

UDK: 378.14:004.42

SUN'IY INTELLEKT DAVRIDA DASTURLASHNI O'QITISHNING ZAMONAVIY YONDASHUVLARI VA AMALIY METODLARI

O'rta Chirchiq tuman 3-son texnikumi

Maxsus fan o'qituvchisi: Jumayev Og'abek Zavqi o'g'li

ANNOTATSIYA

Ushbu maqolada sun'iy intellekt (SI) texnologiyalarining jadal rivojlanayotgan davrida texnikum va kasb-hunar ta'lim muassasalarida dasturlash fanlarini o'qitishning zamonaviy metodlari va pedagogik yondashuvlari keng ko'lamda tahlil qilinadi. Maqolada loyiha asosidagi ta'lim (Project-Based Learning), teskari sinf modeli (Flipped Classroom), gamifikatsiya, juftlikda dasturlash (Pair Programming), shuningdek ChatGPT, GitHub Copilot va Replit AI kabi sun'iy intellekt vositalarini dars jarayoniga muvaffaqiyatli integratsiyalash yo'llari ko'rib chiqiladi. Bundan tashqari, maqolada o'qituvchi kompetentsiyasini rivojlantirish, o'quv dasturlarini zamonaviy IT bozori talablariga moslashtirish va o'quvchi motivatsiyasini oshirishning ilmiy asoslangan usullari taqdim etiladi. Xalqaro va mahalliy tadqiqot natijalari tahlili asosida amaliy tavsiyalar ishlab chiqilgan.

Kalit so'zlar: sun'iy intellekt, dasturlash o'qitish, PBL, gamifikatsiya, GitHub Copilot, ChatGPT, texnikum ta'limi, raqamli transformatsiya, o'qituvchi kompetentsiyasi, o'quv dasturi.

1. KIRISH

XXI asrning ikkinchi o'n yilligida sun'iy intellekt texnologiyalari butun dunyo ta'lim tizimini tubdan o'zgartirmoqda. O'zbekiston Respublikasida "Raqamli O'zbekiston – 2030" strategiyasi doirasida IT ta'limiga alohida e'tibor qaratilayotgan bir paytda, texnikum va kasb-hunar ta'lim muassasalarida dasturlash fanlarini o'qitish

usullari ham yangilanishni talab etmoqda. 2023–2024 o'quv yilida o'tkazilgan so'rovnoma shuni ko'rsatdiki, dasturlash fanlarida o'qiydigan o'quvchilarning 67 foizi darslarni "yetarlicha amaliy emas" deb baholagan, 43 foizi esa o'rgangan narsalarini real hayotda qanday qo'llashni bilmasligini ta'kidlagan.

Bugungi kunda ChatGPT, GitHub Copilot va boshqa AI yordamchi vositalari dasturchilik sohasiga chuqur kirib keldi. Bu vositalar bir tomondan dasturlashni o'rganishni yangicha shaklda ta'minlasa, ikkinchi tomondan an'anaviy o'qitish metodologiyasiga jiddiy savol tug'dirmoqda: agar sun'iy intellekt kod yoza olsa, biz o'quvchilarimizga nima o'rgatishimiz kerak va qanday o'rgatishimiz kerak?

Ushbu maqola ana shu muhim savolga ilmiy va amaliy asosda javob topishga harakat qiladi. Maqolaning asosiy maqsadi – zamonaviy texnologiyalar bilan hamqadam, o'quvchilar uchun samarali, qiziqarli va kelajakka yo'naltirilgan dasturlash ta'limini amalga oshirishning kompleks metodologiyasini taqdim etishdir.

2. ILMIY ADABIYOTLAR TAHLILI

Dasturlash ta'limida zamonaviy yondashuvlar masalasi xalqaro ilmiy hamjamiyatda faol muhokama qilinmoqda. Wing (2006) tomonidan ilgari surilgan "computational thinking" (hisoblash tafakkuri) tushunchasi bugun ham ta'lim tizimlarining asosiy orientiri hisoblanadi. Muallif ta'kidlaydi: dasturlashni o'qitishning eng muhim natijasi – bu muammoni tizimli, mantiqiy va ijodiy yechish qobiliyatidir, nafaqat sintaksisni bilish.

Bransford, Brown va Cocking (2000) o'zlarining "How People Learn" asarida ta'kidlashicha, samarali ta'lim uchun uch shart zarur: oldingi bilimlarni faollashtirish, chuqur tushunishga yo'naltirish va bilimni metakognitiv nazorat ostida shakllantirish. Aynan shu tamoyillar zamonaviy dasturlash ta'limi metodologiyasining asosini tashkil etadi.

Freeman va hammuallimlar (2014) o'tkazgan meta-tahlilda 225 ta tadqiqot natijalari ko'rib chiqilib, faol ta'lim usullari passiv ma'ruza uslubiga nisbatan imtihon natijalarini o'rtacha 6% ga oshirishi, muvaffaqiyatsizlik darajasini esa 1.5 baravar

kamaytirishi aniqlangan. Bu natijalar ayniqsa STEM fanlarida, jumladan dasturlashda yaqqol namoyon bo'ladi.

O'zbekistonda IT ta'limini rivojlantirish masalasida ham bir qator tadqiqotlar amalga oshirilmoqda. Toshkent Axborot Texnologiyalari Universiteti olimlari (2023) o'tkazgan so'rovnoma shuni ko'rsatdiki, respublika texnikumlarida dasturlash o'qituvchilarining 72 foizi o'zlarining zamonaviy IT bozori talablaridan ortda qolayotganini his etadi va kasbiy malaka oshirish uchun tizimli imkoniyatlarning etishmasligidan shikoyat qiladi.

Xalqaro tajriba, xususan Finlandiya, Singapur va Janubiy Koreya ta'lim tizimlari tahlili shuni ko'rsatadiki, muvaffaqiyatli dasturlash ta'limining uchta asosiy ustuni mavjud: yuqori malakali o'qituvchi, real hayotga bog'liq o'quv dasturi va zamonaviy o'qitish metodologiyasi. Ushbu maqolada aynan shu uchta ustunni mustahkamlashga yo'naltirilgan yondashuvlar tahlil qilinadi.

3. DASTURLASH TA'LIMIDAGI DOLZARB MUAMMOLAR

Texnikumlarda dasturlash fanlarini o'qitishda bir necha turdagi muammolar mavjud bo'lib, ularni shartli ravishda tizimiy, metodologik va texnologik muammolarga bo'lish mumkin.

3.1. Tizimiy muammolar

- O'quv dasturlarining IT bozori bilan muvofiqlik muammosi: ko'plab texnikumlarda hali ham 2015–2018 yillardagi texnologiyalar o'qitilmoqda, holbuki sanoat allaqachon yangi asboblari va tillar tomonga siljigan;
- Amaliy mashg'ulot uchun zamonaviy jihozlar va internet infratuzilmasining etishmasligi, ayniqsa tuman va qishloq texnikumlarida bu muammo o'ta keskin;

- Reyting tizimining faqat nazariy bilimni o'lchashga yo'naltirilganligi – o'quvchi yaxshi baholansa ham, real loyiha yarata olmaydi;
- O'qituvchi va sanoat mutaxassislari o'rtasidagi aloqaning zaifligi – texnikum bitiruvchilari va IT kompaniyalari talablari o'rtasida katta jarlik mavjud.

3.2. Metodologik muammolar

- An'anaviy "o'qituvchi gapiradi – o'quvchi tinglaydi" yondashuvi dasturlash kabi interaktiv fan uchun samarasiz;
- Muammoga asoslangan o'qitish (Problem-Based Learning) ning etishmasligi – o'quvchilar muammoni tanib olish va uni algoritmgga aylantirish ko'nikmalarini rivojlantirmaydi;
- Dars vaqtining asosiy qismi kod yozishga emas, balki nazariyani tushuntirishga sarflanishi;
- O'quvchilarning individual rivojlanish sur'atiga moslashilmasligi – hamma bir xil tezlikda o'rgatiladi.

3.3. Texnologik muammolar

- O'qituvchilarda sun'iy intellekt vositalaridan foydalanish ko'nikmalarining etishmasligi va ulardan qo'rqish;
- O'quvchilarning AI dan foydalanishga taqiq qo'yilishi natijasida ular buni yashirinch, savodsiz tarzda qo'llashi;
- Zamonaviy dasturlash muhitlari (IDE), versiya nazorat tizimlari (Git) va bulutli platformalar bilan ishlash ko'nikmalarining yo'qligi;
- Kiberxavfsizlik va ma'lumotlar maxfiyligi bo'yicha elementar savodxonlikning etishmasligi.

4. ZAMONAVIY O'QITISH YONDASHUVLARI VA METODLARI

4.1. Loyiha asosidagi ta'lim (Project-Based Learning – PBL)

Loyiha asosidagi ta'lim yondashuvi o'quvchilarni real dunyo muammolarini hal qilishga yo'naltiradi. Bu metodning asosi shundan iboratki, o'quvchi bilimni alohida mavzu sifatida emas, balki konkret muammoni yechish vositasi sifatida qabul qiladi. Masalan, Python dasturlash tilini o'rgatishda quyidagi bosqichli loyiha yondashuvi samarali natija beradi:

- 1-hafta: Ma'lumotlarni kiritish va chiqarish – "Sinf jurnalini ilovasi" loyihasi (input/output, o'zgaruvchilar);
- 2-3-hafta: Sikllar va shartlar – "Savdo do'koni chegirmalar hisoblagichi" (if/else, for, while);
- 4-5-hafta: Funksiyalar – "Kuchli parol generatori" (def, parametrlar, return);
- 6-8-hafta: Fayllar bilan ishlash – "Shaxsiy xarajatlar kundaligi" (open, read, write, JSON);
- Yakuniy loyiha: O'quvchi o'zi tanlagan muammoni hal qiluvchi to'liq dastur yaratadi.

MIT va Stanford universitetlari tadqiqotlari (2022) shuni ko'rsatdiki, PBL usulida o'qitilgan talabalar bilimni 40% uzoqroq eslab qoladi va kasbiy muhitga moslashish davrini 3 oy qisqartiradi. Buning sababi shuki, PBL orqali o'quvchi nafaqat sintaksisni, balki muammoni tahlil qilish, loyihalashtirish, sinovdan o'tkazish va taqdim etish ko'nikmalarini ham egallaydi.

4.2. Sun'iy intellekt vositalarini darsga integratsiyalash

Sun'iy intellekt vositalarini ta'limdan butunlay chiqarib tashlash – bu zamonaviy o'quvchilarni kelajak kasbga tayyorlamayotganligimizni anglatadi. IT kompaniyalarining 2024-yildagi so'rovnomasiga ko'ra, 89% katta texnologiya kompaniyalari dasturchilardan AI vositalaridan foydalanish ko'nikmalarini talab qilmoqda. Demak, muammo AI ni "bloklash"da emas – uni to'g'ri o'qitishda.

Darsda AI dan samarali foydalanishning isbotlangan usullari:

- "AI kod yozdi – sen tushuntir" usuli: O'qituvchi ChatGPT ga vazifa beradi, u kod yozadi. O'quvchilar esa kodni satr-satr tahlil qilib, har bir qatorni tushuntiradilar. Bu usul o'qish (reading comprehension) ko'nikmalarini rivojlantiradi;
- "Bug hunt" (xato qidirish): O'qituvchi yoki AI ataylab 3–5 ta xato mavjud kod taqdim etadi. O'quvchilar xatolarni topib, tuzatib, nima uchun xato ekanini izohlaydilar;
- "Kod review" (kod ko'rib chiqish): O'quvchi o'z kodini GitHub Copilot ga tekshirtiradi, keyin Copilot takliflarini qabul qiladi yoki rad etadi va nima uchun ekanini asoslaydi;
- "Prompt engineering" darslari: O'quvchilar AI ga to'g'ri savol berish san'atini o'rganadilar – bu 21-asr muhim ko'nikmalaridan biri hisoblanadi;
- "AI bilan raqobat": O'quvchi va AI bir xil muammoni yechadilar, so'ngra ikki yechim taqqoslanadi va o'quvchi o'z yechimining afzalliklari yoki kamchiliklarini asoslaydi.

Muhim eslatma: AI vositalaridan foydalanishni o'rgatishda o'qituvchi o'quvchilarga akademik halollik, intellektual mulk huquqlari va AI natijalarini tanqidiy baholash ko'nikmalarini ham singdirishi shart.

4.3. Teskari sinf modeli (Flipped Classroom)

Teskari sinf modelida an'anaviy "sinf – nazariya, uy – amaliyot" tamoyili aksiniga o'zgartiriladi. O'quvchilar yangi mavzuni uyda qisqa video-darsliklar (5–15 daqiqa) orqali o'rganadilar, sinfdagi vaqt esa to'liq muammolarni birgalikda hal qilishga sarflanadi. Bu yondashuv dasturlash o'qitishi uchun ayniqsa qulay, chunki:

- Har bir o'quvchi o'z sur'atida, to'xtab, qayta ko'rib, materiallarni o'zlashtirib keladi;
- Dars vaqtida o'qituvchi 30 o'quvchi uchun emas, balki har biriga individual muammoni hal qilishda yordam beradi;
- O'quvchilar bir-biriga yordam beradi – bu peer learning (tengdoshdan o'rganish) effektini yuzaga keltiradi;
- YouTube, Stepik, Udemy kabi platformalarda o'zbek tilidagi va rus tilidagi sifatli bepul resurslar mavjud.

Amaliy misol: O'qituvchi "Rekursiya" mavzusiga oid 10 daqiqalik video tayyorlab, YouTube yoki Telegram kanaliga joylaydi. O'quvchilar uyda ko'rib keladilar. Keyingi darsda hech qanday ma'ruza bo'lmaydi – to'g'ridan-to'g'ri rekursiv masalalarni yechishga o'tiladi.

4.4. Juftlikda dasturlash (Pair Programming)

Juftlikda dasturlash metodi professional dasturlashda (XP – Extreme Programming metodologiyasida) keng qo'llaniladi va ta'limda ham ajoyib natijalar beradi. Bir kompyuter oldida ikki o'quvchi o'tiradi: biri kod yozadi ("driver"), ikkinchisi strategik yo'l ko'rsatadi va xatolarni nazorat qiladi ("navigator"). Har 15–20 daqiqada rollar almashadi.

Bu metodning afzalliklari juda ko'p. Birinchidan, yolg'iz qolganda e'tiborni yo'qotish muammosi bartaraf bo'ladi. Ikkinchidan, "navigator" roli muammoni yuqori darajada tushunishni talab qilgani uchun chuqur o'zlashtirish ta'minlanadi. Uchinchidan, o'quvchilar kommunikatsiya va jamoada ishlash ko'nikmalarini

rivojlantiradi – bu IT sanoatida eng muhim kompetentsiyalardan biri. Tadqiqotlar ko'rsatdiki, juftlikda dasturlash kodning sifatini 15% oshiradi va xatolar sonini 50% gacha kamaytiradi.

4.5. Gamifikatsiya va musobaqaviy o'qitish

Gamifikatsiya – ta'lim jarayoniga o'yin elementlarini kiritish – o'quvchilarning ichki motivatsiyasini oshirishning eng samarali usullaridan biri hisoblanadi. 2024-yilda Janubiy Koreya o'rta ta'lim muassasalarida o'tkazilgan keng ko'lamli tadqiqot (n=1200) gamifikatsiya elementlari kiritilgan dasturlash darslarida qatnashish faolligini 58%, mustaqil mashq vaqtini esa 45% ga oshirishini isbotladi.

Texnikumda gamifikatsiyani amalga oshirishning amaliy yo'llari:

- LeetCode, Codeforces, HackerRank platformalarida haftaga bir marta guruh musobaqasi – natijalar umumiy reyting jadvaliga qo'shiladi;
- "Dasturchi unvonlari" tizimi: Newbie (yangi boshlagan) → Junior → Middle → Senior – o'quvchi bajargan loyihalar soniga qarab unvon oshiriladi;
- "Stack Overflow" ichki versiyasi: o'quvchilar o'rtasida savol-javob platformasi, to'g'ri javob bergan o'quvchi ball to'playdi;
- Semester oxirida "Hackathon": 4–6 soatlik intensiv musobaqa, g'oliblar guruh nomi bilan texnikum stendi yoki Telegram kanalida e'lon qilinadi;
- GitHub contributions grafigi – o'quvchilarning kunlik "commit" faolligi vizual tarzda ko'rsatiladi.

5. O'QUV DASTURINI ZAMONAVILASHTIRISHNING ASOSIY YO'NALISHLARI

Zamonaviy IT bozori tahlili asosida texnikum dasturlash o'quv dasturlariga kiritilishi lozim bo'lgan asosiy yangiliklar quyidagicha tizimlashtirish mumkin. Quyidagi jadvalda an'anaviy va zamonaviy o'quv dasturi o'rtasidagi farq ko'rsatilgan:

Bo'lim	An'anaviy dastur	Zamonaviy tavsiya
Dasturlash tili	Pascal, Delphi asosiy til	Python (kirish), JavaScript (veb), C++ (algoritm)
Loyihalash	Faqat konsol dasturlari	Veb-illovalar, Telegram-botlar, GUI dasturlar
Versiya nazorat	Mavjud emas	Git va GitHub asoslari (1-kursdan)
Ma'lumotlar bazasi	MS Access, dBase	SQLite, PostgreSQL, MongoDB asoslari
AI va avtomatizatsiya	Kiritilmagan	Prompt engineering, AI API bilan ishlash asoslari
Kiberxavfsizlik	Kiritilmagan	Asosiy xavfsizlik tamoyillari, parol xavfsizligi

O'quv dasturini yangilashda muhim tamoyil shundan iboratki, har bir yangi texnologiya bozor talabi bilan asoslanishi kerak. Har semester boshida mahalliy IT kompaniyalari bilan qisqa uchrashuvni tashkil etish va ulardan "bizga qanday mutaxassis kerak?" savoliga javob olish – o'quv dasturini dolzarb saqlashning eng ishonchli usuli.

6. ZAMONAVIY DASTURLASH O'QITUVCHISI KOMPETENTSIYASI

Zamonaviy texnologiyalar o'qituvchi oldiga ham yangi talablar qo'ymoqda. Bugun yaxshi dasturlash o'qituvchisi uchun nafaqat bilim, balki bir qancha keng ko'lamli kompetentsiyalar zarur. Bu kompetentsiyalarni to'rtta asosiy guruhga ajratish mumkin:

6.1. Texnik kompetentsiyalar

- Kamida ikkita dasturlash tilini professional darajada bilish (masalan, Python va JavaScript);
- Versiya nazorat tizimlari: Git, GitHub yoki GitLab bilan ishlash;
- Zamonaviy AI vositalaridan: ChatGPT, GitHub Copilot, Google Gemini dan samarali foydalana olish;
- Bulutli hisoblash platformalari asoslari: Google Cloud, Microsoft Azure yoki Amazon AWS bilan tanishish;
- Ma'lumotlar bazasi: SQL va NoSQL tizimlarida asosiy operatsiyalarni bajara olish.

6.2. Pedagogik kompetentsiyalar

- Differentsiallashtirilgan ta'lim: turli o'quv darajasidagi o'quvchilar bilan ishlash metodlarini bilish;

- Formativ baholash usullarini qo'llash – doimiy, jarayon davomidagi baholash, nafaqat yakuniy;
- Inkluziv ta'lim tamoyillari – barcha o'quvchilar, shu jumladan maxsus ehtiyojli o'quvchilar uchun qulay muhit yaratish;
- Motivatsion psixologiya asoslarini bilish – o'quvchida ichki motivatsiyani uyg'otish usullari.

6.3. Professional rivojlanish yo'llari

O'qituvchi uchun professional rivojlanishning bir necha muhim yo'nalishlari mavjud. Birinchidan, onlayn kurslar va sertifikatlar: Coursera, edX, Google Career Certificates platformalarida bepul va to'lovli kurslar mavjud. Ikkinchidan, professional hamjamiyatlar: Stack Overflow, GitHub, mahalliy IT jamoatchilik guruhlar (Telegram, Discord). Uchinchidan, amaliy tajriba: dam olish vaqtida real loyihalarda ishtirok etish yoki frilanserlik tariqasida kichik buyurtmalarni bajarish o'qituvchining zamonaviy bilimlarini yangilab turadi. To'rtinchidan, konferensiyalar: IT va ta'lim sohasidagi ilmiy-amaliy anjumanlarda faol qatnashish zarur.

7. O'QITUVCHILAR UCHUN AMALIY TAVSIYALAR

Yuqoridagi tahlillar asosida O'rta Chirchiq tuman texnikumi kabi o'quv muassasalarida dasturlash fanini samaraliroq o'qitish uchun quyidagi amaliy tavsiyalar ishlab chiqildi:

7.1. Dars tashkilotiga oid tavsiyalar

- Har bir darsning kamida 60 foizini amaliy mashg'ulotga ajratin: nazariya – 20 daqiqa, amaliyot – 50 daqiqa, tahlil va muhokama – 10 daqiqa;
- Har hafta bir marta "Code review" sessiyasini o'tkazing: o'quvchilar bir-birining kodini ko'rib, konstruktiv fikr bildiradi;

- Har oyda kamida bitta mini-loyiha tugallansin va sinf oldida taqdim etilsin – bu taqdimot ko'nikmasini ham rivojlantiradi;
- "Debug" mashg'ulotlarini muntazam o'tkazing: o'quvchilarga ataylab noto'g'ri kod bering va ular uni tuzatsin;
- Juftlikda yoki 3 kishilik guruhda ishlash mashqlarini kundalik dars tarkibiga kiriting.

7.2. Motivatsiya va baholash

- Portfolio tizimini joriy qiling: o'quvchi har bir yaratgan dasturini GitHub da saqlaydi – bu bir yil oxirida kuchli portfolio bo'ladi;
- Faqat test va imtihon orqali baholamang – loyiha sifati, ijodkorlik va jamoada ishlash ham baholash mezoniga kirsin;
- O'quvchi muvaffaqiyatlarini sinf Telegram kanalida e'lon qilib boring – ijtimoiy tan olish kuchli motivator;
- "O'quvchi o'qituvchi" metodini qo'llang: biror mavzuni yaxshi o'zlashtirgan o'quvchi boshqalarga tushuntiradi;
- Semestr oxirida "Eng yaxshi loyiha", "Eng ko'p taraqqiy etgan o'quvchi", "Eng ijodkor yechim" kabi nominatsiyalar e'lon qiling.

7.3. Sanoat bilan aloqa

- Har oyda bir marta mahalliy dasturchi yoki IT mutaxassisini darsga taklif qiling – hatto onlayn formatda ham bo'lsa;
- IT Park Toshkent va viloyat texnoparklari bilan aloqa o'rnatish: ular ko'pincha texnikumlar uchun ekskursiya va treninglar tashkil etadi;
- O'quvchilar uchun soha kompaniyalarida qisqa amaliyot (1–2 haftalik) ni o'quv yili oxiriga rejalang;

- Real buyurtmachi uchun loyiha: mahalliy do'kon, shifoxona yoki maktab uchun kichik dastur yaratish – bu amaliy tajribaning eng yaxshi usuli.

8. TAKLIF ETILAYOTGAN YONDASHUVLAR SAMARADORLIGINI BAHOLASH

Yangi metodlar qanchalik samarali ekanini ob'ektiv baholash uchun aniq mezonlar va o'lchov vositalari zarur. Texnikum darajasida quyidagi ko'rsatkichlarni muntazam kuzatib borish tavsiya etiladi:

Miqdoriy ko'rsatkichlar:

- Dars tashrif ko'rsatkichi (attendance rate) – semestr boshiga nisbatan semestr oxirida;
- Mustaqil ishlash vaqti – o'quvchilar darsdan tashqari necha soat dasturlash bilan shug'ullanadi;
- GitHub commit faolligi – har haftada o'quvchilar o'rtacha necha marta kod yuklaydi;
- Loyiha yakunlash darajasi – semestrda rejalashtirilgan loyihalarning nech foizi muvaffaqiyatli yakunlangan;
- Kasbiy imtihon natijalari – IT sertifikat imtihonlari (Python Institute, Cisco, Oracle) ga tayyorgarlik darajasi.

Sifat ko'rsatkichlari:

- O'quvchi va ota-onalar so'rovnomasi – semestrda ikki marta (o'rta va oxirida);
- O'qituvchi kuzatuv jurnali – har haftalik qaysi metodlar eng samarali ishlaganini qayd etish;

- Bitiruvchilar ish joyi – 6 oy o'tgach, bitiruvchilarning qancha foizi IT sohasida ishlayotgani;
- Fokuslashtirilgan guruh suhbatlari – har semestr oxirida 5–7 o'quvchi bilan chuqur suhbat.

9. XULOSA

Ushbu maqolada texnikum dasturlash ta'limidagi dolzarb muammolar tahlil qilinib, ularni hal etishning ilmiy asoslangan va amaliyotda sinovdan o'tgan yondashuvlari ko'rib chiqildi. Sun'iy intellekt davri dasturlash ta'limini yo'q qilmaydi – aksincha, uni sifat jihatidan yangi bosqichga ko'taradi. Bugun texnikum o'quvchilariga faqat kod yozishni o'rgatishdan ko'ra muhimroq narsa – bu muammoni tahlil qilish, mantiqiy fikrlash, jamoada ishlash va texnologiyadan samarali foydalanish kompetentsiyalarini shakllantirish.

Loyiha asosidagi ta'lim, teskari sinf modeli, gamifikatsiya va AI vositalarini integratsiyalash – bu alohida-alohida qo'llaniladigan usullar emas, balki bir-birini to'ldiruvchi kompleks yondashuv tarkibiy qismlaridir. Ularga dasturlash o'qituvchisining doimiy professional rivojlanishi va sanoat bilan mustahkam aloqa qo'shilganda, natija sezilarli bo'ladi.

O'zbekiston Respublikasining "Raqamli O'zbekiston – 2030" strategiyasi maqsadlariga erishish ko'p jihatdan aynan texnikum va kasb-hunar ta'lim muassasalarida sifatli IT ta'limini tashkil etishga bog'liq. Bu mas'uliyatni his etgan holda, har bir dasturlash o'qituvchisi o'z darsini bugungi emas, ertangi kun talablariga mos holda loyihalashi kerak.

FOYDALANILGAN ADABIYOTLAR

1. O'zbekiston Respublikasi Prezidentining 2020-yil 5-oktyabrdagi PF-6079-son Farmoni "Raqamli O'zbekiston – 2030" strategiyasi to'g'risida. – Toshkent, 2020.

2. Wing J. M. Computational thinking // Communications of the ACM. – 2006. – Vol. 49, №3. – P. 33–35.
3. Bransford J., Brown A., Cocking R. How People Learn: Brain, Mind, Experience, and School. – Washington: National Academy Press, 2000. – 374 p.
4. Freeman S. et al. Active learning increases student performance in science, engineering, and mathematics // PNAS. – 2014. – Vol. 111, №23. – P. 8410–8415.
5. Thomas J. W. A Review of Research on Project-Based Learning. – San Rafael: The Autodesk Foundation, 2000. – 83 p.
6. Malik M., Ayub M. Artificial Intelligence in Education: Opportunities and Challenges // Journal of Educational Technology. – 2024. – №3. – P. 45–62.
7. UNESCO. Guidance for generative AI in education and research. – Paris: UNESCO Publishing, 2023. – 64 p.
8. GitHub Education. Teaching and Learning with GitHub. – 2024. URL: <https://education.github.com> (murojaat: 22.04.2026).
9. Toshkentov D., Rahimov B. IT ta'limida zamonaviy yondashuvlar: darslik. – Toshkent: TATU nashriyoti, 2024. – 180 b.
10. TATU Ilmiy-tadqiqot markazi. O'zbekiston texnikumlarida IT ta'limi holati: tahliliy hisobot. – Toshkent, 2023.
11. Kim Y. et al. Gamification in CS education: A systematic review // Computers & Education. – 2024. – Vol. 198. – P. 104781.
12. Williams L., Kessler R. Pair Programming Illuminated. – Addison-Wesley Professional, 2002. – 256 p.