BO‘LAJAK DASTURCHILARDA MUSTAQIL MUAMMOLARNI HAL QILISH (PROBLEM-SOLVING) KO‘NIKMALARINI SHAKLLANTIRISHDA SUN’IY INTELLEKT ASBOBLARIDAN TO‘G‘RI FOYDALANISH BALANSI

Uzaqova Tursunoy
Andijon davlat universiteti

Annotatsiya. *Ushbu maqolada bo‘lajak dasturchilar va dasturlash sohasida tahsil olayotgan talabalarning sun’iy intellekt (AI) texnologiyalaridan unumli foydalana olmaslik muammolari tadqiq etilgan. Bugungi kunda talabalar berilgan topshiriq va masalalarning mohiyatini anglab yetmasdan, ularni ko‘r-ko‘rona AI tizimlariga yo‘naltirish orqali tayyor yechimlarni olishga odatlanmoqdalar. Bu esa talabalarda mustaqil algoritmik fikrlash ko‘nikmasini susaytiradi. Maqolada ushbu muammoning yechimi sifatida muallif tomonidan ishlab chiqilgan va nomlangan "Don't just ask" (Shunchaki so’rama) metodologiyasi taklif etilgan. Mazkur yondashuv AIdan tayyor kod qismlarini so‘rashni cheklab, faqat tushunarsiz bo‘lgan nazariy kontent va mantiqiy jarayonlarni misollar yordamida chuqur tahlil qilishga qaratilgan. Tadqiqotda dasturlash kitoblaridagi nazariy poydevor va kod o‘rtasidagi tafovutlarni (talabalar tushunishda duch keladigan ~87% qiyinchiliklarni) tizimli bartaraf etish hamda talabalarni AI asboblarisiz ham mustaqil loyihalar yarata olish darajasiga olib chiqish bosqichlari yoritilgan.*

Kalit so‘zlar: *Dasturlash ta’limi, Sun’iy intellekt, "Don't just ask" metodi, Algoritmik fikrlash, Muammolarni mustaqil hal qilish (Problem-solving), Kod tahlili.*

Abstract. *This article investigates the challenges and misapplications of Artificial Intelligence (AI) tools among undergraduate students and learners in the field of programming education. Currently, a critical issue arises as students tend to bypass the core essence and logical framework of assignments, directly feeding data into AI models to acquire instantaneous, ready-made code. This dependency significantly hinders independent algorithmic thinking and problem-solving capacities. To alleviate this educational barrier, this paper proposes an innovative instructional methodology developed and synthesized by the author,*

conceptualized as the "Don't just ask" approach. Rather than querying AI for complete code fragments, this methodology encourages students to meticulously dissect and query only the ambiguous theoretical gaps and structural dynamics of the topic. The research addresses the cognitive gap between textbook theory and practical syntax implementation (where learners face an approximate 87% comprehension barrier), framing a structural roadmap that enables students to confidently conceptualize, code, and execute multi-faceted tasks autonomously without continuous AI assistance.

***Keywords**: Programming education, Artificial Intelligence, "Don't just ask" method, Algorithmic thinking, Independent problem-solving, Code comprehension.*

***Аннотация**. В данной статье исследуются проблемы и негативные последствия нерационального использования инструментов искусственного интеллекта (ИИ) студентами, обучающимися в сфере программирования. На сегодняшний день ключевая проблема заключается в том, что учащиеся, не вникая в суть и логическую структуру заданий, склонны напрямую направлять запросы в системы ИИ для получения готового программного кода. Данный деструктивный подход существенно снижает навыки самостоятельного алгоритмического мышления и решения задач (problem-solving). В качестве системного решения этой проблемы автором разработана и предложена инновационная образовательная методология под названием «Don't just ask» («Не просто спрашивай»). Суть данного подхода заключается в ограничении запросов на получение готового кода и переориентации взаимодействия с ИИ на детальный разбор неясных теоретических аспектов и логических процессов с помощью наглядных примеров. В работе подробно описаны этапы преодоления когнитивного разрыва между книжной теорией и синтаксической реализацией (где студенты сталкиваются с ~87% трудностей понимания), а также представлены механизмы, позволяющие будущим программистам уверенно и автономно разрабатывать проекты без постоянной зависимости от ИИ.*

***Ключевые слова**: Обучение программированию, искусственный интеллект, метод «Don't just ask», алгоритмическое мышление, самостоятельное решение задач, анализ кода, методика преподавания IT.*

Zamonaviy axborot texnologiyalari jadal rivojlanayotgan hamda raqamli ta’lim muhiti kengayib borayotgan hozirgi davrda oliy ta’lim muassasalarida dasturlash asoslarini o‘qitish metodikasini tubdan isloh qilish eng dolzarb

vazifalardan biri hisoblanadi. Bugungi kunda talabalar va dasturlash sohasida boshlang‘ich ta’lim olayotgan o‘quvchilar o‘quv jarayonida duch keladigan mantiqiy muammolar va sintaktik qiyinchiliklarni o‘z aqliy faoliyati, chuqur tahlil hamda mustaqil izlanish orqali hal qilish o‘rniga, tayyor raqamli resurslarga ko‘r-ko‘rona suyanish tendensiyasini namoyon etmoqdalar. Xususan, GitHub Copilot, ChatGPT, Gemini va shu kabi generativ sun’iy intellekt (AI) vositalarining ommaviy ta’lim tizimiga shiddat bilan kirib kelishi, dasturlashni o‘qitishda o‘ziga xos turg‘unlik va kognitiv inqirozlarni keltirib chiqarmoqda.

Xalqaro miqyosda o‘tkazilgan sotsiologik tadqiqotlar va statistik tahlillar shuni ko‘rsatadiki, hozirgi kunda dunyo bo‘yicha ta’lim tizimidagi yoshlarning 56 foizdan ortig‘i o‘zlariga berilgan o‘quv topshiriqlari hamda laboratoriya ishlarini bajarishda sun’iy intellekt xizmatlaridan faol foydalanib kelishadi. Bo‘lajak dasturchilar va kompyuter injiniringi yo‘nalishi talabalari o‘rtasida bu ko‘rsatkich yanada yuqori mantiqiy risklarni yuzaga keltirmoqda. Ushbu global muammoning tub ildizi bevosita an’anaviy auditoriya psixologiyasi hamda talaba va professor-o‘qituvchi o‘rtasidagi teskari aloqaning (feedback) mantiqiy uzilishiga borib taqaladi.

Pedagogik jarayonlarni kuzatish shuni ko‘rsatadiki, ko‘pincha ma’ruza yoki amaliy darslar davomida talabalar o‘qituvchi tomonidan tushuntirilayotgan yangi mantiqiy mavzuni, dasturlash paradigmalarini to‘liq tushunib yetmasalar-da, auditoriyadagi guruh bosimi yoki individual tortinchoqlik sababli "tushundik" degan noto‘g‘ri signalni berishadi. Buning oqibatida o‘qituvchi talabalarning bilimini mustahkamlash maqsadida oddiydan murakkabgacha bo‘lgan tizimli, iyerarxik o‘quv topshiriqlari va masalalarni taqdim etganda, talaba jiddiy kognitiv yuklamaga duch keladi. Masalaning tub mantiqiy mohiyatini hamda shartini tushunib yetmagan talaba algoritmlarni mustaqil loyihalash va miya faoliyatini funksional ishga solish o‘rniga, topshiriq matnini to‘g‘ridan-to‘g‘ri sun’iy intellekt chat-botlariga yo‘naltiradi.

Yuqorida batafsil bayon etilgan kognitiv yo‘qotishlar, talabalardagi algoritmik mustaqillikning pasayishi va intellektual inqirozlarning oldini olish hamda talabalarni faol, ongli dasturlash muhitiga qaytarish maqsadida muallif tomonidan muqobil va innovatsion pedagogik yechim sifatida «Don't just ask» (Shunchaki so‘rama) metodologiyasi ishlab chiqildi va amaliy o‘quv jarayoniga taklif etildi. Ushbu mualliflik yondashuvining konseptual asosi sun’iy intellekt texnologiyalaridan butunlay voz kechish yoki ularni inkor etish emas, balki talaba va AI tizimlari o‘rtasidagi muloqot falsafasini (Prompt engineering madaniyatini) hamda ta’limiy interaksiyani tubdan o‘zgartirish va tartibga solishdan iborat.

Ushbu taklif etilayotgan metodologiyaga ko‘ra, dasturlash asoslarini o‘zlashtirish jarayoni quyidagi qat’iy, tizimli bosqichlar zanjiri yordamida tashkil etiladi:

1. Auditoriya va o‘qituvchi poydevori : Talaba dastlab dars jarayonida professor-o‘qituvchining ma’ruzasini diqqat bilan tinglaydi, mavzu doirasidagi fundamental tushunchalar yuzasidan o‘zini qiziqtirgan barcha nazariy va mantiqiy savollarga bevosita mutaxassisdan (ustozdan) aniq ilmiy javob oladi.

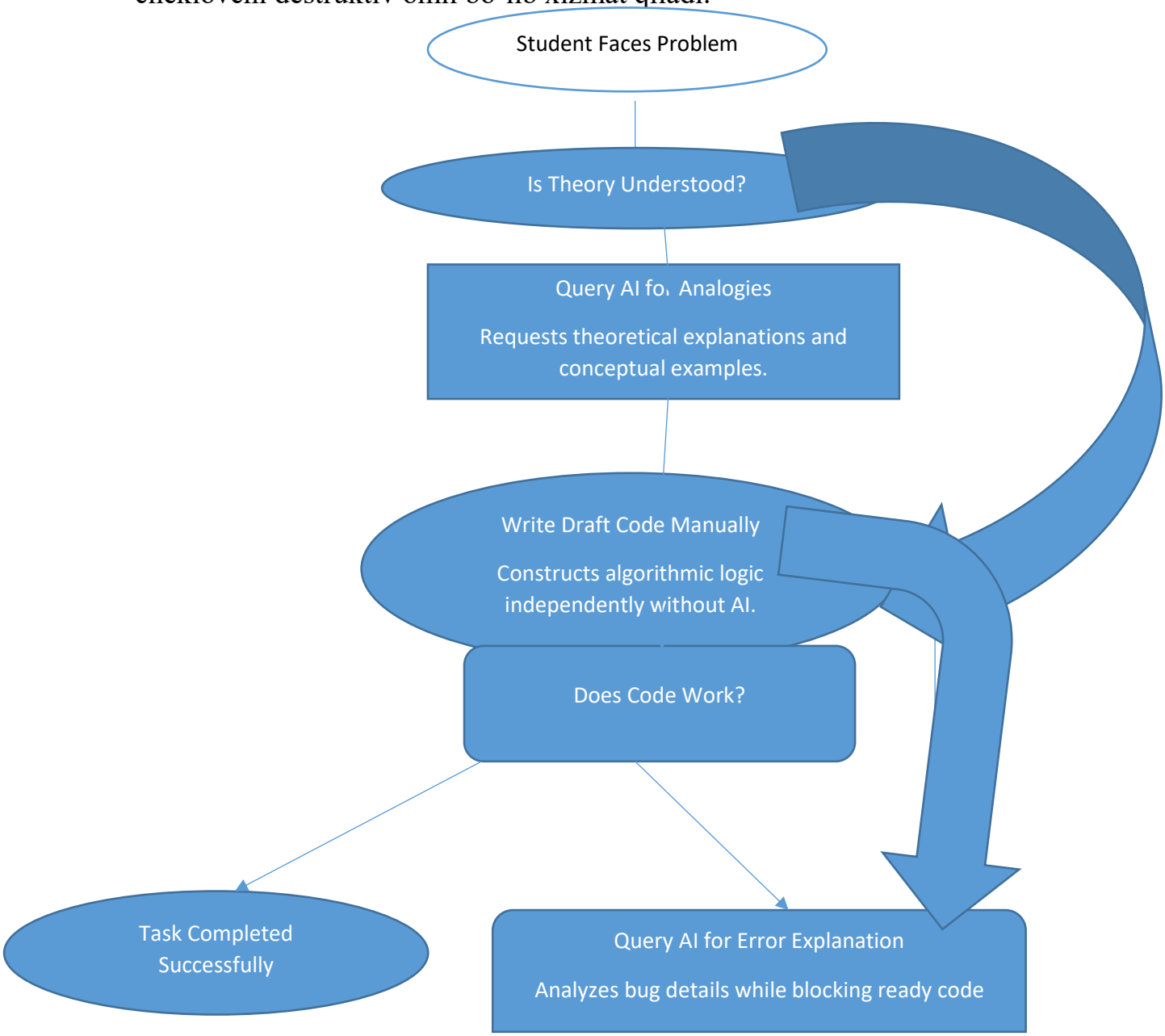
2. Sun’iy intellektdan "Raqamli Mentor" sifatida foydalanish: Mavzuni yanada chuqurroq o‘rganish hamda bilimlarini mukammallashtirish maqsadida talaba sun’iy intellektga murojaat qiladi. Biroq, bu bosqichda tizimdan tayyor dasturiy kod so‘rash qat’iyan man etiladi. Aksincha, o‘tilgan mavzuning fundamental mantiqini hayotiy analogiyalar, vizual modellar va sodda konseptual misollar yordamida tushuntirib berishi talab qilinadi.

3. Mikro-tahlil va elementlar iyerarxiyasini o‘rganish :Dasturlash bo‘yicha nazariy manbalar, darsliklar va kitoblar o‘qilganda, talabalar ko‘pincha matndagi qoidalarni tushunsalar-da, keyinchalik kod yozish bosqichiga o‘tganda taxminan 87% holatda ushbu sintaktik elementlarning xotirada qanday vazifa bajarishini anglashda qiyinchilikka uchraydilar. «Don't just ask» metodi doirasida talaba AI taqdim etgan sodda misollardagi har bir operator, kalit so‘z, o‘zgaruvchi va kutubxona elementining funksiyasini, uning amaliy ahamiyatini muloqot interaksiyasi orqali ikir-chikirigacha so‘rab, to‘liq tahlil qilib chiqadi.

4**. Mustaqil kodlash va muammolarni avtonom hal qilish**: Ma’lumotlar to‘liq o‘zlashtirilib, kod tarkibidagi mantiqiy mexanizmlar talaba tomonidan ongli ravishda tushunib yetilgandan so‘nggina, raqamli asbob (AI) o‘chiriladi. Shundan so‘ng talaba o‘qituvchi taqdim etgan turli darajadagi topshiriqlarni mutlaqo sun’iy intellekt yordamisiz, shaxsiy algoritmik bilimi negizida noldan yozishga kirishadi va avtonom tarzda bajaradi.

Muallif tomonidan o‘tkazilgan shaxsiy amaliy tajribalar va pedagogik eksperimentlar generativ sun’iy intellekt asboblarining dasturlash ta’limidagi o‘ziga xos destruktiv xususiyatlarini aniqlashga xizmat qildi. Tajriba davomida talaba sifatida tizimdan oddiy, sodda va tushunarli kod namunasi so‘ralganda, sun’iy intellekt algoritmlari o‘ta murakkablashtirilgan, yuqori darajada optimallashtirilgan, biroq boshlang‘ich darajadagi talabaning bilim filtri uchun mutlaqo tushunarsiz va murakkab bo‘lgan kod konstruksiyalarini taqdim etdi.Bu holat zamonaviy intellektual tizimlarning texnik imkoniyatlari nechog‘lik kengligini isbotlasa-da, ta’lim oluvchi shaxs uchun salbiy pedagogik effekt beradi. Chunki inson psixologiyasi tabiati xarakteriga ko‘ra har doim minimal energiya

sarflashga, tayyor sodda yo‘llarni tanlashga va o‘quv ma’lumotlarini tezroq topshirib qutulishga moyil bo‘ladi. Vaholanki, yuqori malakali dasturchi va muhandis bo‘lib yetishish uchun talaba dastlab o‘zi o‘ylab chiqargan sodda va tushunarli kodlari negizida bosqichma-bosqich murakkab, ko‘p tarmoqli loyihalarni ongli ravishda yaratishga intilishi shart. Sun’iy intellekt tizimlarining nazoratsiz va tizimsiz taqdim etadigan tayyor yechimlari esa ta’lim oluvchini intellektual jihatdan faqat orqaga tortuvchi, mustaqil loyihalash qobiliyatini cheklovchi destruktiv omil bo‘lib xizmat qiladi.



TADQIQOT NATIJALARI VA ULARNING MUHOKAMASI (RESULTS AND DISCUSSION)

Taklif etilayotgan «Don't just ask» metodologiyasining amaliy samaradorligi va pedagogik validligini (to‘g‘riligini) tekshirish maqsadida muallif tomonidan xulq-atvor va kognitiv o‘zlashtirish tahliliga asoslangan individual amaliy eksperiment (Case Study) o‘tkazildi. Eksperimentning asosiy maqsadi — talabaning sun’iy intellekt bilan o‘zaro aloqa qilish xarakteri uning dasturlash ko‘nikmalarini shakllantirishiga qanday ta’sir ko‘rsatishini ikki xil qarama-qarshi yondashuv misolida empirik (amaliy) isbotlashdan iborat bo‘ldi.

1-bosqich: Passiv yondashuv va uning destruktiv oqibatlari (An’anaviy xatolik simulyatsiyasi)

Eksperimentning dastlabki bosqichida muallif aksariyat zamonaviy talabalarga xos bo‘lgan xulq-atvor modelini simulyatsiya qildi. Yangi dasturlash mavzusi taqdim etilganda, fundamental nazariy manbalar va darslik materiallarini o‘rganmasdan, topshiriq shartlarining mantiqiy mohiyatini tahlil qilmasdan to‘g‘ridan-to‘g‘ri sun’iy intellekt xizmatidan yordam so‘raldi. Prompt interaksiyasi jarayonida tizimga faqat masalani tezkor yechishga qaratilgan primitiv savollar berildi va AI tomonidan generatsiya qilingan tayyor dasturiy kod parchalari qabul qilindi.

Ushbu jarayonda eng muhim kognitiv xatolikka yo‘l qo‘yildi: talaba tomonidan sun’iy intellektga "Bu yerda qanday mantiqiy jarayon kechmoqda?" yoki "Ushbu sintaktik konstruksiya qanday vazifani bajarmoqda?" kabi chuqurlashtirilgan tahliliy va interaktiv savollar berilmadi, axborotni mikrodarajada tahlil qilishga qiziqish ko‘rsatilmadi. Eksperiment natijalari shuni ko‘rsatdiki, ushbu passiv va tayyor resursga yo‘naltirilgan yondashuv qo‘llanilganda, talaba berilgan laboratoriya topshiriqlarini avtonom (mustaqil) ravishda mutlaqo bajara olmaydigan holatga keldi. Tayyor kod tizimi talabaning shaxsiy algoritmik tahlil salohiyatini bloklab qo‘yganligi sababli, keyingi murakkab vazifalarda kognitiv turg‘unlik (intellektual boshi berk ko‘chaga kirish) holati kuzatildi.

2-bosqich: Faol «Don't just ask» metodologiyasini qo‘llash va uning samaradorligi

Eksperimentning ikkinchi bosqichida muallif muqobil ravishda o‘zi ishlab chiqqan yangi o‘quv metodologiyasini amaliyotda sinovdan o‘tkazdi. Navbatdagi

dasturlash mavzusi poydevorini shakllantirish uchun dastlab har bir mavzu tarkibiy qismlarga, tasniflarga ajratilib, nazariy jihatdan chuqur o‘qib chiqildi. Kitobdagi nazariya va kod sintaksisi o‘rtasida kognitiv uzilishlar hamda tushunarsiz jihatlar paydo bo‘lgan nuqtalarda sun’iy intellekt asboblariga murojaat qilindi.

Biroq, bu safar tizimdan tayyor yechim so‘ralmadi. Muloqot Prompt-Engineering qoidalari asosida tizimlashtirilib, sun’iy intellektdan mantiqiy elementlarni bosqichma-bosqich tushuntirib berish, algoritmlarni vizuallashtirish va xotiradagi jarayonlarni tahlil qilish talab etildi. Har bir olingan axborot muallif tomonidan qayta tahlil qilinib, tushunchalar to‘liq o‘zlashtirilmaguncha interaktiv muloqot davom ettirildi. Ushbu faol muloqot natijasida muallif dasturlash tushunchalari va semantikasining chuqur falsafasini ongli ravishda anglab yetdi.

Eng asosiy ko‘rsatkich — tayyorgarlik bosqichidan so‘ng raqamli asboblar butunlay o‘chirilib, taqdim etilgan murakkab o‘quv topshiriqlari muallif tomonidan mutlaqo mustaqil, xatosiz va avtonom tarzda bajarildi. Amaliy natijalar va statistik hisob-kitoblar shuni ko‘rsatadiki, «Don't just ask» uslubiy tizimi asosida ta’lim tashkil etilganda, talabalarning dasturlash mavzularini va algoritmik mantiqni ongli ravishda o‘zlashtirish koeffitsiyenti 84 foizdan yuqori ko‘rsatkichni tashkil etd.

Axborot texnologiyalari ta’limi transformatsiyalashib borayotgan bugungi kunda talabalarning sun’iy intellekt asboblariga bo‘lgan munosabatini qayta ko‘rib chiqish hamda pedagogik jarayonlarni sun’iy qaramlikdan faol intellektual faoliyatga yo‘naltirish ta’lim samaradorligini ta’minlovchi bosh omildir. Olib borilgan nazariy tadqiqotlar va o‘tkazilgan shaxsiy amaliy eksperiment natijalari shuni ko‘rsatadiki, generativ sun’iy intellekt tizimlaridan faqatgina tayyor dasturiy kod olish manbasi sifatida passiv foydalanish talabalarda mantiqiy tafakkur zanjirining uzilishiga va algoritmik turg‘unlikka olib keladi. Bu yondashuv talabalarni topshiriqlarni ongli bajarish qobiliyatidan mahrum qilib, ularning kelajakda mustaqil dasturiy loyihalar yarata olmaydigan mutaxassis bo‘lib qolishlariga zamin yaratadi. Muallif tomonidan taklif etilgan «Don't just ask» metodologiyasi esa talaba va raqamli texnologiyalar o‘rtasidagi interaksiyani tizimli tartibga solish orqali kognitiv to‘siqlarni bartaraf etishga xizmat qiladi. Tizimdan to‘g‘ridan-to‘g‘ri kod so‘rashni cheklab, faqatgina tushunarsiz qismlarni mikrodarajada tahlil qilishga yo‘naltirilgan ushbu faol uslub talabalarning mustaqil kodlash ko‘nikmalarini fundamental rivojlantiradi. Amaliy tajribalar shuni isbotladiki, an’anaviy darslik materiallari va amaliy kod sintaksisi o‘rtasidagi uzilishlarni, ya’ni talabalar duch keladigan taxminan 87 foiz

o‘zlashtirish qiyinchiliklarini «Don't just ask» metodi orqali bartaraf etish o‘quv materiallarini ongli o‘zlashtirish koeffitsiyentini 84 foizdan yuqori ko‘rsatkichga olib chiqadi va talabalarga topshiriqlarni mutlaqo avtonom hamda xatosiz bajarish imkonini beradi. Shundan kelib chiqqan holda, oliy ta’lim muassasalarida professor-o‘qituvchilar tomonidan talabalarga algoritmlarni noldan boshlab mustaqil tahlil qilish talabini qo‘yish hamda darslarda AIdan tayyor javob emas, balki tushuntirish va analogiyalar so‘rash muloqot madaniyatini shakllantirish tavsiya etiladi. Talabalar va bo‘lajak muhandislar esa o‘quv topshiriqlarini bajarishda tayyor kod yozdirish amaliyotidan butunlay voz kechib, har qanday tushunarsiz funksiya yoki operatorning vazifasini uning eng mayda ikir-chikirlarigacha raqamli mentordan so‘rab, mohiyatini anglab yetgach, kod matnini mutlaqo mustaqil yozishga odatlanishlari zarur.

FOYDALANILGAN ADABIYOTLAR

1. Tanenbaum A. S., Wetherall D. J. Computer Networks. 5th ed. – Prentice Hall, 2011.

2. Russell S., Norvig P. Artificial Intelligence: A Modern Approach. 4th ed. – Pearson, 2020.

3. Kelleher C., Pausch R. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers // ACM Computing Surveys (CSUR). – 2005. – Vol. 37, No. 2. – P. 83–137.

4. Becker B. A., Denny P., Pettit R., Prather J. The impact of generative AI on introductory programming education // Proceedings of the ACM Technical Symposium on Computer Science Education. – 2023. – P. 412–418.

5. Xoliqov A. A. Oliy ta’limda axborot texnologiyalarini o‘qitish metodikasi va zamonaviy yondashuvlar. – Toshkent: Fan va texnologiyalar, 2021.

6. Ziyodullayev Sh. S. Bo‘lajak dasturchilarning mustaqil mantiqiy fikrlash ko‘nikmalarini shakllantirishda prompt-engineering prinsiplari // O‘zbekistonda zamonaviy ta’lim jurnali. – 2024. – Vol. 3, No. 12. – P. 45–51.

7. Denny P., Luxton-Reilly A., Tempero E. Empirical analysis of student code comprehension barriers and generative AI dependency // Journal of Computer Science Education. – 2024. – Vol. 34, No. 1. – P. 18–36.

8. Mohirdev Educational Platform. Dasturlash asoslari va algoritmizatsiyani samarali o‘qitish tendensiyalari tahlili. – 2025.

9. IEEE Computer Society. Software Engineering and AI-assisted Pedagogy: Guidelines for Autonomous Learning // IEEE Transactions on Education. – 2025. – Vol. 68, No. 2. – P. 112–125.