автор тарапынан рефакторингтен өтті. Бұл нәтиже ЖИ-дің әлі де адамның стратегиялық бақылауын қажет ететінін, бірақ төмен деңгейлі жұмыстың 80-85%-ын толығымен автоматтандыра алатынын көрсетеді.

Төменде екі құралдың салыстырмалы талдауының негізгі тұстары сипатталған.



Сурет 1. Cursor бойынша статистика

1. Интерфейс және қолдану ыңғайлылығы:

– Cursor IDE: VS Code-қа негізделген интерфейсі арқасында көптеген бағдарламашылар үшін үйреншікті және интуитивті. ЖИ-мен өзара әрекеттесу чат-панель және инлайн-командалар арқылы жүзеге асады, бұл жылдам түзетулер мен прототиптеу үшін өте ыңғайлы. Jupyter Notebooks және Mermaid диаграммаларын визуализациялау мүмкіндігі деректерді талдау міндеттерінде артықшылық береді [2].

– Claude Code: Таза CLI/Терминал интерфейсі бар. Бұл тәжірибелі backend және DevOps инженерлері үшін қуатты құрал болғанымен, жаңадан бастағандар үшін күрделі болуы мүмкін. Интерактивтілігі төмен, бірақ автономдылығы жоғары.

2. Автономдылық және тапсырманы орындау:

1. Cursor IDE: Көбінесе адамның тікелей нұсқауын талап етеді. Background Agent мүмкіндігі болғанымен, әрбір маңызды қадамды пайдаланушы растауы керек. Ол

«жұптық бағдарламашы» (pair programmer) рөлін атқарады.

2. Claude Code: Жоғары деңгейдегі автономдылыққа ие. Ол бір тапсырманы (мысалы, «бүкіл жобадағы ескірген кітапхананы жаңасына ауыстыр және тесттердің өтуін қамтамасыз ет») 7-8 сағат бойы өз бетінше орындай алады [5]. Ол «автономды агент» немесе «виртуалды қызметкер» рөліне жақынырақ.

3. Кодпен жұмыс істеу ерекшеліктері:

– Cursor IDE: Веб-іздеу арқылы ең өзекті құжаттаманы тауып, код генерациялауда тиімді. Мысалы, Ruby кітапханаларының жаңа нұсқаларымен жұмыс істеуде жоғары нәтиже көрсетті. Мультифайлдық өзгерістерді енгізуде Composer функциясы өте қуатты.

– Claude Code: Терең архитектуралық өзгерістер мен масштабты рефакторингтерде күштірек. Оның ойлау процесі күрделірек, сондықтан ол ұсынған шешімдер аз қайта өңдеуді талап етеді (шамамен 30% аз) [4]. Алайда, кейде құжаттаманы дұрыс түсінбей,

кітапхананы қолданудың орнына кастомды HTTP-сұраныстар жасайтын жағдайлар кездесті.

4. Құны және интеграциялар:

– Cursor IDE: Pro нұсқасы айына шамамен \$20 тұрады және 500-ге жуық «баяу» сұраныс лимитін ұсынады. Slack және GitHub-пен интеграциясы командалық жұмыс үшін ыңғайлы.

– Claude Code: Қолдану интенсивтілігіне байланысты. Белсенді пайдаланғанда сағатына ~\$8 құрауы мүмкін, бұл айтарлықтай қымбат. Бірақ GitHub Actions, VS Code/JetBrains плагиндері және кастомды агенттерге арналған SDK ұсынады, бұл оны корпоративтік ортада икемді етеді. Немесе \$200 тұратын бірақ, жылдамдығы жәй лимитсіз тарифі де бар.

Бұл нәтижелер екі құралдың да өз артықшылықтары мен кемшіліктері бар екенін және оларды бір-бірін толықтыратын құралдар ретінде қарастыруға болатынын көрсетеді.

Алынған нәтижелер бағдарламалық қамтамасыз етуді әзірлеу саласындағы терең трансформацияны айғақтайды және «вайбкодинг» тұжырымдамасының өміршеңдігін дәлелдейді. Біздің кейс-стади көрсеткендей, бір адамның ЖИ-агенттерді басқару арқылы қысқа мерзімде үлкен көлемді жұмысты атқаруы Фредерик Брукстың «Мифтік адам-ай» еңбегіндегі классикалық постулаттарға қарсы келеді [6]. Брукс жобаға жаңа адамдарды қосу коммуникациялық шығындарды арттырып, жұмысты тездетудің орнына баяулататынын айтқан. ЖИ-агенттерде мұндай коммуникациялық кедергілер жоқ, бұл олардың командалық жұмыстағы тиімділігін түсіндіреді.

Біздің зерттеуіміз Стэнфорд университетінің ғалымдары жүргізген жұмыстармен үндеседі. Олар ЖИ-

көмекшілерді қолданатын бағдарламашылардың өнімділігі 55%-ға дейін артатынын анықтаған [7]. Алайда, біздің нәтижелер бұл көрсеткіштің әлдеқайда жоғары болуы мүмкін екенін көрсетеді, егер бағдарламашының рөлі код жазудан ЖИ-ді басқаруға толық ауысса. Бұл жағдайда өнімділіктің артуы пайызбен емес, есемен өлшенеді.

«Вайбкодинг» феномені бағдарламашының негізгі құзыретті-лігінің өзгеруін сипаттайды. Ендігі маңызды дағды – бұл синтаксисті білу емес, жүйелік ойлау (systems thinking). Бағдарламашы жобаның барлық компоненттерінің (frontend, backend, дерекқор, инфрақұрылым) қалай өзара әрекеттесетінін түсініп, дұрыс технологиялық стек пен архитектураны таңдауы керек. Мысалы, біздің кейсте React-ты таңдау себебі – оның компоненттік құрылымы ЖИ-ге UI элементтерін генерациялауға оңай тапсырма беруге мүмкіндік береді. Бұл – технологиялық сауаттылықтың маңыздылығын көрсететін мысал. Бұл тұжырым ACM (Association for Computing Machinery) жариялаған соңғы есептермен расталады, онда болашақ бағдарламашыларға абстрактілі ойлау және күрделі жүйелерді жобалау дағдыларын дамытуға кеңес беріледі [8].

Сонымен қатар, ЖИ генерациялаған кодтың сапасы мен қауіпсіздігі туралы мәселені де ескеру қажет. Chen және оның әріптестерінің зерттеуі көрсеткендей, LLM-дер (үлкен тілдік модельдер) кейде қауіпсіздік осалдықтары бар кодты генерациялауы мүмкін [1]. Біздің тәжірибемізде де ЖИ ұсынған кейбір шешімдер қосымша тексеруді қажет етті. Бұл «вайбкодинг» парадигмасында адамның рөлі тек жоғары деңгейлі нұсқау берумен шектелмейтінін, сонымен қатар сапаны бақылау, кодты аудиттеу және соңғы шешімді қабылдау

жауапкершілігін де қамтитынын білдіреді. Бағдарламашы – ЖИ-дің «ақылды супервайзері».

Cursor мен Claude Code-тың архитектуралық айырмашылықтары (IDE-центристік және агенттік-терминалды) [9, 10] әртүрлі типтегі бағдарламашылар мен тапсырмалар үшін оңтайлы құралды таңдаудың маңыздылығын көрсетеді. Интерактивтілікті қажет ететін фронтенд-әзірлеушілер үшін Cursor тиімді болса, ұзақ, автономды процестерді қажет ететін инфра-құрылымдық инженерлер үшін Claude Code артықшылыққа ие. Демек, болашақтың бағдарламашысы тек технологияларды ғана емес, сонымен қатар оларды басқаратын ЖИ-құралдардың ерекшеліктерін де терең білуі тиіс.

Қорытынды. Бұл зерттеу жасанды интеллект құралдарының бағдарламалау саласына әкелген өзгерістерін талдап, «вайбкодинг» тұжырымдамасын ұсынды. Алынған нәтижелер мен талқылаулар негізінде келесідей негізгі қорытындылар жасауға болады:

1. Парадигманың ауысуы: Бағдарламалық қамтамасыз етуді әзірлеу процесі төмен деңгейлі код жазудан жоғары деңгейлі, стратегиялық басқаруға – «вайбкодингке» ауысуда.

2. Бағдарламашының жаңа рөлі: Заманауи бағдарламашы – бұл код жазушы емес, ЖИ-агенттерді басқаратын, технологиялық шешімдер

қабылдайтын және жүйе архитектурасын жобалайтын технологстратег.

3. Өнімділіктің серпілісі: Cursor IDE және Claude Code сияқты ЖИ-агенттерді тиімді қолдану өнімділікті дәстүрлі әдістермен салыстырғанда бірнеше есе арттыруға мүмкіндік береді, бұл біздің кейс-стадидегі 10 күнде 140 000 жол код генерациялау арқылы дәлелденді.

4. Технологиялық сауаттылықтың маңызы: Бағдарламашының нарықтағы құндылығы оның нақты бір тілдің синтаксисін білуімен емес, әртүрлі технологиялардың (фреймворктер, кітапханалар, бұлттық сервистер) артықшылықтары мен кемшіліктерін терең түсініп, оларды жоба мақсатына сай дұрыс таңдай білу қабілетімен анықталады.

5. Құралдарды таңдаудың маңыздылығы: Әртүрлі ЖИ-агенттердің (мысалы, IDE-интеграцияланған Cursor және автономды Claude Code) өз ерекшеліктері бар. Тапсырманың сипатына қарай оңтайлы құралды таңдау – табыстың маңызды факторы.

6. Адамның бақылауының сақталуы: ЖИ-дің барлық артықшылықтарына қарамастан, адамның стратегиялық бақылауы, сапаны тексеру және соңғы шешімді қабылдау рөлі сақталады. ЖИ – қуатты құрал, бірақ ол әлі де тәжірибелі маманның басшылығын қажет етеді.

Әдебиеттер тізімі

1. Chen, M., et al. Evaluating Large Language Models Trained on Code. arXiv. 2021. – [Электрондық ресурс]. – Қол жеткізу режимі. DOI: 10.48550/arXiv.2107.03374.
2. Cursor. Cursor Changelog Version 1.2. 2025. – [Электрондық ресурс]. – Қол жеткізу режимі: Retrieved from <https://cursor.com/changelog>
3. Sweetpad. A Look at Cursor's System Prompts. 2024. – [Электрондық ресурс]. – Қол жеткізу режимі: Retrieved from <https://sweetpad.io/blog/a-look-at-cursors-system-prompts>.
4. Anthropic. [Электрондық ресурс]. – Қол жеткізу режимі: Claude 4 Technical Report. Anthropic PBC. – 2025.
5. Morin, A. I Let an AI Agent Code for 7 Hours. Here's What Happened. Towards Data Science. – 2024.

6. Brooks, F. P. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Addison-Wesley Professional. – 1995.
7. Noy, S., & Zhang, W. Experimental evidence on the productivity effects of generative artificial intelligence. Stanford University & MIT. – 2023
8. ACM Task Force on Curricula. Computing Curricula 2023: A Vision for the Future of Computing Education. Association for Computing Machinery. – 2023
9. OpenAI. 2023 – GPT-4 Technical Report. arXiv. [Электрондық ресурс]. – Қол жеткізу режимі: doi:10.48550/arXiv.2303.08774
10. Google. [Электрондық ресурс]. – Қол жеткізу режимі: Gemini: A Family of Highly Capable Multimodal Models. Google AI. – 2023

Б.М.Жақсыбаев¹, Ж.К.Тасжурекова², М.А.Ахметжанов³

Таразский университет имени М.Х.Дулати, г.Тараз, Казахстан

ПРОГРАММИРОВАНИЕ В ЭПОХУ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА: ФЕНОМЕН «ВАЙБКОДИНГА» И ПРЕИМУЩЕСТВО ТЕХНОЛОГИЧЕСКОЙ ГРАМОТНОСТИ

Аннотация. В данной статье анализируется влияние современных инструментов искусственного интеллекта (ИИ), таких как Cursor IDE и Claude Code, на сферу разработки программного обеспечения. Автор вводит понятие «вайбкодинг» – процесс, при котором программист направляет ИИ посредством высокоуровневых инструкций, проектирования архитектуры и выбора технологического стека, вместо написания низкоуровневого кода. Представлен кейс-стади, основанный на опыте создания собственного стартапа, где за 10 дней было сгенерировано 140 000 строк кода исключительно с помощью ИИ, что демонстрирует возможность замены работы большой команды при традиционном подходе. Ключевой вывод исследования заключается в том, что ценность программиста сегодня определяется не скоростью написания кода, а его способностью глубоко понимать технологии, правильно их выбирать и эффективно управлять ИИ-агентами.

Ключевые слова: искусственный интеллект, генерация кода, программная инженерия, Cursor, Claude Code, вайбкодинг.

B. M. Zhaksybayev¹, Zh. K. Tashurekova², M. A. Akhmetzhanov³

M. Kh. Dulaty Taraz University, Taraz, Republic of Kazakhstan

PROGRAMMING IN THE ERA OF ARTIFICIAL INTELLIGENCE: THE PHENOMENON OF "VIBECODING" AND THE ADVANTAGE OF TECHNOLOGICAL LITERACY

Abstract. This article analyzes the impact of modern artificial intelligence (AI) tools, such as Cursor IDE and Claude Code, on the software development field. The author introduces the concept of "vibecoding," a process where a programmer directs AI through high-level instructions, architectural design, and technology stack selection, rather than writing low-level code. A case study is presented based on the experience of developing a personal startup, where 140,000 lines of code were generated in 10 days solely using AI, demonstrating the potential to replace the work of a large team under traditional approaches. The key conclusion of the research is that the value of a programmer today is determined not by the speed of writing code, but by the ability to deeply understand technologies, select them correctly, and effectively manage AI agents.

Keywords: artificial intelligence, code generation, software engineering, Cursor, Claude Code, vibecoding.

References

1. Chen, M., et al. Evaluating Large Language Models Trained on Code. arXiv. – 2021 [Electronic resource]. DOI: 10.48550/arXiv.2107.03374.
2. Cursor. Cursor Changelog Version 1.2. – 2025 [Electronic resource]. Retrieved from <https://cursor.com/changelog>
3. Sweetpad. A Look at Cursor's System Prompts. – 2024 [Electronic resource]. Retrieved from <https://sweetpad.io/blog/a-look-at-cursors-system-prompts>.
4. Anthropic. Claude 4 Technical Report. Anthropic PBC. – 2025.
5. Morin, A. I Let an AI Agent Code for 7 Hours. Here's What Happened. Towards Data Science. – 2024.
6. Brooks, F. P. The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition. Addison-Wesley Professional. – 1995.
7. Noy, S., & Zhang, W. Experimental evidence on the productivity effects of generative artificial intelligence. Stanford University & MIT. – 2023.
8. ACM Task Force on Curricula. Computing Curricula 2023: A Vision for the Future of Computing Education. Association for Computing Machinery. – 2023
9. OpenAI. 2023 – GPT-4 Technical Report. arXiv. [Electronic resource]. DOI: 10.48550/arXiv.2303.08774.
10. Google. Gemini: A Family of Highly Capable Multimodal Models. Google AI. – 2023

18.07.2025 ж. баспаға түсті

29.08.2025 ж. түзетулермен түсті.

30.09.2025 ж. басып шығаруға қабылданды

Мақалаға сілтеме:

Жақсыбаев, Б.М. Жасанды интеллект дәуіріндегі бағдарламалау: «Вайбкодинг» феномені және технологиялық сауаттылықтың артықшылығы [Мәтін] / Б.М. Жақсыбаев, Ж.К.Тасжурекова, М.А. Ахметжанов // Dulary University Хабаршысы. – 2025. – №3. – Б. 56 - 63 <https://doi.org/10.55956/VTRK6250>



Copyright: © 2024 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY NC) license (<https://creativecommons.org/licenses/by-nc/4.0/>).